

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**



**НАВЧАЛЬНИЙ ПОСІБНИК**  
**«Базові засади веб-розробки»**

Лабораторний практикум з дисципліни «Глобальні інформаційні мережі»

для студентів галузі знань «12 Інформаційні технології»  
спеціальності 123 «Комп'ютерна інженерія»

Рекомендовано Науково-методичною радою  
Національного університету "Львівська політехніка"

Автори: **Юрчак І.Ю.**, к.т.н., доцент кафедри ЕОМ

**Гузинець Н.В.**, асистент кафедри ЕОМ

Рецензенти: **Морозов Ю.В.** - к.т.н., доцент, доцент кафедри ЕОМ

**Ханас Ю.Ю.** – старший викладач кафедри САП

**Шаров Б.Г.** – к.т.н., директор ТзОВ «Елси»

Рекомендовано Науково-методичною радою  
Національного університету “Львівська політехніка”  
(протокол №      від      )

Навчальний посібник «Базові засади веб-розробки». Лабораторний практикум з дисципліни «Глобальні інформаційні мережі»: для студентів галузі знань «12 Інформаційні технології» спеціальності 123 «Комп’ютерна інженерія»/ Автори Юрчак І.Ю., Гузинець Н.В. – Львів: Видавництво Національного університету “Львівська політехніка”, 2022. – 180 с.

До посібника увійшли методичні вказівки до лабораторних робіт з навчальної дисципліни «Глобальні інформаційні мережі», питання для самоперевірки і термінологічний словник. Методичні вказівки до лабораторних робіт містять теоретичні матеріали, покрокові інструкції для виконання, варіанти індивідуальних завдань, контрольні питання, вимоги до звітів, електронні джерела інформації, рекомендовану літературу.

Навчальний посібник призначений для студентів спеціальності “Комп’ютерна інженерія” галузі знань “Інформаційні технології”.

© Юрчак І.Ю, Гузинець Н.В. 2022

© Національний університет

«Львівська політехніка», 2022

ISBN

Навчальний посібник з дисципліни “Глобальні інформаційні мережі” призначений для студентів першого (бакалаврського) рівня вищої освіти спеціальності 123 “Комп’ютерна інженерія”.

Лабораторні роботи орієнтовані на вивчення основних принципів та базових засобів створення веб-ресурсів, які є основними складовими глобальних інформаційних мереж.

Виконання лабораторних робіт побудовано на покроковому вивченні базових основ створення веб-документів, наповнення їх контентом різного типу, інтеграції зовнішніх сервісів, розміщенні на серверах.

При цьому опрацьовуються такі питання:

- Створення сайту за допомогою конструкторів.
- Створення статичного веб-сайту за допомогою HTML, CSS, JavaScript.
- Ознайомлення з теоретичними засадами веб-графіки та веб-типографіки.
- Практичне засвоєння інтеграції зовнішніх сервісів у веб-проект.
- Практичне опанування фреймворку Bootstrap для створення адаптивного, кросбраузерного та кросплатформного веб-продукту.
- Тестування та відлагодження створеного веб-проекту за багатьма критеріями: працездатність, функціональність, юзабіліті, сумісність тощо.
- Створення динамічного сайту за допомогою CMS, мов серверного програмування та бази даних.

Проведення лабораторних занять не передбачає використання спеціального апаратного чи програмного забезпечення навчальних лабораторій кафедри електронних обчислювальних машин Національного університету “Львівська політехніка”. Всі наведені у посібнику засоби є Open Source, тобто надаються для кінцевого користувача з відкритим кодом за ліцензіями GNU/Linux, MIT та інших. Лабораторний практикум максимально адаптовано до занять в умовах дистанційного навчання.

Методичні матеріали призначені для допомоги студентам у підготовці до лабораторних занять, виконанні індивідуальних завдань, захисті отриманих результатів.

За підсумками лабораторних робіт студенти виконують розрахунково-графічну (самостійну) роботу, яка акумулює в собі практичні знання, отримані під час виконання лабораторних робіт.

Виконання самостійної роботи та термінологічний словник надають змогу студентам перевірити свої знання та вміння при підготовці до контрольного заходу.

# ЗМІСТ

Вступ .....	5
Лабораторна робота № 1. ....	7
Лабораторна робота № 2. ....	23
Лабораторна робота № 3. ....	40
Лабораторна робота № 4. ....	63
Лабораторна робота № 5. ....	77
Лабораторна робота № 6.....	87
Лабораторна робота № 7.....	103
Виконання самостійної роботи.....	123
Додатки	
Додаток А. Основи HTML .....	152
Додаток Б. Основи CSS.....	169
Додаток В. Таблиця поширених елементів HTML.....	173
Термінологічний словник.....	178

## Веб-розробка – сучасні напрямки, технології, професії

Існування Веб налічує майже 30 років. Етапи розвитку Веб накладаються один на одного, привносячи нові можливості і частково замінюючи старі технології щодо подання інформації та взаємодії з користувачем. Веб поєднує потужність використання додатків, розподілених систем та інформаційного змісту нового покоління.

Веб індустрія швидко розвивається - все більше компаній мають свої представництва в Інтернет просторі, створюються нові проекти з різних сфер діяльності людства. У Веб в електронній формі представлено таку велику кількість форм інформації та взаємодій об'єктів, що їх неможливо реалізувати одним інструментом.

Свобода і демократичність Інтернету, відсутність соціальних стримуючих факторів дозволяє легко впроваджувати передові ідеї, об'єднувати у спільноти фахівців і створювати сучасні потужні проекти. Сучасний багатофункціональний проект використовує цілий ряд технологій і їх інтеграція має не порушувати працездатність сайту чи функціональність його складових.

На сьогоднішній день існує величезна кількість технологій для створення веб-проектів різного призначення і складності. Вони різняться між собою специфікою застосування, популярністю, вартістю тощо. Оптимальне поєднання технологій дозволяє досягнути високої продуктивності, стабільності, масштабованості, знизити витрати, пов'язані з розробкою та супроводом веб-проектів, а також запобігти можливості виникнення типових проблем у майбутньому.

Взаємодія веб-технологій, завдяки яким все працює злагоджено, залишається прихованою від звичайного користувача. І це є результатом безперервних зусиль великих ІТ-компаній та відкритої веб-спільноти, яка допомагає розробляти новітні технології, такі як HTML5, CSS3, WebGL тощо, і домагається їх підтримки всіма браузерами.

Допоки не існує єдиної класифікаційної і сертифікаційної системи оцінки сайтів, це відкриває широкі перспективи перед розробниками, дозволяючи розвиватися і реалізовувати найнесподіваніші веб-проекти. Грамотний підхід дозволяє зробити сайт - зручним, простим, ергономічним, відповідним до поставлених завдань і естетично привабливим. Саме це перетворює веб-проект в якісний продукт творчості, глибоких знань і великого досвіду веб-розробників.

Кожний рік в Інтернеті відбуваються якісні зміни. Вони пов'язані з появою нових технологій, народженням нових концепцій, ідей і знань. Веб починає потребувати сучасні кадри. В середовищі веб-розробників з'являються нові професії і спеціалізації, які потребують спеціальні знання та досвід. Чим складніша індустрія, тим важче бути в ній професіоналом.

Науково-технічний прогрес, соціальні зміни, економічні процеси відбуваються настільки швидко, що всього знати просто неможливо. Спеціаліст повинен володіти все більшими знаннями про доволі вузьку ділянку своєї спеціалізації. В кожного професійного веб-фахівця і в кожної професійної веб-студії своя спеціалізація. Сьогодні над розробкою проекту працює група фахівців: дизайнерів, верстальників, програмістів, оптимізаторів, копірайтерів та інших.

Виникнення нових спеціальностей є корисним для суспільства, оскільки це позбавляє від дилетантів-розробників. В середньої людини пам'яті і обчислювальних здібностей менше, ніж у комп'ютера. Тому, людського ресурсу, зазвичай, не вистачає на те, щоб якісно володіти

кількома професіями одночасно. Багато спеціалізацій розгалужується і знання однієї професії перекладаються на іншу.

Безперервна еволюція Веб призводить до того, що для більшості непрофесійних розробників невисокої кваліфікації не залишиться можливостей працювати без оволодіння спеціальними знаннями. Мережі потрібні вузькі фахівці. Потрібно вибирати тему і спеціалізуватися вже сьогодні, щоб не залишитися без роботи завтра.

Розвиток веб не зупиняється, і якщо веб-розробник не буде вчитися і практикувати, то доволі швидко він залишиться далеко позаду. Слідування за розвитком вимагає більших зусиль, щоб бути в курсі останніх тенденцій веб-індустрії, але придбані знання і досвід нададуть веб-фахівцям завжди бути затребуваним. Обмеження, що накладаються розвинутою індустрією та новітніми веб-технологіями, не дозволяють створювати неякісний Інтернет ресурс, і залишають тих, хто не встиг за ними в своєму розвитку.

Якщо Ви вже поринули в цей захоплюючий світ, вдосконалюйте свої знання, ставайте мудрішими і досвідченішими. Інакше доведеться спостерігати за цим світом з боку, а не творити його власними руками.

# Лабораторна робота №1. Конструктори сайтів

**Мета роботи.** Ознайомитися з популярними конструкторами сайтів, що надають змогу легко створити сайт за шаблоном. Набути практичних навичок з користування конструктором, застосувати його можливості для створення повноцінного, працездатного сайту і розмістити його на безкоштовній хостинговій площадці.

## Конструктори сайтів

З появою і швидким розвитком Інтернет перед користувачами відкрилося багато нових можливостей, зокрема, можливість спілкування. Тепер можна обговорювати різні теми на форумах і отримувати цінні поради, розповідати про себе у блозі, знаходити давніх друзів за допомогою соціальних мереж і багато іншого.

Значно ширше поле для діяльності відкриває для користувачів наявність власного сайту. Це можливість заявити про себе, свої захоплення, роботу, компанію. Але створення навіть простого за дизайном чи функціоналом сайту потребує базових знань мов HTML, CSS та JavaScript, навичок обробки зображень, уставляння мультимедійних об'єктів, розміщення веб-документів на серверах Інтернету.

В міру зростання інтересу до створення сайтів, з'явилися й особливі платформи – конструктори сайтів. З їх допомогою користувач будь якого рівня обізнаності може створити власний сайт-візитку, багатосторінковий сайт для компанії або онлайн-магазин.

Конструктор сайтів дозволяє сформувати і об'єднати веб-сторінки в цілісну структуру сайту, а також керувати ними, не володіючи спеціальними технічними знаннями і навичками. Створений в конструкторі ресурс розміщується на хмарі - віддаленому сервері-хостингу, збереження і працездатність якого підтримується командою адміністраторів конструктора без втручання користувача.

Конструктори мають дружній інтерфейс, що буде зрозумілим для непідготовленого користувача. Редагування сторінок, зовнішнього вигляду дизайну і загальне налаштування відбувається в онлайн-режимі за допомогою зручної панелі управління.

### Переваги конструкторів сайтів

Для звичайного користувача, який не має особливих знань і навичок, конструктор є одним з кращих варіантів швидко і якісно створити власний інформаційний ресурс в Інтернеті, розпочати онлайн бізнес або презентувати власні досягнення.

- **Швидкість створення.** Користувачі «конструюють» дизайн на основі блоків або обирають вже готові шаблони. Створити сайт можна за кілька годин.
- **Ціна.** Вартість створення сайту за допомогою конструктора значно нижча, ніж звертатися до професіональних веб-фахівців, чия робота оцінюється погодинно і може коштувати чимало.
- **Простота.** Жодних особливих знань для роботи в конструкторі не потрібно. В основному функціоналі можна розібратися за допомогою зрозумілих інструкцій.

- **Доступність.** Конкуренція між розробниками конструкторів призвела до того, що значна частина базових рішень надається безкоштовно. Платними залишаються особливі функції і додаткові можливості.
- **Наявність шаблонів.** Користувачу, що створює сайт самостійно, надається можливість вибрати один із готових дизайнів, який надається до редагування: зміна колірної схеми, гарнітури та розміру шрифтів, додавання або видалення інформаційних блоків тощо.

### Недоліки конструкторів сайтів

- **Обмеженість.** Конструктор надає можливості створювати сайт з вже готових блоків, і зазвичай, індивідуальні задуми реалізувати важко.
- **Продуктивність і швидкість роботи.** Ресурси, що є невеликими за розміром та складністю працюють доволі швидко. Але, зі збільшенням потужності сайту, його функціонування помітно пригальмовується, що підштовхує власника обрати інший дорожчий тариф.
- **Зростання абонентської плати.** Для простого лендингу або візитки можна обрати безкоштовний або мінімальний тариф. Але із збільшенням функціоналу доведеться змінювати тариф на дорожчий.
- **Складнощі інтеграції сервісів та систем.** Багато сучасних конструкторів намагаються враховувати останні тренди, але існує ризик, що певні сервіси або платформи не будуть підтримуватися.
- **Залежність від розробників сервісу.** Працездатність та доступність сайту залежить від того, як працює команда розробників конструктора. До цього відноситься технічна підтримка, сервісне обслуговування, потужності хостингової хмари, надійність серверів та інше. Також, ризик, що платформа конструктора буде закрита, існує завжди і тоді всі проблеми власник сайту має вирішувати самостійно.

Конструкторів для створення сайтів існує значна кількість, вони мають власні особливості, переваги, тому перед вибором варто ознайомитися з їх функціоналом та оцінити зручність роботи.

### *Конструктор сайтів Wix (<https://uk.wix.com/>)*

**Wix** - популярний конструктор, що орієнтується, в першу чергу, на потреби початківців користувачів з нульовими знаннями щодо створення сайтів (рис.1.1). Ідеально підходить для створення яскравих за формою і змістом візиток. Сервіс поставляється зі зручним візуальним редактором, в якому більшість дій виконується за допомогою мишки.



# Створіть свій професійний сайт

Відкрийте для себе платформу, яка надасть вам повну свободу для створення, дизайну, керування та розробки сайту своєї мрії.

Розпочати

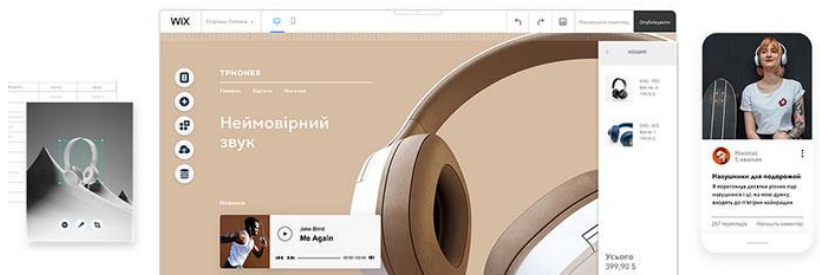


Рис.1.1. Інтерфейс конструктора сайтів Wix

Конструктор надає надвелику збірку безкоштовних шаблонів, які розподілені за тематичними категоріями, що спрощує вибір та визначає сферу їх застосування. Хоча в процесі роботи над сайтом поміняти дизайн на інший не вийде, проте, є значні можливості щодо його зміни.

Панель управління зручна, функціонально добре продумана, реалізовано додавання віджетів, компонентів і різноманітних налаштувань. Серед цікавих можливостей: додавання відео на фон сайту, широкоформатні лендинги, геометричні форми, іконки, ефекти паралакса, анімація і багато іншого. Є один з кращих редакторів зображень міні-Photoshop. Перед публікацією можна обробити фото (кадрування, розтягнення, зміна масштабу, накладення ефектів, корекція кольору) і розмістити у потрібному місці на сторінці.

Вбудований магазин додатків здатний посилити існуючий функціонал. Можна прикріпити форум, під'єднати онлайн платежі, встановити інтерактивного чат-бота (live-chat), додати можливість виставлення рахунків, працювати з налаштуваннями SEO та багато іншого. Асортимент додатків великий і постійно зростає.

Wix надає широкі можливості для початківців, які бажають просунути себе або свій бізнес: створення сайтів-візиток, портфоліо робіт, промо-сторінок, магазинів на невелику кількість товарів, ефектних лендингів, блогів.

## *Конструктор сайтів SITE123 (<https://www.site123.com/>)*

**SITE123** – сервіс, що призначений як для початківців та й для фахівців. Конструктор пропонує інший підхід до формування сторінок ніж Wix. Замість віджетів використовуються модулі - готові блоки, що створені під конкретні завдання. Будь-який з модулів має кілька варіантів оформлення, які піддаються додатковому налаштуванню.

Інтерфейс панелі управління зручний і зрозумілий (рис.1.2). Всі зміни видно в реальному часі, хоча відсутня можливість безпосередньо виділити елемент на сторінці і почати його

редагувати. Зміна сторінок проходить централізовано по заданих напрямних - через налаштування в панелі управління для кожного конкретного модуля.

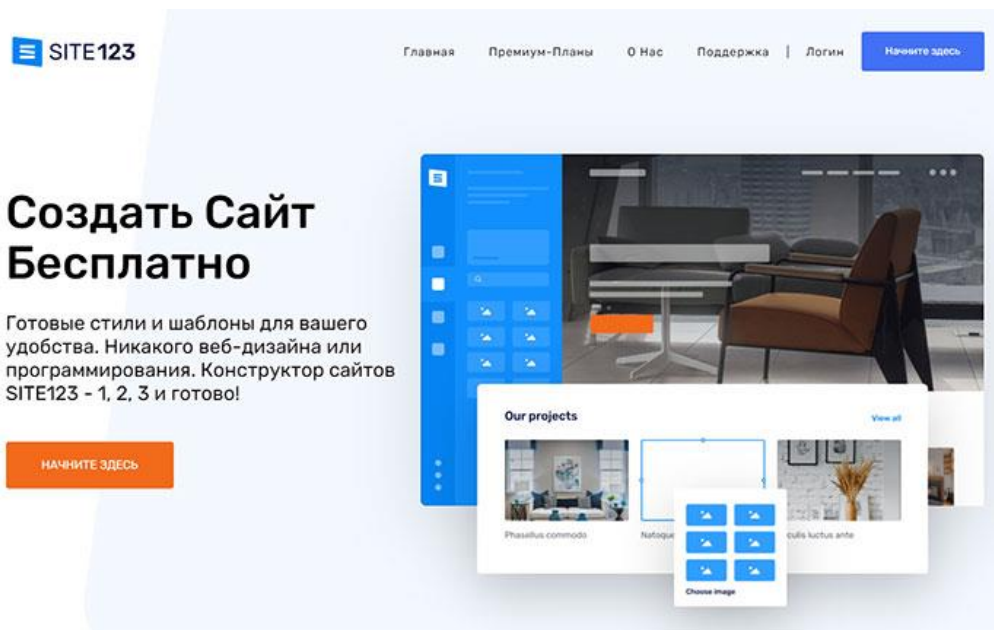


Рис.1.2. Интерфейс конструктора сайтов SITE123

Конструктор надає до використання сучасні шаблони, які мають адаптивний дизайн і легко налаштовуються: вибір типу сайту та структури сторінок, гнучка робота зі шрифтами і кольорами. Використовуючи даний арсенал можна швидко отримати унікальний дизайн.

SITE123 добре підходить для створення магазину з невеликою кількістю товару. Присутня можливість налаштування оплати (PayPal), способів доставки, вибору валюти, можливість впровадити кілька мов на сайт, знижки та інше. Надано збірку плагінів, які допоможуть просунути сайт, зібрати клієнтську базу, аналізувати статистику, інтегрувати сервіси соціальних мереж і багато іншого.

### *Конструктор сайтів Tilda (<https://tilda.cc/ua/>)*

Tilda - конструктор для тих, хто уважно ставиться до презентації контенту, щоб подати матеріали вигідно і красиво (рис.1.3). Інноваційний блоковий механізм редагування надає можливості швидко і легко зібрати сторінку з готових блоків, спроектованих професіоналами. Бібліотека містить сотні блоків і постійно поповнюється. Розробники створюють нові, сучасні елементи відповідно до трендів веб-дизайну. В кожному блоці є гнучкі налаштування для індивідуального дизайну.

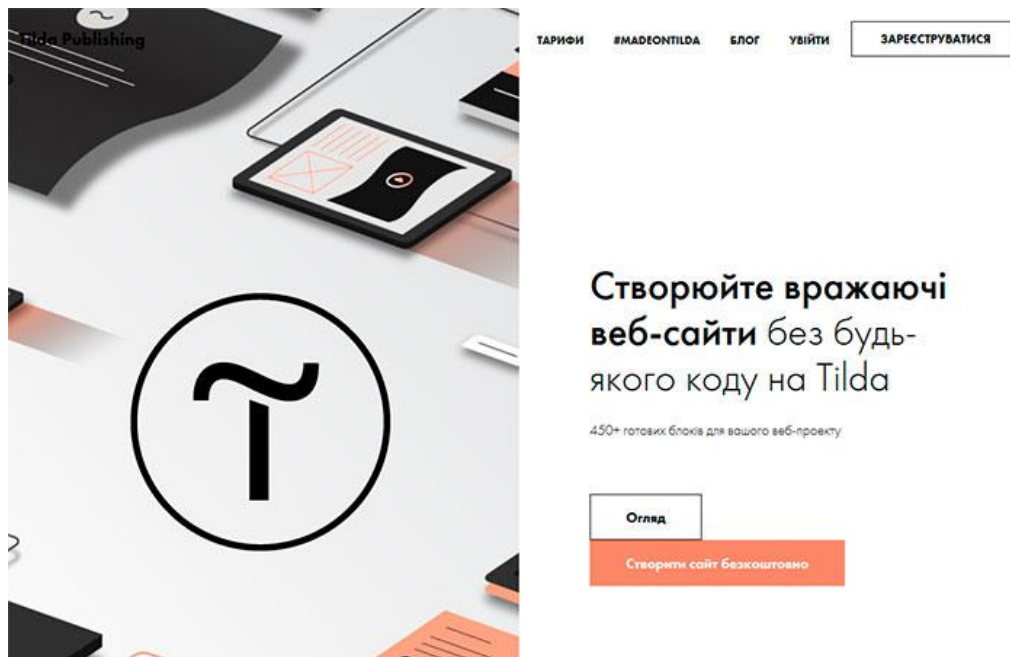


Рис.1.3. Інтерфейс конструктора сайтів Tilda

Надано велику збірку шаблонів готових сторінок: лендинги, тематичні сайти, магазини, промо-сторінки, блоги та інше. Увагу приділено до типографіки: гармонійні пропорції, зручна ширина рядків, правильний інтерліньяж, оптимальний розмір шрифту, комбінації заголовків.

Широкі налаштування в подачі візуального контенту: повноекранні фотографії, готові фотогалереї, комбінації фотографій і тексту. Вбудований редактор зображень надає можливості додавати до зображень надписи, ефекти, тонування, змінити насиченість, яскравість, розмір і пропорції.

Представлено колекцію безкоштовних іконок та бібліотеку безкоштовних зображень. Інструменти роботи з мультимедіа (відео та аудіо контент) сприяють поживленню сторінок: фонове відео (YouTube, MP4, WEBM), вбудовування відео з YouTube і Vimeo, готові поєднання відео та тексту, інтегровані сервіси SoundCloud і Coub, уставка відео та аудіо плеєра через блок html-embed.

Конструктор надає можливості безкоштовного користування, хоча там присутні обмеження: сайт міститиме не більше 50 сторінок, домен від конструктора, обмежена кількість блоків і граничний обсяг завантажених файлів в 50 Мб. Для повноцінної роботи варто скористатися платними тарифними планами.

### *Конструктор сайтів Ucoz (<https://www.ucoz.ua/>)*

uCoz - це сервіс створення та обслуговування сайтів, що має велику кількість переваг та привертає увагу користувачів завдяки своїм характеристикам. Платформа пропонує потужний функціонал, має доступні тарифні плани з багатьма бонусами. Вона – універсальна та надає змогу використовувати багато можливостей абсолютно безкоштовно (рис.1.4).

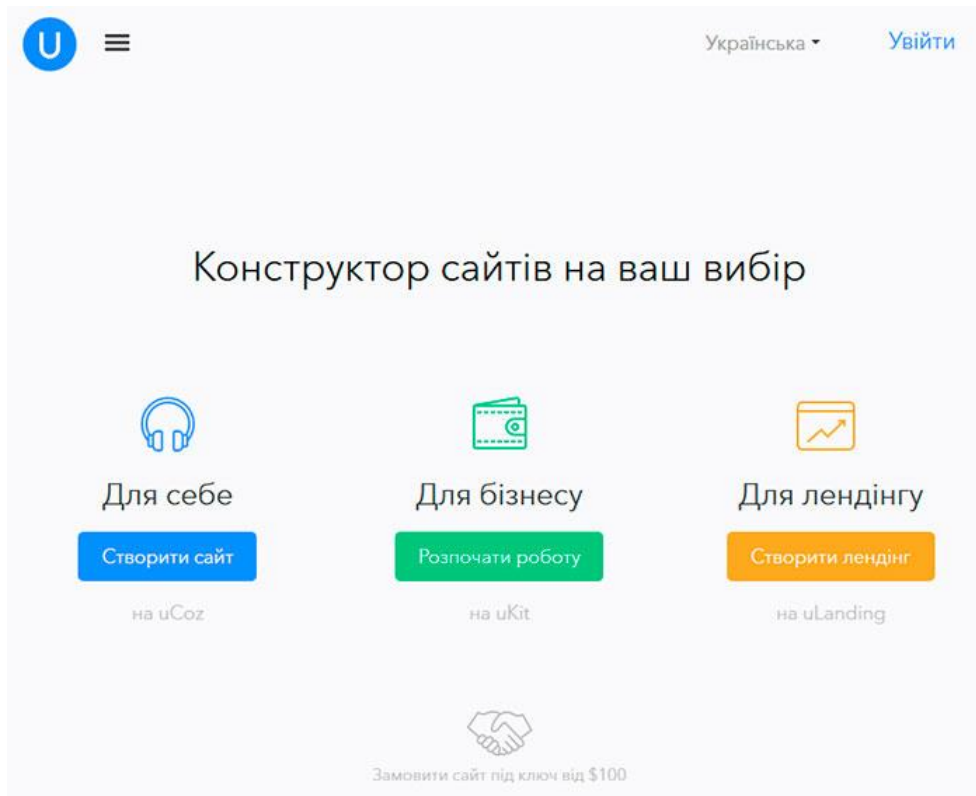


Рис.1.4. Інтерфейс конструктора сайтів uCoz

Єдиний аспект, що може стати перешкодою для початківців – це певна складність оволодіння всіма функціями сервісу. Професіональні розробники знайдуть тут все те, що зазвичай бракує іншим конструкторам сайтів.

Якщо розкрити весь потенціал системи, то можна створити будь-який тип сайту. Завдяки універсальності сервісу, не буде потреби у додаткових інструментах для створення якісного та функціонального ресурсу, що забезпечить суттєву економію часу.

### *Конструктор сайтів Weblium (<https://ua.weblium.com/>)*

Weblium - має естетичну панель керування. Робочий простір ділиться на особистий кабінет зі списком створених проектів і візуальний редактор. Майже вся функціональність сайту вибудована навколо різноманітності готових секцій і налаштування їх компоновок. Інших параметрів мінімум - тільки найнеобхідніше, весь фокус сконцентрований на можливостях візуального редактора високого рівня якості (рис.1.5).

## Weblium – розумний конструктор сайтів

Створіть швидкий та ефективний сайт для  
вашого бізнесу!

Створити сайт безкоштовно

Замовити сайт у студії

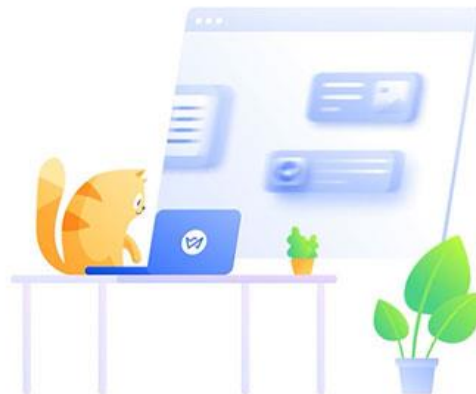


Рис.1.5. Інтерфейс конструктора сайтів Weblium

Однією із особливостей Weblium є вбудований штучний інтелект. Він спроможний підказувати найбільш вдале розміщення блоків, допомогти вибрати кольори, шрифти, правити дрібні недоліки заради досягнення оптимального результату. Користувач обирає секцію, блок, колір, шрифт з багатьох варіантів.

Змінити дизайн можна в один клік. Система супроводжує робочий процес підказками у форматі чек-листа, що є корисним для початківців - вони нічого не пропустять, виконуючи інструкції покроково.

### *Конструктор сайтів WIX.com*

#### **Цільова аудиторія**

Wix найкраще підходить для створення сайтів-візиток - складних з дизайнерської і функціональної точок зору: портфоліо, бізнес-сайт, промо-сторінка або лендинг.

*Область застосування Wix можна охарактеризувати трьома факторами:*

- Сайти з невеликою кількістю сторінок, оскільки дизайн кожної потрібно формувати вручну.
- Необхідність глибокого і тонкого налаштування дизайну, а також введення великої кількості дрібних функціональних елементів.
- Магазин додатків Wix містить великий асортимент додаткового функціоналу. Багато категорій корисних додатків (понад 250 найменувань) допоможуть досягти практично будь-якої мети.

Wix є ідеальним для створення малих і середніх за обсягом сайтів зі складним дизайном та функціональністю. Цей конструктор можна використовувати для конструювання складних сторінок з багатьма деталями: анімаціями, ефектами, формами, інтерактивними елементами, спливаючими вікнами авторизацій різних етапів допуску до інформації, опитуваннями, табами, колонками і іншим.

Wix можна використовувати як для створення першого сайту так і для розробки складних проектів. Система побудована таким чином, що дозволяє легко нарощувати можливості дизайну.

Сервіс здатний допомагати реалізовувати складні з технічної та дизайнерської точок зору задумки. Базово насичений конструктор з широкими додатковими можливостями для масштабування функціоналу.

### Функціональні можливості

Сукупні можливості Wix складаються з: функціоналу редактора, загальних налаштувань, панелі управління, встановлених сторонніх додатків і опцій, які можна впровадити на сайт самостійно за допомогою інструменту Wix Code (рис.1.6).

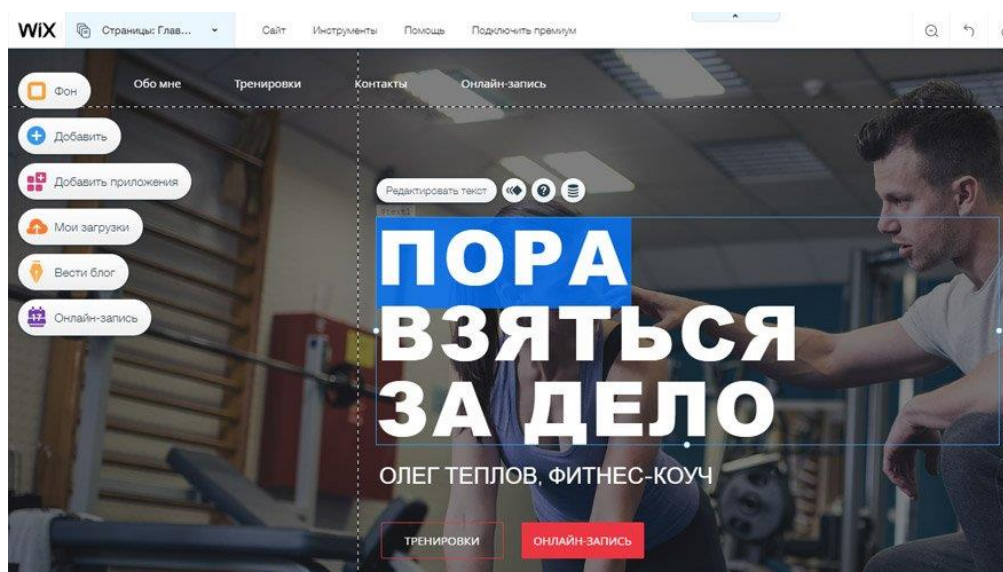


Рис.1.6. Редактор конструктора

### Структура Редактора дизайна сайта Wix:

**Меню і сторінки.** Містить перелік всіх статичних сторінок сайту і пов'язаних з ними пунктів меню. При додаванні сторінки автоматично буде додано пункт меню, який перетягуванням можна поставити в потрібне місце або зробити підпунктом іншого пункту меню (підтримується вкладеність до 2 рівня). Можна створювати нові сторінки, посилання в меню (на ярік, веб-адресу, пошту, номер телефону, документ) і задати сценарій їх відкривання (в поточному або новому вікні). На сайт можна додавати динамічні сторінки, які будуть підтягувати контент в міру його поновлення з хмари або бази даних. Для кожної сторінки можна призначити видимість і налаштувати SEO, дублювати, зробити головною, задати загальний вигляд макета, правила доступу або конвертувати в динамічну сторінку.

**Фон.** Можна вибрати фон для сайту у вигляді картинки (завантаженої або стокової від Wix), кольору або відео. Відео-фон здатний перетворити сайт в яскраве видовище. Ролик для відео-фону можна завантажити з різних джерел. Можна активувати ефект паралакса для додавання сайту зорової глибини при прогортанні за рахунок різниці швидкості руху об'єктів, розташованих на різних відстанях.

**Додати.** Велика добірка типів елементів, які можна додати на сторінку. До неї входять:

- *Текст* - вибір готових стилів тексту / заголовків, робота з текстовим редактором.
- *Database* - додавання контенту в хмарні бази даних, під'єднання джерел даних для звичайних і динамічних сторінок.
- *Фото* - завантаження власних, використання колекцій зображень від Wix (стокові фото, кліпарт) або їх експорт з соціальних мереж.
- *Списки і таблиці* - додавання на сайт списків і таблиць, близько 30 варіантів дизайнерських шаблонів.

**App Market.** Фірмовий магазин додатків, які можна додати на сайт. Використання даного розділу в кілька разів розширює стоковий функціонал платформи. Тут міститься величезна кількість додатків різної спрямованості: форми, робота з соціальними мережами, галереї, додатки для магазину і маркетингу. Можна додати інтерактивний календар, наприклад, інструменти для маркетингу, опитування, Dropbox, eCommerce-модулі, SEO-інструменти та інші модулі в допомогу стандартного функціоналу конструктора. Всі додатки відсортовано за категоріями. Більшість з них розроблені командою Wix, інші, як правило, є інтеграцією можливостей сторонніх сервісів. Приблизно половина додатків є безкоштовними, а інші мають як безкоштовний варіант використання, так і платний з додатковими можливостями.

**Мої завантаження.** Збірка завантажених на хостинг власних фотографій, документів та аудіофайлів. Це користувацька бібліотека, файли якої можна розміщувати на сайті.

**Блог.** Даний розділ з'являється після активації блогу на сайті. Звідси можна додавати нові і редагувати існуючі пости. При створенні матеріалу можна вибрати рубрику, вид посилання, заповнити мета-теги і додати фотографії чи відео. Параметри форматування з'являються після виділення тексту, за замовченням вони приховані. Можна вибрати варіант подачі стрічки блогу - в стовпчик, плитками, картками тощо.

**Онлайн-запис.** Окремий розділ з функціоналом програми Wix Bookings - потужного інструменту для збору заявок на бронювання чого-небудь (запис до лікаря, оренда будинків, номерів готелів тощо). Можна налаштувати прийом групових (заняття йоги, танців) або персональних (послуги перукаря, психотерапевта, наприклад) заявок за календарем, під'єднати прийом онлайн-платежів через PayPal, налаштувати розклад зайнятості для працівників і збирати клієнтську базу. Віджет для доступу відвідувачів до запису на послугу можна розмістити в будь-якій частині сайту. Доступне детальне налаштування макету блоку, його дизайну і вибір відображуваних даних (ціна, опис, тривалість, дні, назва послуги).

### **Панель редактора з загальними опціями**

Верхню частину редактора займає панель з корисними опціями: збереження змін, крок назад, попередній перегляд, публікація, доступ до редактора мобільної версії шаблону. Тут же знаходиться додаткове меню можливостей (рис.1.7):

**Сайт** - меню швидкого доступу до значної кількості важливих функцій і розділів: збереження сайту, репост посилання на сайт в соцмережі або на поштові акаунти для залучення відгуків, публікування і пошукова оптимізація сайту, під'єднання домену, активація версії сайту для користувачів зі слабким зором (обов'язкова річ для сайтів державних організацій), посилення різкості опублікованих фотографій, доступ до загальних налаштувань і панелі управління, а також перегляд історії сайту (перелік резервних копій для відновлення).

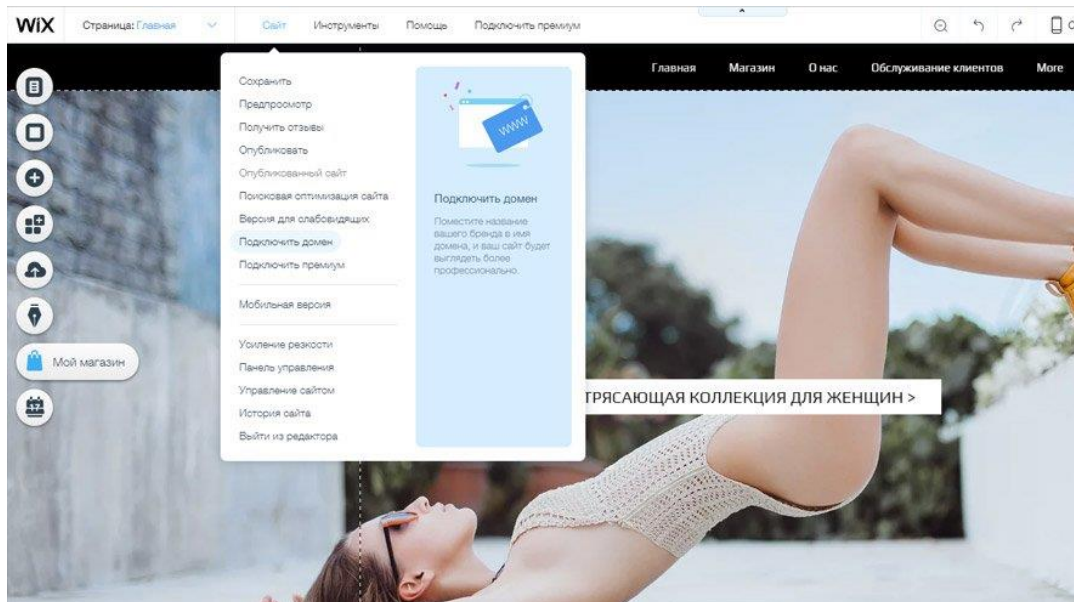


Рис.1.7. Додаткове меню редактора

**Інструменти** - активація панелі інструментів для регулювання розмірів і масштабування елементів сайту, їх автоматичного вирівнювання, додавання лінійки, меж сайту і вмикання функцій розробника (панель властивостей, показ прихованих елементів, активація Wix Code, можливість створення нових функцій за допомогою JavaScript).

**Допомога** - список гарячих клавіш редактора (копіювання, уставка, дублювання, зсув на піксель, повернення, видалення, поворот, вибірка елементів, попередній перегляд та інше), Довідкова система по роботі з редактором і SEO-оптимізації сайту, умови використання та політика конфіденційності Wix.

**Під'єднання преміум** - вибір і покупка відповідного тарифного плану Wix.

### Робота в візуальному редакторі Wix

Візуальний WYSIWYG-редактор підтримує функцію drag-n-drop, тобто можливість перетягувати віджети, потягнувши за ярлик «drag» на їх верхній межі. Шляхом звуження або розширення пунктирних ліній на межах об'єктів можна змінювати за допомогою мишки їх розмір. При виділенні будь-якого елементу або блоку на сторінці з'являються значки, що призначені для налаштувань і швидких дій над об'єктом. Наприклад, для галереї можна вибрати і впорядкувати фото, для відео - поставити посилання на ролик або вибрати завантажений з бібліотеки, призначити правила відтворення і подібне (рис.1.8).

Конструктор має добре реалізований функціонал для роботи з фотографіями і оздобленням сайту. Додаток Wix Pro Gallery, наприклад, дозволяє налаштувати якість і різкість фотографій при відображенні на сайті, захистити їх від вивантаження або додати кнопки «Поділитися». Можна гнучко налаштовувати макет виведення зображень - структуру, розмір мініатюр, відступи, насиченість колажу і орієнтацію в просторі, а також побудувати магазин зображень.

В конструкторі присутня збірка сінемаграфів - колекція зображень з плавною анімацією в форматі GIF (знайти можна в налаштуваннях фону - безкоштовна бібліотека Wix, категорія «Кіно»). Так, конструктор пропонує велику колекцію безкоштовних готових картинок з багатьох тематик, є також платні стокові варіанти. В налаштуваннях фону можна увімкнути паралакс та



інші ефекти прогортання блоків. Те ж саме можна зробити з окремими елементами, після виділення та вибору пункту «Анімація».

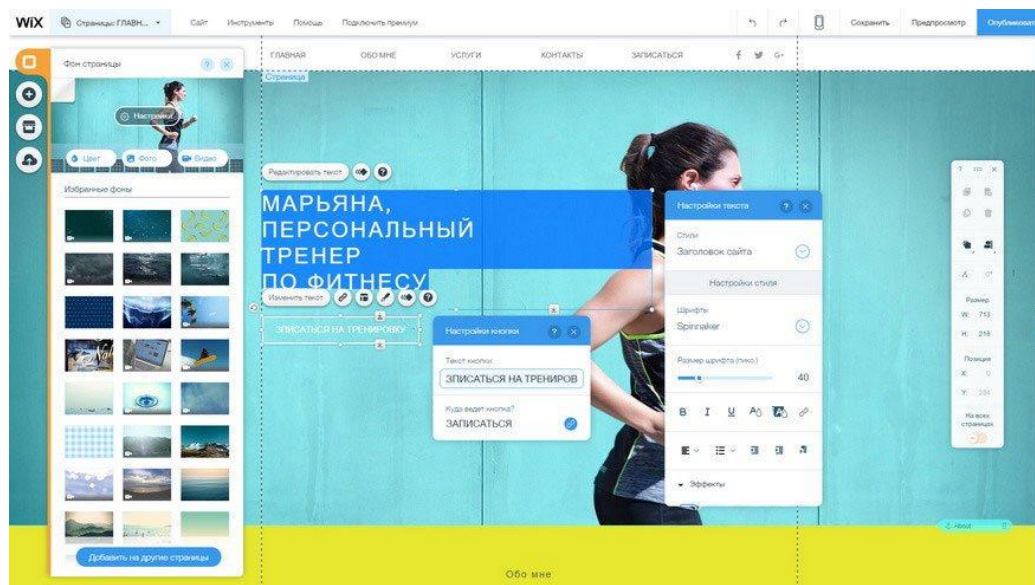


Рис.1.8. Швидке редагування у редакторі

Фотографії можна завантажити на сайт з Instagram, а в самій соціальній мережі залишити посилання на даний сайт, що важливо для власників акаунтів. Доповнює тему оздоблення сайту можливість завантаження власних шрифтів. За замовчуванням можна використовувати шрифти Google Fonts. Якщо цього виявиться замало, можна використовувати власні. Робиться це в розділі «Мої завантаження» - «Завантажити шрифти».

### Панель управління Wix

Серцем, основним командним пунктом сайту є панель управління Wix. У ній відображаються у вигляді плиток важливі функції, що доступні за замовченням, і всі встановлені додатки. Тут можна отримати доступ до всіх можливостей сайту і контролювати процес (рис.1.9).

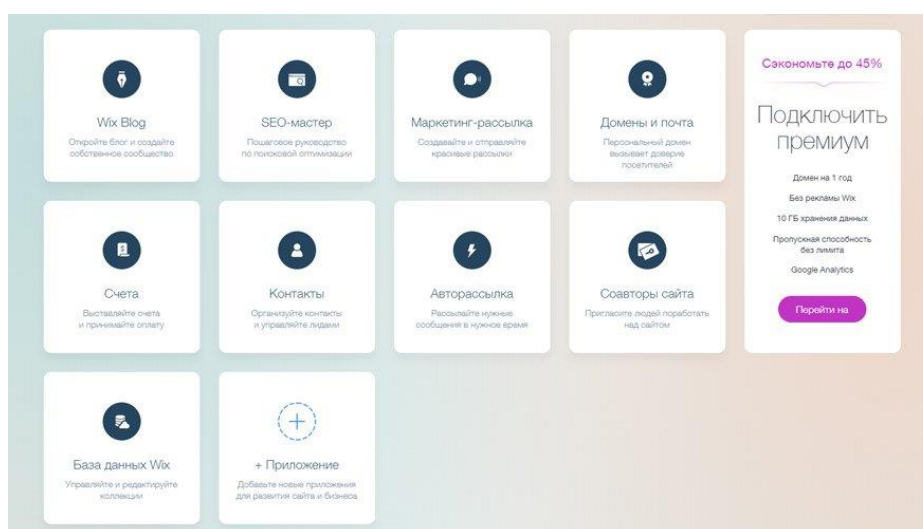


Рис.1.9. Панель управління Wix

**Wix Blog** – під'єднання блогу до сайту і його налаштування в редакторі.

**SEO-майстер** - комплексний інструмент для просування сайту.

**Маркетинг-розсилка** - створення новинних і рекламних розсилок з поштових баз за допомогою спеціалізованого додатку ShoutOut. Потрібно створити макет листа за допомогою налаштувань дизайну і вибрати поштову базу для розсилки (контакти можна імпортувати зі облікового запису Gmail або CSV-файлу). Можна розмістити банер в соціальній мережі. Можливими є запуск автоматичних розсилок за заданими умовами, вибір шаблонів для банерів і листів, прикріплення форми підписки на розсилку і контроль статистики по запусчених розсилках.

**Домени і пошта** - покупка нового або прикріплення існуючого домену до сайту, додавання корпоративної пошти зі всіма бізнес-інструментами від Gmail (пошта з назвою домену, пакет офісних додатків, сховище листів, відеоконференції).

**Рахунки** - створення і надсилання рахунків до клієнтів, прийом онлайн-платежів (дебетові та кредитні картки, PayPal), відстеження статусів платежів, звіти. Формат рахунків налаштовується, можна додавати податки, вибирати валюту і порядок нумерації.

**Контакти** - клієнтська база сайту. Може використовуватися для розсилання, збору інформації, підписок тощо. Є можливість поділення користувачів на групи, блокування, фільтрації вибірки і ручного створення нових контактів.

**Автоматичне розсилання** - налаштування опцій для автоматичного надсилання листів за різними сценаріями. За замовченням є шаблони для привітання нових контактів, зареєстрованих і тих, хто підписався на оновлення сайту. В такий спосіб можна надсилати файли, дякувати за покупки, пропонувати купони на знижку, запрошувати відвідати магазин повторно, просити залишати відгуки, вітати зі святами і подібне. Розклади, умови розсилок і оформлення вкладень можна гнучко налаштовувати.

**Співавтори сайту** - додавання через поштову адресу людей з правами адміністратора або менеджера бек-офісу. Для того, щоб допомагати вести і наповнювати сайт.

**Бази даних** - створення колекцій інформації, які будуть зберігатися в хмарі. Дані необхідно додавати в таблиці з заданою кількістю колонок і полів. Ці колекції можна підключати до різних елементів на сайті для збору інформації або навпаки - її видачі.

Нові додатки будуть додані в панель управління для швидкого і зручного доступу подібно до тих розділів, що перелічено вище. Така структура дозволяє легко орієнтуватися в великому функціональному потоці Wix. Більшість додатків оформлені і подані подібно до інтерфейсу конструктора - стилістично і в плані зручності використання. Володіючи інтерфейсом Wix, можна встановлювати значну кількість додатків без ризику заплутатися. Додавання нових функцій не ускладнить роботу над сайтом.

#### **Інші корисні можливості:**

- Можна безкоштовно під'єднати до сайту SSL сертифікат (захист сайту за https протоколом).
- Wix має мобільні додатки iOS / Android, що дозволяє вести живий чат з клієнтами сайту (для цього потрібно активувати додаток Wix Chat в панелі управління), додавати фотографії, товари або постити матеріали в блог.

- Можливість оптимізувати зображення для відображення в мобільній версії сайту. Потрібно зайти в редактор мобільного шаблону, вибрати потрібну картинку, обрізати, збільшити / зменшити початкове зображення, залишити ту частину, яка потрібна.
- Можна ставити на фон зображення 360° за допомогою додатку 360 Images.
- Можна захищати паролями окремі сторінки сайту або їх приховати.
- Завдяки сітці завжди видно область, яка відобразиться в мобільній версії сайту.
- В редакторі можна використовувати прошарки, приховувати окремі елементи і відновлювати їх за потребою.
- При активації режиму розробника з'являється панель властивостей елементів і консоль для додавання значень цих властивостей.
- Можна створювати копії одного сайту всередині акаунту або передавати сайт на інший рахунок.

Стоковий функціонал конструктора нерозривно пов'язаний з можливостями додатків з фірмового магазину. Їх можна розцінювати як модулі, які за замовченням не встановлено. В процесі роботи можна додавати потрібні додатки в будь-якій кількості. Додатки написані або розробниками самого Wix, або командами інших компаній.

## Дизайн і робота з шаблонами

Wix на поточний момент містить понад 1500 шаблонів, що відсортовані за категоріями: магазин, бізнес, фотографія, відео, музика, дизайн, ресторани, події, портфоліо, блог, краса і здоров'я, мода, мистецтво, лендинги. Після реєстрації на сайті, буде запропоновано вибрати дизайн для сайту. Можна фільтрувати збірку шаблонів за новизною або популярністю. Обраний шаблон не можна змінити в процесі роботи, тому варто підходити до вибору серйозно (рис.1.10).

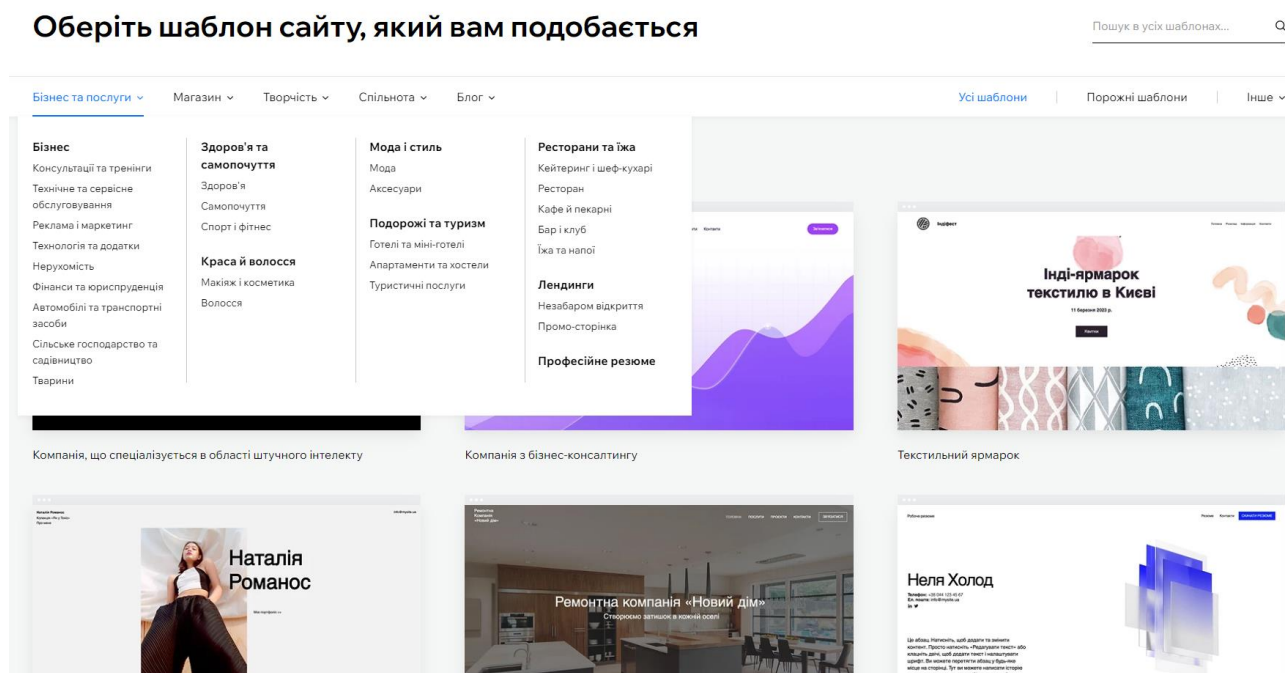


Рис.1.10. Вибір шаблону сайту

Wix містить досить великий набір готових шаблонів. Розробники періодично поповнюють бібліотеку дизайнів, на рік приходиться приблизно 30-60 нових зразків, які можна

використовувати безкоштовно. Кожен шаблон є адаптивним. Також, можна вибрати чистий аркуш і оформити кожну сторінку сайту з нуля в редакторі.

Неможливість зміни шаблону в процесі пояснюється абсолютним позиціонуванням елементів на сайтах Wix. Це надає свободу дій в оформленні дизайну, але позбавляє можливості експорту налаштувань і контенту в новий шаблон. Тобто, правило одного шаблону, обумовлено чисто технічними причинами.

### **Сильні та слабкі сторони конструктора**

Wix - багатоцільова машина для створення різноманітних сайтів. Завдяки якісним додатків і гнучкому редактору функціональні рамки дуже широкі, а з врахуванням потенціалу Wix Code вони практично зникають. Використання кодингу (HTML, JavaScript) можливо, але зовсім необов'язково. У міру простий, яскравий конструктор, зі значною кількістю корисних можливостей. При прямому порівнянні аналогів по частині функціональності не має.

Базовий функціонал Wix можна оцінити на вищому рівні. Підтвердженням якості служать більше 120 мільйонів сайтів, опублікованих користувачами в рамках Wix. Багато речей вперше в ніші з'явилися саме в цьому конструкторі.

Великі обсяги можливостей, міцне якість реалізації, стандартизація інтерфейсів додатків, креативна атмосфера, багато докладних довідкових матеріалів, текстових та відео (у Wix є канал на YouTube). Конструктор Wix постійно розвивається, тому, її розробники можуть собі дозволити інвестувати в нові функції.

#### *Переваги конструктора*

- **Часті оновлення.** Нові модулі, такі як Wix Code, ADI, інтегрований графічний редактор, додавання панорамних картинок і сінемаграфів, співавторів для сайту і багато інших приходять в нішу від Wix. Wix часто виступає законодавцем мод в світі конструкторів сайтів з візуальним редактором.
- **Магазин додатків.** Великий асортимент корисних речей, що виконані в єдиному стилі інтерфейсу з конструктором. Додатки сприймаються як нативні, але тимчасово під'єднані модулі сайту. З їх допомогою можна легко вирішити багато важливих і специфічних завдань.
- **Присмний інтерфейс.** Це проявляється всюди - в ергономіці і атмосфері редактора, меню, опціях, форматі панелі управління, прозорості окремих елементів, їх розмір та інше.
- **Оформлення дизайну.** Wix дозволяє створювати різноманітні, цікаві візуально сайти. Елементи на макеті не прив'язані один до одного (абсолютне позиціонування), їх можна комбінувати в будь-яких поєднаннях і позиціях.
- **Рентабельність.** Вартість тарифів Wix є прийнятною, у порівнянні з цінами конкурентів. На відміну від багатьох інших систем постійно надаються величезні знижки.

#### *Недоліки*

- **Не інтуїтивний.** Інтерфейс всіляко допомагає користувачеві, але потрібен час на звикання до нього і освоєння сценаріїв використання можливостей конструктора.

- **Відсутній імпорт товарів.** Не додано можливість завантаження товарів у магазин з CSV або інших табличних форматів.
- **Присутні обмеження.** На безкоштовному та дешевих тарифних планах передбачено обмеження пропускну здатності каналу. Тобто, за великої кількості медіафайлів на сторінках і середньому напливі відвідувачів сайт почне відчутно гальмувати чи виникне відмова у завантаженні.

### *Аналоги і конкуренти*

Прямої аналогів Wix не має, жоден інший конструктор з візуальним редактором не володіє навіть 70% можливостями Wix. Це відбувається завдяки Wix App Market – велика кількість модулів, включаючи форум, організацію готельного, ресторанного бізнесу, продаж цифрових товарів, бронювання чого-небудь, організація заходів тощо. Втілення штучного інтелекту ADI поки залишаються без аналогів з боку інших систем. Wix Code з базами даних і різноманітністю сценаріїв їх застосування.

### *Порядок роботи*

1. Ознайомитися з наданими теоретичними відомостями щодо безкоштовних конструкторів.
2. Обрати онлайн-конструктор для створення сайтів. Зареєструвати акаунт і ознайомитися з можливостями редактора. Зберегти ідентифікатори доступу, зокрема поштову адресу реєстранта, оскільки на неї надходять листи зі створеного сайту.
3. Визначитися щодо тематики та типу сайту. Відповідно до теми сайту віднайти доречний шаблон. Відкрити шаблон для редагування. Уважно дослідити можливості адміністративної панелі, наявні інструменти та їх функції.
4. Відповідно до теми сформувати інформацію: тексти і зображення. Визначитися з переліком розділів, колірною схемою, стилем подання інформації для різних типів пристроїв, наявністю мультимедійних об'єктів та зворотного зв'язку.
5. Користуючись інструментами адміністративної панелі або прямим редагуванням на сторінці, змінити назви розділів/підрозділів, додати/видалити відповідні сторінки. Наповнити сторінки сайту відповідною інформацією. Мова сайту - українська! Оздобити текст доречними зображеннями та мультимедійним контентом (відео, аудіо об'єкти, карти).
6. Обов'язковим є наявність авторської сторінки, де розміщується фотографія чи аватар студента, коротка інформація (Ім'я, нік, е-мейл тощо), карта із зазначенням домівки чи іншого місця.
7. Після завершення редагування зберегти зміни та отримати доменну адресу для опублікування сайту. Опублікувати сайт в Інтернеті і переглянути результати через браузер. За потребою внести зміни у сайт.
8. По результатах роботи оформити звіт.

### *Набір тем для сайту*

Для виконання лабораторної роботи студент створює сайт відповідної тематики, що містить 3-4 змістовних сторінок та сторінку автора сайту. Нижче наведено перелік тем, які можна використати або реалізувати власну.

- Портфолію робіт певного фахівця (дизайнер, програміст, тренер тощо).
- Послуги умовної компанії (бізнес, медицина, розваги, готелі, кафе тощо).
- Візитка відомої особи: письменника, спортсмена, актора, співака, музичного гурту, політичного діяча.
- Галерея творчих робіт: фотографії, картини, книги, пісні чи інше.
- Новинки технічних досягнень: комп'ютери, телефони, автомобілі, роботи.
- Міста та країни світу. Улюблені краєвиди, історичні екскурси, цікаві відомості.
- Хобі студента: спорт, творчість, навчання, домашні тварини та інше.
- Паблік для популярного фільму, серіалу, мюзиклу чи творчої спільноти.
- Електронна бібліотека із збіркою певних книжок, статей, коміксів тощо.
- Домашня сторінка студента, де викладено його захоплення, мандрівки, фотографії друзів, домашніх тварин та інша відкрита інформація.

## *Зміст звіту*

1. Назва та мета виконання лабораторної роботи.
2. Вказати доменну адресу, назву і короткий опис створеного сайту.
3. Скріншоти послідовності створення сайту з коротким описом кожного з етапів.
4. У висновку оцінити зручність використання конструктора і адміністративної панелі.

## *Контрольні питання*

1. Перелічити популярні конструктори сайтів і зазначити сервіси, які надають можливості безкоштовного користування.
2. Вказати особливості конструкторів, зокрема можливості реалізації індивідуальних змін.
3. Навести перелік позитивних характеристик обраного конструктора, а також відмітити обмеження стосовно певних функціональностей.
4. Яким чином можна отримати хостинг та доменну адресу від конструктора?
5. Назвати основні причини, за якими користувач може обрати створення сайту за допомогою конструктора.

## *Використані джерела*

1. Конструктор WIX. - <https://uk.wix.com/>
2. Wix.com у Facebook. - <https://www.facebook.com/wix/>
3. Кращі безкоштовні конструктори сайтів і Інтернет-магазинів. - <https://project-seo.net/uk/blog-uk/krashchi-bezkoshtovni-konstruktory-saytiv-i-internet-magazyniv-2019/>
4. Як створити свій сайт: керівництво для початківців - <https://ru.wix.com/blog/2021/08/kak-sozdat-sayt-polnoye-rukovodstvo>
5. Дизайнер сайтів на WIX - <https://www.wix.school/>
6. ТОП конструкторів лендингів, порівняння, рейтинг і огляд - <https://smartlanding.biz/obzor-konstruktorov-landing-page.html>

## Лабораторна робота №2. Базовий інструментарій

**Мета роботи.** Ознайомитися з безкоштовними базовими засобами для створення простої HTML сторінки. Встановити на комп'ютері необхідні програми і набути практичних навичок з користування. Ознайомитися з безкоштовними майданчиками для хостингу веб-проектів. Набути навичок з користування панеллю керування, засобами передачі за FTP-протоколом, налаштування параметрів хоста та доменного імені.

### Базовий інструментарій

Створення сайту - це трудомісткий і тривалий процес, який відбувається в кілька етапів, в міру проходження яких ідеї перетворюються в реальний функціонуючий сайт.

Для того, щоб розробник відчував себе максимально комфортно при розробці нового проекту, окрім робочого місця і персонального комп'ютера йому необхідно мати певний набір програмного забезпечення, базовий інструментарій розробника сайту.

Досвідчені розробники користуються професійними програмами, що розроблено відомими компаніями і за їх використання доводиться платити чималу ціну. Для початківців популярною альтернативою є безкоштовні редактори, які, можливо, і мають меншу функціональність, але є цілком придатними для роботи над веб-проектом.

Розробнику-початківцю слід орієнтуватися у різноманітті найменувань та призначень існуючих програм, і встановити необхідний набір інструментів.

- Зареєструвати хостинговий майданчик, для фізичного розміщення сайту в Інтернеті.
- Зареєструвати доменну адресу, за якої сайт буде відомий для користувачів та пошукових систем.
- Встановити на комп'ютері відповідні програми – базовий інструментарій веб-розробника:
  - Редактори для написання коду або інтегровані середовища розробки IDE.
  - Поширені браузері для наочного перегляду результатів кодування.
  - Програми з FTP-клієнтом для перенесення документів сайту на хостинг.
  - Програми для віддаленого управління та колективної роботи.

### Веб-хостинг

**Веб-хостинг** - це фізичне розміщення веб-сторінок на сервері. Це віртуальний аналог оренди приміщення, але орендується місце на диску, яке обчислюється мегабайтами. Від того, де буде розміщено сайт, залежить багато якісних характеристик, тому, важливо вибрати оптимальний майданчик для сайту, що відповідає критеріям надійності та стабільності.

Хостинг умовно можна поділити на безкоштовний і платний. Для початківців, які починають перші кроки в області веб-розробки вірним рішенням буде реєстрація майданчика на безкоштовному хостингу.

### Плюси безкоштовного хостингу

- **Не потрібно платити.** Розробник початківець тільки починає робити перші кроки і мало обізнаний в нюансах розміщення, показниках якості хостингу, налаштуваннях панелі та

акаунту. Тому, варто, безкоштовно опанувати прийоми і зауважити ті критерії, які важливі для ефективної роботи: простота налаштувань, зручність адміністративної панелі, наявність вбудованих редакторів та файлових менеджерів.

- **Безкоштовний тариф пропонує послуги аналогічні до платних.** Надаються всі основні можливості (з певними функціональними обмеженнями), що потрібні для розміщення сайту: дисковий простір, підтримка мов програмування, під'єднання до баз даних, встановлення поширеної CMS - системи управління контентом, створення поштового акаунту та інше.
- **Технічна підтримка.** Користувачам надаються конструктори та інші інструменти, що полегшують створення, підтримку та просування сайту. Повсюдно присутня довідкова система, яка охоплює багато загальних та специфічних проблем, з якими стикаються початківці. Наявність спільнот так само містить багато форумів та обговорень, де можна завжди з'ясувати шляхи вирішення поширених проблем.
- **Можливість отримання безкоштовної доменної адреси 3 рівня.** Безкоштовні хостинги надають можливості для вибору будь якого під домену до основного домену хостингу. Початківець позбувається проблеми щодо реєстрації, оплати та підтримки доменної адреси.

#### Обмеження безкоштовного хостингу

- **Обмежений дисковий простір**, але цілком достатній для розміщення нескладного сайту. На сьогодні цей показник складає від 300Мб і більше.
- **Немає гарантій щодо захисту та доступності сайту.** На таких площадках не роблять резервні копії сайтів і тому, у разі збоїв серверів сайт може бути пошкоджений або знищений без можливості відновлення. Створення резервних копій є турботою власника сайту.
- **Деякі безкоштовні хостинги мають дохід за рахунок реклами.** Де, скільки і за якою тематикою буде реклама на сайті користувача вирішують власники сервісу. Як правило, заборонено розміщувати власну рекламу і будь-яку іншу комерційну інформацію.
- **Можуть бути присутні обмеження на розміщення файлів певного формату** (аудіо, відео, скрипти), а також на просте складування документів.
- При явній популярності сайту власники хостингу можуть висунути **умови перенесення ресурсу на платний тариф**, погрожуючи знищенням сайту. Прикрий випадок, коли вкладено багато часу, сил і фінансових коштів в створення сайту, в наповнення контенту і просування. Закрити сайт, коли він вже став популярним і має чималу аудиторію відвідувачів не завжди просто.

#### Поради для вибору безкоштовного хостингу

- Віддавати перевагу більш популярним сервісам, бажано з давньою історією та хорошою репутацією.
- Заздалегідь почитати відгуки про даний хостинг в Інтернеті: чи достатньо там функціональних можливостей, наскільки зручно користуватися цим сервісом (способи редагування файлів і завантаження на сервер, показники надійності та функціональності хостингу).



На сьогодні розробникам початківцям надається великий вибір безкоштовних майданчиків, як українських так і іноземних. Вони дещо різняться за своїми показниками та умовами розміщення, але в загальному мають приблизно однаковий набір функціоналу.

- Free Web Hosting <https://www.000webhost.com/>
- Byet Host <https://byet.host/>
- Award Space <https://www.awardspace.com/>
- Free Hosting EU <https://www.freehostingeu.com/>
- Hosting 1Gb <https://www.1gb.ua/>

## *Вибір доменного імені (адреси)*

Перше, з чим знайомиться потенційний відвідувач ще до відвідування сайту – це доменне ім'я. Нижче наведено чинники, які потрібно враховувати при виборі доменного імені сайту, що впливають як на сприйняття людиною, так і на сприйняття пошуковими системами.

- **Легко запам'ятовується.** Користувачі рідше вводять в адресний рядок назву сайту, зазвичай переходять за посиланнями або вводять назву сайту в рядок пошукової системи. Тому, варто дотримуватися правила «чим коротше назва, тим її легше запам'ятати».
- **Легко вимовляється.** Доменне ім'я має легко вимовлятися, так відвідувачам буде простіше його запам'ятати. Варто уникати використання в імені дефісів та вживання неоднозначної транслітерації: х (h, kh), ц (ts, z), в (w, v), к (с, k, ck), ф (f, ph). Наприклад, доменні імена foto.ua і photo.ua звучать абсолютно однаково – фото, але написання літер є різним.
- **Має зрозумілий сенс.** Ідеально, коли за назвою зрозуміло, яка тематика сайту. Якщо просувається бренд, то домен варто назвати так само, якщо це особистий блог - то можна назвати сайт за іменем чи прізвищем автора.
- **Просте написання.** Чим довше і складніше доменне ім'я - тим важче буде відвідувачам правильно його написати. Коректне введення довгого доменного імені може виявитися занадто складним. Наприклад, багаторазовий повтор складів як ababahalataha.com.ua, в якому кількість повторень запам'ятати складно.

## *Програми для кодингу*

### **Текстові HTML-редактори**

Створення сайту в текстових редакторах є доцільним, якщо розробник знайомий з мовою HTML і володіє нею на належному рівні. Є значна кількість текстових HTML-редакторів, що використовуються розробниками різного рівня фаховості, втім, писати код можна і в звичайному блокноті, який є на кожному комп'ютері.

До позитивних особливостей текстових редакторів потрібно віднести те, що розробник власноруч пише код всіх сторінок майбутнього сайту. Зазвичай, в багатьох текстових редакторах є і підсвічування синтаксису, і уставляння стандартних фрагментів коду, і багато інших корисних функцій, але редактор не додає жодного рядка без відома розробника, на відміну від візуальних редакторів. Тому, код сайту буде зрозумілим і позбавлений

надлишковості. Розробник точно буде знати призначення того чи іншого фрагмента html-коду і за потреби, може легко внести зміни. До популярних безкоштовних редакторів можна віднести:



Notepad++ (<https://notepad-plus-plus.org/>)

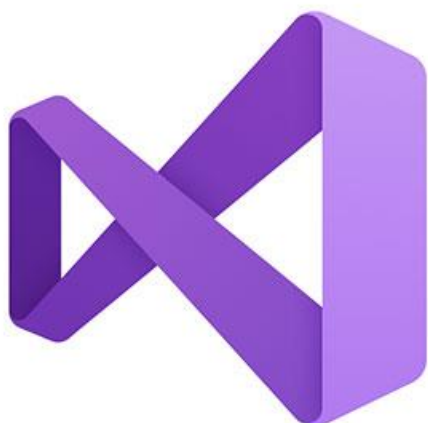


Sublime text (<https://www.sublimetext.com/>)

### Інтегровані середовища розробки

Інтегроване середовище розробки (IDE, Integrated Development Environment) — комплексне програмне рішення для розробки програмного додатку. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду.

Сучасні інтегровані середовища розробки пропонують значно більше можливостей, ніж описаний вище необхідний мінімум. Наприклад, багато сучасних IDE є візуальними - вони дозволяють створювати інтерфейс програми в такому вигляді, в якому він надається користувачеві



Visual Studio  
(<https://code.visualstudio.com/>)



Adobe Dreamweaver  
(<https://www.adobe.com/ua/products/dreamweaver.html>)

## FTP-клієнти

Невід'ємною частиною хостингу є FTP-сервер. Після отримання акаунту на хостингу, доступ до сервера здійснюється, зазвичай, за протоколом FTP. Для здійснення віддаленого управління власними ресурсами існує два типи доступу до хостингу:

- Доступ через адміністративну панель хостингу, яка надається хостинг-компанією для розробників.
- Доступ за допомогою програм з вбудованим FTP-клієнтом.

FTP-клієнт - це утиліта, яка спрощує доступ користувача до FTP-серверів. FTP-клієнтом може бути файловий менеджер, браузер або спеціалізований додаток. Вибір засобу залежить від завдань і уподобань користувача.

Незалежно від того, як відбувається зв'язок з FTP-сервером потрібно авторизуватися і ввести ті ідентифікатори, що надає хостинг: адресу FTP-сервера, назву протоколу, ім'я користувача і пароль. Ці дані хостер висилає на пошту після активації акаунту.

До популярних програм з вбудованим FTP-клієнтом можна віднести:



Filezilla (<https://filezilla-project.org/>)



Total Commander (<https://www.ghisler.com/>)

Практично у всіх файлових менеджерах реалізовано підтримку FTP. Всі операції з файлами та папками на віддалених серверах здійснюються аналогічно, як на локальному комп'ютері. Функціональності менеджерів цілком достатньо для вивантаження файлів з сервера та завантаження їх на сервер, а також для віддаленого перейменування, видалення або редагування файлів/папок.

Функціональність спеціалізованих FTP-клієнтів, зазвичай, є значно ширшою - вони забезпечують вивантаження/завантаження файлів в більш безпечному режимі (за рахунок підтримки відповідних протоколів, різних варіантів шифрування тощо) з більшою швидкістю. Застосовується багатопотоковість завантаження і одночасного стиснення файлів.

Процес вивантаження/завантаження файлів є зручнішим, оскільки в них реалізовані засоби швидкого доступу до часто відвідуваних FTP-серверів, можливим є під'єднання до кількох серверів одночасно, передбачено роботу за розкладом тощо.

## Браузери

Основним призначенням браузера є відображення веб-ресурсів. За виведення вмісту сторінки у вікно браузера відповідає важливий елемент - модуль відображення. За замовченням модуль відображення здатний відображати HTML-документи, зображення, а також багато об'єктів в різних форматах: PDF-файли, мультимедійні файли, динамічні об'єкти.

Значна частина відмінностей браузерів випливає з відмінностей у модулі відображення сторінок. Ці відмінності приводять до того, що сторінки в різних браузерах виглядають дещо по-різному.



**Trident** - розроблено компанією Microsoft. Використовується в багатьох браузерах: Internet Explorer, Windows Mobile, AOL Browsers



**EdgeHTML** - браузерний модуль від Microsoft, який втілено в Microsoft Edge. Є гілкою від Trident з видаленням застарілого коду і заміною його на більш сучасний. Вперше реалізований в Internet Explorer 11 як частина Windows 10 Preview 9926 build.



**Gecko** - движок з відкритим вихідним кодом, який було розроблено ще за часів браузера Netscape, але почав використовуватися тільки з браузера Mozilla Firefox.



**WebKit** - движок з відкритим кодом, він розроблявся такими компаніями як Google, Apple, Nokia та іншими. Спочатку було розроблено для платформи Linux і адаптовано компанією Apple для Mac OS і Windows. Використовується в браузері Safari від Apple, і в багатьох браузерах мобільних пристроїв.



**Blink** - заснований на WebKit, що розробляється Google. Використовується браузером Google Chrome, а також браузерами Opera, і браузерами для мобільних пристроїв на Android.

Браузер надсилає до сервера запит, щоб отримати певну сторінку веб-сайту і відобразити на екрані користувача. Веб-сервер приймає запит, шукає запитану сторінку і надсилає до клієнта її вміст або повідомлення про помилку, якщо такої сторінки не знайдено чи доступ до неї є забороненим. Веб-сервер не аналізує вміст сторінки, яку передає. Всю роботу зі структуризації і відображення отриманої інформації виконує браузер (рис.2.1).

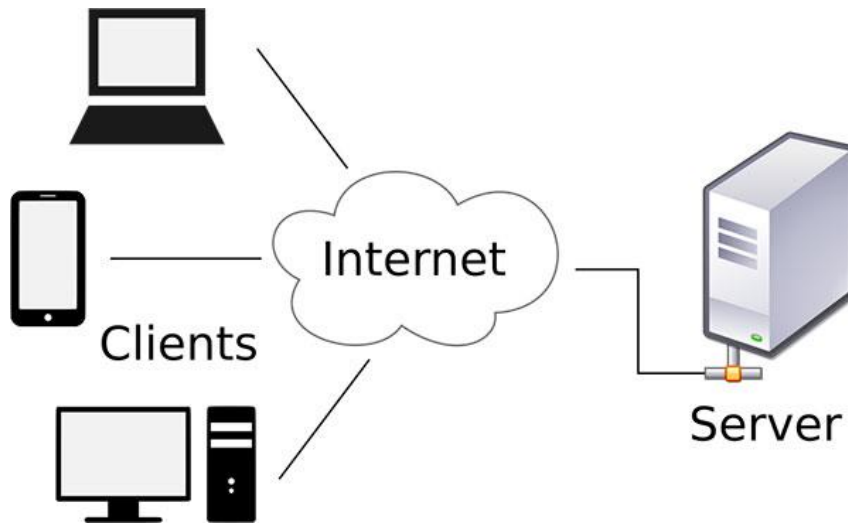


Рис.2.1. Взаємодія клієнта (браузера) і сервера

### Обробка сервером запиту від браузера

Пошук потрібної сторінки здійснюється в певній директорії, яку відведено на сервері для сайту (рис.2.2). Зазвичай, назва директорії збігається з назвою доменної адреси сайту. У разі звернення до сайту в цілому (наприклад, *site.ua*), веб-сервер автоматично надсилає «стартову сторінку» з назвою *index.\** (*index.html, index.php*) або в деяких випадках — *default.\** (*default.html*).

Віддалений сайт: /home/eom/public\_html

Ім'я файлу	Розмір ф...	Тип файлу	Востаннє мод...	Дозволи	Власник/Г...
usr		Папка фай...	02.07.2017 20:0...	drwxr-xr-x	eom eom
.htaccess	117	Файл НТА...	02.07.2017 19:2...	-rw-r--r--	eom eom
ajaxcaptcha.php	470	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom
ajaxmail.php	1 600	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom
ajaxresponse.php	448	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom
blank.php	0	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom
captcha.php	742	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom
cautresponse.php	5 287	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom
fun.php	45	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom
fun_html.php	21 085	Файл PHP	15.01.2020 11:3...	-rwxrwxrwx	eom eom
<b>index.php</b>	209	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom
robots.txt	67	Текстовий ...	02.07.2017 19:2...	-rw-r--r--	eom eom
sitemap.php	4 615	Файл PHP	02.07.2017 19:2...	-rw-r--r--	eom eom

Рис.2.2. Фізична структура сайту eom.lp.edu.ua

Цей документ повинен розташовуватися в кореневій директорії, яку відведено для розміщення сайту. Решта файлів можна розміщувати або в цій же директорії, або у вкладених директоріях,

що іноді буває зручним, особливо у разі, коли сайт містить кілька тематичних розділів або рубрик.

Деякі веб-сервери на відміну від локальних комп'ютерів є чутливими до регістру літер, в якому представлено імена директорій і файлів, тому, назви «documents», «DOCUMENTS» або «Documents» будуть позначати різні директорії. Варто назви всіх об'єктів сайту писати рядковими (маленькими) літерами.

Щоб уникнути помилок, які пов'язані з відмінностями в обробці кодувань кириличної мови різними серверами, назви слід зазначати латиницею.

Для безпомилкового отримання потрібної інформації і в потрібному форматі використовується універсальний покажчик ресурсу URL (Universal Resource Locator), який однозначно ідентифікує будь-який ресурс в мережі Інтернет. Саме такий рядок відображається в адресному полі браузера.

<https://www.site.lviv.ua/documents/page.html>

<http://213.82.46.1/documents/page.html>

Універсальний покажчик ресурсу відображає:

- **Протокол відповідної служби.** В даному прикладі використано протокол `https://` – протокол передачі гіпертексту.
- **Назву служби.** В даному прикладі це служба Веб – `www`.
- **Доменну або IP-адресу,** яка однозначно ідентифікує веб-сервер в мережі Інтернет, на якому розміщено потрібний сайт чи інший ресурс - `site.lviv.ua`.
- **Шлях,** що складається з імен директорій, розділених символом «/» (слеш), послідовно відкриваючи які, можна «дістатися» до потрібної інформації. У даному прикладі інформація знаходиться в директорії «documents».
- **Ім'я файлу,** який містить потрібну інформацію. В цьому прикладі інформація знаходиться у файлі `page.html`.

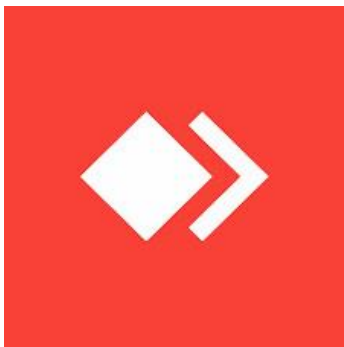
Якщо браузер використовують для служби Веб, то в адресному полі можна не вказувати назву протоколу і назву служби. Хоча іноді, залежно від налаштувань, веб-сервер скеровує запити `www.site.lviv.ua` та `site.lviv.ua` в різні директорії.

Отже, для доступу до створеної сторінки за доменною адресою сайту, потрібно, щоб її назва була `index.*`, в протилежному випадку для доступу буде потрібно вказувати повний шлях до файлу `http://yourdomenadress/page.*`.

## **Програми для віддаленого управління**

Програми віддаленого управління - це засоби, що з'єднують комп'ютер користувача (адміністратора) з різними віддаленими комп'ютерними пристроями по всьому світу. За допомогою цих програм можна вирішувати технічні проблеми та використовувати ресурси комп'ютерів на відстані.

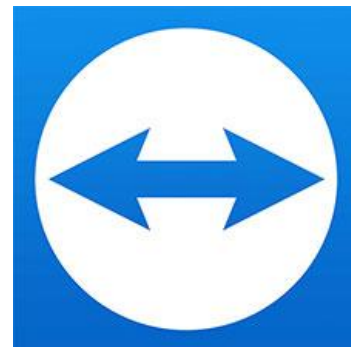
Після отримання доступу до віддаленого пристрою, ним можна керувати, як ніби він знаходиться поруч: змінювати налаштування, запускати встановлені додатки, доступатися до вмісту дисків, переглядати, редагувати і копіювати файли, бачити дії користувача та багато іншого. До популярних програм можна віднести:



Any Desk  
(<https://anydesk.com/>)



Ammyy Admin  
(<https://www.ammyy.com/>)



TeamViewer  
(<https://www.teamviewer.com/>)

Завдяки програмам віддаленого доступу учасники робочої групи можуть ефективно працювати в команді, перебуваючи на відстані один від одного. Розробники мають можливість взаємодіяти в будь-який час, за допомогою будь-яких пристроїв і на значній відстані.

### *Створення простої сторінки*

Після створення веб-хостингу і доменного імені можна приступати до безпосередньої розробки сайту. Потрібно визначитися з підходами щодо створення:

**Підхід 1:** створення сайту за допомогою конструкторів ([лабораторна робота №1](#)).

**Підхід 2:** створення сайту за допомогою CMS. ([лабораторна робота №7](#)).

**Підхід 3:** створення веб-сайту з нуля. Знання HTML, CSS, JavaScript, мови серверного програмування та бази даних.

- **HTML (Hyper Text Markup Language).** Основна структурна мова для веб-сторінок і веб-додатків, яка робить контент зрозумілим для веб-браузера.
- **CSS (каскадні таблиці стилів).** Стили, які використовуються для оформлення розмітки HTML на веб-сторінці.
- **JavaScript (клієнтська мова).** Використовується для додавання динаміки елементів на сторінці. Цю мову розуміє будь-який браузер і вона використовується як для динамічних так і для статичних сайтів
- **PHP, Java, Python, Ruby та інші (мови серверного програмування).** Мови написання різноманітних сценаріїв для складних, динамічних сайтів. Файли сценаріїв задають систему поведінки сайту, реакції на дії користувача, формування сторінок, відповідно до запиту і подібне.
- **MySQL, MongoDB, PostgreSQL та інші (системи управління базами даних).** Великі таблиці інформації, що використовуються для зберігання, управління і доступу до користувацьких даних на веб-сайт. Система управління базами даних використовується на стороні сервера для ефективного виконання завдань, що прописані в сценаріях.

### *Онлайн сервіси, генератори, редактори*

Секрет швидкого навчання полягає в тому, що для того, щоб створити достатньо функціональний сайт, не треба знати весь HTML - там багато нюансів, які можна опустити.

Більшість книг по створенню сайтів з гучними назвами і об'ємними томами, де розписано весь курс HTML, більше годяться на роль довідників, а не підручників. В даному курсі лабораторних робіт обмежимося лише необхідною частиною, яка дозволить самостійно створювати працюючі сайти. Решта знань набуваються під час створення наступних, більш складних проектів.

Для допомоги розробникам-початківцям існує багато онлайн-сервісів, які надають певні елементи (код, структуру, зображення, елементи), що в кінцевому результаті мають належну якість.

#### Переваги онлайн-сервісів:

- Не потрібно встановлювати на комп'ютер дороге ліцензоване програмне забезпечення, достатньо просто відкрити вікно браузера.
- Відпадає проблема сумісності між тією чи іншою програмою і операційною системою на комп'ютері користувача.
- Немає необхідності відслідковувати оновлення, сервіс онлайн - це завжди актуальна версія програми.
- Онлайн-сервіс доступний завжди і скрізь, він не прив'язаний до конкретного комп'ютера.

#### Генератор структури HTML-сторінки CreateTemplater (<http://createtemplate.hellohtml.ru/>)

Генератор HTML та CSS шаблонів (макетів) - це онлайн-сервіс, який генерує архів, що містить 2 файли: HTML і CSS, які мають базову розмітку (каркас) для майбутнього сайту (рис.2.3).

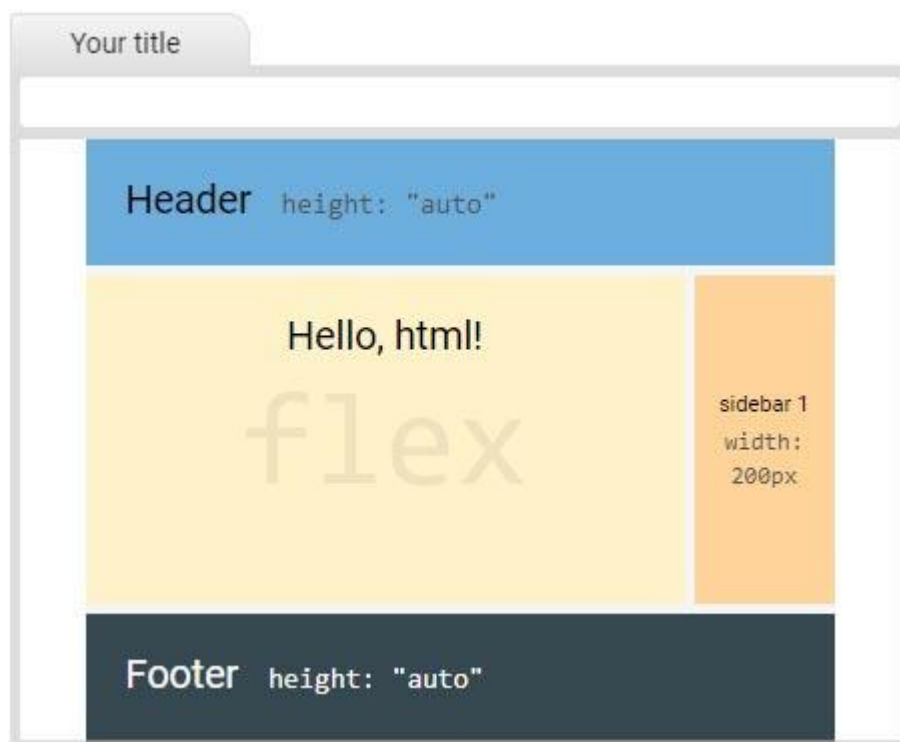


Рис.2.3. Інтерфейс онлайн-сервісу CreateTemplater

Сервіс може бути корисний для розробників, які мають деякі навички роботи з технологіями HTML і CSS, але створення базової структури сторінки (що складається з таких елементів, як "шапка", колонка з контентом, бічні колонки (сайдбар) і "підвал") викликає певні труднощі.



У всіх макетах використовується тільки блокова верстка. Макети, що пропонуються Генератором HTML та CSS шаблонів, протестовано і вони коректно відображаються в сучасних браузерях.

### **Онлайн редактори HTML, CSS, JS**

- Codly <https://codly.ru/editor>
- Codepen <https://codepen.io/pen/>
- JsFiddle <https://jsfiddle.net/>
- Jsbin <https://jsbin.com/>
- Htmlg <https://htmlg.com/html-editor>
- Froala <https://froala.com/online-html-editor>
- Schoolsw3 <https://schoolsw3.com/>

### **Генератори CSS-кода**

- {CSS} Portal <https://www.cssportal.com/>
- CSS3 Generator <http://www.css3generator.com/>
- Gradient Generator <https://www.colorzilla.com/gradient-editor/>

### ***Фізична структура статичних сайтів***

Під фізичною структурою проекту розуміють порядок зберігання файлів проекту в його директорії. Часто доводиться бачити, коли всі файли знаходяться разом, назви файлів зазначено комбінацією малих та великих літер, загальними назвами типу page1, page2, цифрами, кирилицею тощо.

Це неприпустимо, по-перше, це неповага до фахівців, які будуть працювати з проектом в подальшому, по-друге, чим більше буде проект, тим більше буде ставати файлів і, зрештою можна заплутатися, що до чого відноситься і що потрібно, а що ні.

Зазвичай, в корені сайту розташовують html-документи, що відповідають кожному розділу. Сторінки підрозділів на розсуд розробника можуть міститися або в корені або рознесені до різних відповідних папок.

Всі зображення, що використовуються у сторінках, прийнято зберігати в окремих папках. Зображення, що використовуються в оформленні зовнішнього вигляду сторінок традиційно розміщують у папку з назвою images або img. Це сприяє стрункій систематизації файлів і є зрозумілим для розробників, що працюють над проектом. Аналогічно, для скриптів можна створити папку script, а для стилів style.

Зображення, що відносяться до контенту певних розділів, наприклад, збірка фотографій за різні роки, то для них можна зробити окремі папки, але із змістовними назвами, наприклад, photo/2022/summer/

Дотримуючись цих правил структура проекту буде чіткою і зрозумілою (рис.2.4).

Веб-сайт - Youbook (D:\web\You-books\y...	Папка
css	Папка
bootstrap.css	151KB Файл "CSS"
materialdesignicons.css	10.7KB Файл "CSS"
mystyle.css	13KB Файл "CSS"
fonts	Папка
materialdesignicons-webfont.eot	286KB Файл "EOT"
materialdesignicons-webfont.svg	2.11MB SVG-документ
materialdesignicons-webfont.ttf	286KB Файл шрифта TrueType
materialdesignicons-webfont.woff	141KB Файл "WOFF"
materialdesignicons-webfont.woff2	10.7KB Файл "WOFF2"
images	Папка
counter.jpg	3KB ACDSee Pro 3 JPEG Image
favicon.ico	2KB Значок
favicon.png	2KB ACDSee Pro 3 PNG Image
logo.png	7KB ACDSee Pro 3 PNG Image
up.png	2KB ACDSee Pro 3 PNG Image
js	Папка
app.js	8KB Файл сценария JScript
back-to-top.js	4KB Файл сценария JScript
bootstrap.js	58KB Файл сценария JScript
bootstrap.min.js	28KB Файл сценария JScript
jquery-1.8.2.min.js	92KB Файл сценария JScript
jquery.sticky.js	5KB Файл сценария JScript
modernizr.custom.js	9KB Файл сценария JScript
picture	Папка
index.html	62KB Файл "HTML"

Рис.2.4. Фізична структура статичного сайту

Існують також правила щодо найменування файлів проекту. Найчастіше застосовуються такі імена: головна сторінка - це `index.html`, стилі проекту `styles.css` або `*.css`, скрипти `scripts.js` або `*.js`, мінімізовані версії файлів мають префікс `.min` (`scripts.min.js`), картинки мають не розлогі назви або набори цифр, а відображають те, що на них зображено, наприклад, `button.png`, `download-icon.png`, `logo.png` тощо.

У складі сайту мають бути присутні файли, які можна віднести до обов'язкових та службових. Зазвичай, їх розміщуються в корені сайту. Це файли, що призначені для виконання певних завдань і мають стандартну назву, а також файли, назви яких є довільними і визначаються розробниками.

#### Обов'язкові файли

`index.html` (`index.*`) — за замовченням є файлом головної сторінки, а також веб-сторінок, що розміщені в папках і повинні відкриватися при їх вказуванні в адресі. Назву і розширення індексового файлу, можна змінити в залежності від технології створення сторінок, типу веб-сервера та його налаштувань.

`favicon.png` (`favicon.*`) — іконка сайту у вигляді зображення у форматі `.png` або `.svg`. Вона з'являється в браузері біля адреси сайту в адресному рядку, у вкладці браузера біля назви сайту, у сторінці пошукової системи біля короткого опису сайту.

`style.css` чи `**css` – файл з таблицею стилів. Жоден з сучасних сайтів не обходиться без стилів, які зазвичай, розміщуються в окремому файлі. Таке розміщення надає кілька переваг — можна легко змінити вигляд елементів на всіх сторінках, за рахунок зміни параметрів лише в одному

місці, файл кешується і завантажується швидше, зменшується загальний об'єм всіх документів. Файл зі стилями має розширення .css.

## **Поради**

Для того, щоб всі посилання у сайті працювали коректно, документи відкривалися правильно і браузер не видавав помилок при зверненні до певних розділів ресурсу, при створенні його фізичної структури слід дотримуватися кількох простих правил.

- Призначати назви папок, назви і розширення всіх файлів сайту з використанням символів лише латинського алфавіту і лише в рядковому регістрі. Такий підхід гарантує універсальність і працездатність на різних платформах.
- Не використовувати в назвах файлів пробіли, це може викликати помилки в деяких операційних системах. Для візуального розділення змістовних частин назви краще використати дефіс або символ підкреслення, наприклад, about-us.html, my\_cat.jpg.
- При привласненні назв файлів прагнути, щоб вони були змістовними. Це буде корисно і під час створення сайту і згодом, коли потрібно буде внести певні зміни у сайт, а також для інших розробників, які працюють над сайтом. Важко орієнтуватися серед сторінок, що позначаються як 1.htm, 2.htm, 3.htm тощо.
- На сьогоднішній день популярними стають назви - транслітерація латинськими літерами, наприклад, rozklad-dlya-studentiv.html або mij\_kotyk\_bavytsia.png. Це є додатковою інформацією для пошукових систем, які навчилися інтерпретувати транслітерацію в національні символи і розуміти сенс назви.

## ***Безкоштовні шаблони дизайну***

- OS Templates <https://www.os-templates.com/>
- Free html5 <https://freehtml5.co/>
- W3Layouts <https://w3layouts.com/>
- HTML5 UP <https://html5up.net/>
- Free CSS <https://www.free-css.com/>

## ***Порядок роботи***

1. Ознайомитися з інформацією стосовно базового інструментарію та початкових знань розробника. Встановити потрібні для роботи програми.
2. Обрати відповідну безкоштовну хостингову площадку та доменне ім'я. Отримати ідентифікатори для FTP-доступу. Встановити та налаштувати програму FTP-клієнт.
3. За допомогою генератора структури створити початковий макет сторінки. Ознайомитися із згенерованими тегами та стилями.
4. Створити сторінку, наближену до наведених зразків

## Лабораторна робота №3

### Створення простої сторінки



[Сайт Львівської політехніки](#)



[Сайт кафедри САПР](#)

#### Інститут комп'ютерних наук та інформаційних технологій

- АСУ Кафедра автоматизованих систем управління
- ПІ Кафедра прикладної лінгвістики
- САПР Кафедра систем автоматизованого проектування
- ІТВС Кафедра інформаційних технологій видавничої справи
- ІСМ Кафедра інформаційних систем та мереж
- ПЗ Кафедра програмного забезпечення
- СПІ Кафедра систем штучного інтелекту

## Лабораторна робота №3

### Створення простої сторінки

Група КН-204

[Сайт кафедри САПР](#)

[Сайт ІКНІ](#)

[Сайт Львівської політехніки](#)

1. *Високий рейтинг*
  1. Василів Василь
  2. Гавриленко Олена
  3. Петренко Максим
2. *Середній рейтинг*
  1. Іванченко Світлана
  2. Коляда Сергій
  3. Лобода Микита



## Лабораторна робота №3

### Створення простої сторінки



**Бенгальський породи котів** вдалося поєднати в собі ніжність домашніх виконавців і грайливість і мисливські темпераменти диких звірів. Найлюбленіша їхня гра – це гонитва за м'ячиком, іграшками або комахами, які залетіли до квартири. До таких тварин потрібно ставитися з особливою любов'ю, інакше вони можуть вирости дуже дикими. Однак, якщо кіт бенгальської породи полюбить вас, він буде відданий вам до кінця життя.

Країна і рік походження	США, 1961 рік
Тип хутра	короткошерста
Розмір	середній
Середня тривалість життя	12-15 років

## Лабораторна робота №3

### Створення простої сторінки

#### 10 порід собак з неймовірним інтелектом



1. [Бордер-коллі](#)
2. [Пудель](#)
3. [Німецька вівчарка](#)
4. [Золотистий ретривер](#)
5. [Доберман-піччер](#)
6. [Шотландська вівчарка](#)
7. [Лабрадор-ретривер](#)
8. [Папильон](#)
9. [Ротвейлер](#)
10. [Австралійська вівчарка](#)

## Лабораторна робота №3

### Створення простої сторінки

#### Погода у Львові

		Т, °С	Атм. тиск, мм рт. ст.	Вітер, м/с	Вологість, %
6:00		+5	726	ПнЗ 4	93
9:00		+6	728	Пн 5	81
12:00		+6	729	Пн 5	81
15:00		+6	730	Пн 4	93

## Лабораторна робота №3

### Створення простої сторінки

#### Маршрути трамваїв львів

<u>Трамвай № 1</u>	<u>Трамвай № 2</u>
Залізничний вокзал - Центр. Маршрут трамваю № 1: ВОКЗАЛ → вул. Чернівецька → вул. С. Бандери → вул. Коперника → вул. Дорошенка → пл. Ринок → вул. Руська → вул. Підвальна → вул. Винниченка → вул. І. Франка → вул. Вітовського → вул. Котляревського → вул. Київська → вул. С. Бандери → вул. Чернівецька → ВОКЗАЛ.	вул. Пасічна - вул. Коновальця. Маршрут трамваю № 2: вул. КОНОВАЛЬЦЯ → вул. Чупринки → вул. Коперника → вул. Дорошенка → пл. Ринок → вул. Руська → вул. Підвальна → вул. Личаківська → вул. ПАСІЧНА.
<ul style="list-style-type: none"><li>Графік руху трамваю №1: 06:00 - 22:00, інтервал: 10-15хв.</li><li>Вартість проїзду: 5 грн.</li></ul>	<ul style="list-style-type: none"><li>Графік руху трамваю №2: 06:00 - 21:45, інтервал: 5-10хв.</li><li>Вартість проїзду: 5 грн.</li></ul>

## Лабораторна робота №3

### Створення простої сторінки

Львів за один день: маршрут, місця, які варто відвідати



#### Музеї

- Будинок Стрілецького товариства
- Миський арсенал

#### Кафе

- Ресторан-кнайпа "Кривка"
- Гасова льмпа
- Дім легенд

#### Театри

- Театр «Воскресіння»
- Театр ім. Леся Курбаса
- Драматичний театр

## Лабораторна робота №3

### Створення простої сторінки

#### Львівська політехніка



- Вищий технічний навчальний заклад у Львові
- Національний університет «Львівська політехніка» — найстаріший вищий технічний навчальний заклад України та Східної Європи, заснований 1816 року як Реальна школа з дозволу австрійського імператора Франца I.
- Адреса: [вулиця Степана Бандери, 12, Львів, Львівська область, 79000](#)
- Ректор: [Бобало Юрій Ярославович](#)

5. Додати певні елементи: бічну панель, навігацію, таблицю тощо. Внести власні зміни щодо розмірів, кольорів, наповнення тощо.

6. Додати до сторінки стилі основних елементів (h1-h6, p, table, td, ul,ol,li тощо) в різний спосіб

a. **В самому елементі.** Наприклад, `<h1 style="color:red; text-align:center;">Заголовок</h1>`

b. **В елементі <style>**, що знаходиться в елементі <head> або <body>.

Наприклад,

`<style>`

`h1 {color:red; text-align:center;}`

`</style>`

- c. **В окремому файлі style.css**, що потрібно прив'язати до сторінки html (назви і кількість файлів зі стилями є довільними).

```
<head>  
<link href="style.css" rel="stylesheet">  
</head>
```

7. Додати до елементів html-коду різноманітні класи, що стосуються кольору, фону, розмірів, відступів. Поєднати в елементі кілька класів (через пробіл). **Увага!!!** Назви класів пишуться латиницею, не містять пробілів, не починаються з цифри.
8. Побудувати правильну файлову структуру майбутнього проекту, створити папки для зображень (img або images), таблиць стилів (css) та скриптів (js). Перевірити написання назв файлів та папок латиницею.
9. Відкрити FTP-доступ до сервера в різний спосіб: через спеціалізований FTP-клієнт або файловий менеджер від хостингу. Тестову сторінку та супутні файли (стили, зображення) розмістити на хостингу. Передивитися завантажену на сервер сторінку в різних браузерах і відмітити можливу різницю у відображенні. Результат виконання демонструвати в Інтернеті.
10. На іншому комп'ютері (власному чи колеги) встановити клієнт для віддаленої роботи. Встановити тестовий зв'язок і здійснити певні дії на віддаленому комп'ютері (пересилання файлів, кодування у редакторі тощо).

## *Зміст звіту*

1. Назва та мета виконання лабораторної роботи.
2. Зазначити обраний хостинг та зареєстровану доменну адресу.
3. Скріншоти послідовності роботи з коротким описом кожного з етапів.
4. У висновку оцінити зручність обраного хостингу, адміністративної панелі і наявних сервісів. Оцінити зручність використання та можливості різних FTP-клієнтів. Зазначити відмінності у різних браузерах.
5. Зазначити можливості використаних редакторів для макетування та кодування сторінки.

## *Контрольні питання*

1. Перелічити мінімальний набір програмного забезпечення для веб-розробника.
2. Які види хостингу пропонують для розміщення сайтів?
3. Які причини сприяють вибору безкоштовного хостингу? Які існують застереження щодо використання безкоштовного хостингу?
4. Навести коротку характеристику хостингу, який було відкрито для виконання даної лабораторної роботи.
5. Провести порівняльний аналіз популярних програм, які містять утиліту для FTP-доступу.
6. Перелічити підходи, які застосовують для створення сторінок сайту.
7. Який мінімальний набір засобів має бути для створення простого статистичного сайту?
8. Що таке структура сторінки? Назвіть основні блоки.

9. Які особливі семантичні елементи входять у склад мови HTML5, в чому їх особливості?
10. Перелічити основні правила формування фізичної структури сайту. Які обов'язкові папки мають бути в структурі проекту?
11. В чому відмінності інформації, що вказана у тегах <head> та <body>?
12. В чому основне призначення стилів для HTML-розмітки? Яким чином можна під'єднати стилі до HTML-коду?

### **Використані джерела**

1. Топ-5 БЕЗКОШТОВНИХ хостингів. - <https://www.websiteplanet.com/uk/blog/найкращих-безкоштовних-насправді-се/>
2. Гід по українському хостингу. - <https://hostgid.net/>
3. Як вибрати хостинг. - <https://hostgid.net/baza-znaniy/hosting/kak-vybrat-hosting.html>
4. Як вибрати доменну адресу. - <http://mamontov.com.ua/how-choose-domain-name-ukrainian-company.html>
5. Принципи роботи браузерів. - <https://www.html5rocks.com/ru/tutorials/internals/howbrowserswork/>
6. HTML-довідник <https://schoolsw3.com/html/index.php>
7. CSS-довідник <https://schoolsw3.com/css/index.php>
8. JS-довідник <https://schoolsw3.com/js/index.php>
9. HTML Book <https://html5book.ru/html-html5/>
10. Web-технології <https://htmlweb.ru/html/>

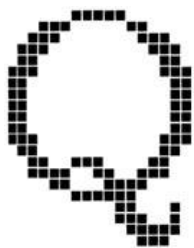
## Лабораторна робота №3. Шрифти та зображення

**Мета роботи.** Набути практичних навичок з втілення нестандартних (довільних) шрифтів у веб-сторінки в різний спосіб. Наочно зрозуміти відмінності між різними графічними веб-форматами. Опанувати прийомами створення зображень та анімації програмним шляхом – SVG, CSS, JavaScript. Ознайомитися з іконковими шрифтами та правилами їх інтеграції до веб-проекту.

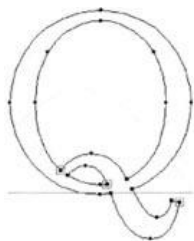
### Шрифти

Шрифт є одним із важливих елементів дизайну документа, який може підсилити чи навпаки зменшити ефективність донесення інформації до користувача. Важко переоцінити значення шрифту в оформленні тексту, будь то веб-сторінка чи журнальна стаття. Вірно підібраний шрифт полегшує сприйняття і додає сторінці свого неповторного стилю.

*За способом створення шрифти бувають растровими і векторними (контурними):*



**В растрових шрифтах** кожен символ описано у вигляді набору точок (пікселів), що розташовані у вузлах сітки растру. Тут, шрифт є звичайним точковим рисунком. Растрові шрифти є непридатними для високоякісного друку і використовуються, в основному, в програмах з текстовим інтерфейсом. Вони широко використовувалися в епоху матричних принтерів і моніторів низької роздільності.



**В векторних (або контурних) шрифтах** символи є криволінійними контурами, які описано математичними формулами. Кожен символ містить координати опорних точок, які сполучені прямими або кривими лініями і утворюють контур символу без прив'язки до абсолютного розміру чи роздільності. Такий опис дозволяє легко змінювати масштаб зображення без втрати якості, що є неможливим у випадку з растровими шрифтами. Векторні шрифти добре відтворюються як на екрані, так і в друці. Векторний комп'ютерний шрифт - це не просто таблиця символів, а невелика і досить складна програма.

*За сімейством шрифти бувають*

### Текст

#### Шрифт із засічками (Serif)

**Засічки або серіфи** (фр. *serif*) - це горизонтальні елементи закінчення основних чи сполучних штрихів, які можуть мати різні форми: прямокутну, зігнуту, заокруглену тощо.

Засічки полегшують читання, ніби сполучаючи літери між собою, і одночасно розділяють окремі букви, щоб вони не зливалися між собою. Шрифти із засічками також називають *антиквенними*, тобто античними.



Шрифти із засічками виглядають традиційно, типово і сприяють легшому сприйняттю великих об'ємів тексту. Яскравим прикладом серіфних шрифтів є *Times*, *Bodoni*, *Garamond*.

## Текст

### Шрифти без засічок (Sans-Serif)

Це так звані **гротески** (нім. *Grotesk*, англ. і фр. *Grotesque*, амер. *Gothic*). В шрифтах без засічок відсутні завершальні елементи на кінцях штрихів. Шрифти без засічок виглядають сучасно і стильно і є широко вживаними для електронних документів, що переглядаються на екрані монітору.

В операційних системах шрифти без засічок користуються більшою популярністю, ніж шрифти із засічками, тому перелік універсальних шрифтів без засічок, що доступні на платформах Windows, Linux і MacOS є більш широким. Це *Arial*, *Impact*, *Helvetica*, *Lucida*, *Tahoma*, *Trebuchet*, *Verdana*.

## Текст

### Шрифти курсивного зображення (Cursive)

Це каліграфічні шрифти, що подібні до рукописних. Вони можуть бути зв'язаними (кожна літера з'єднується з сусідніми) і незв'язаними (де кожна літера візуально є окремою). Можливі також проміжні напівзв'язані форми, де з'єднуються певні частини літер або літери з'єднуються лише з одного боку.

Рукописні шрифти застосовують, зазвичай, не для інформативного тексту, а скоріше для привабливості заголовків.

## Текст

### Моноширинний шрифт (Monospace)

Відрізняється від інших шрифтів однаковою шириною і висотою символів, ця особливість виділяє його серед пропорційних шрифтів, в якому символи відрізняються один від одного. Подібні види гарнітур використовують для таблиць, програм та інших випадків, коли необхідно забезпечити збіг знаків або колонок по вертикалі.

## Текст

### Альтернативні шрифти (Display)

Це шрифти, які створені у власному неповторному стилі. В основному - це новий вигляд шрифтових форм, що не базуються на традиційних символах шрифтів.



### Іконкові шрифти (Icon)



Іконкові шрифти - це шрифти, де кожен символ представлений картинкою. До появи іконкових шрифтів на веб-сторінках використовували растрові зображення іконок, що було незручно. На сьогодні широко вживають векторні іконки, що виконані у вигляді шрифтів.

Принцип роботи таких шрифтів простий, до веб-сторінки, за допомогою CSS стилів, підключається додатковий файл шрифту. В стилях сайту вказуються властивості іконок (розмір, колір, тінь та інші) і в розмітці сторінки пишуться спеціальні теги, на місці яких і з'являються іконки.

Популярні іконкові шрифти:

Font Awesome - <https://fontawesome.com/>

Line Awesome - <https://icons8.com/line-awesome>

*За роллю на сторінці шрифти можна поділити:*

- **Текстові (Text).** Текстові шрифти використовуються для великих фрагментів тексту в книгах і журналах, для набору основного тексту. Особлива увага в дизайні таких шрифтів приділяється зручності при читанні.
- **Акцидентні (Display).** Акцидентні шрифти добре виглядають у великих розмірах і використовуються для заголовків.
- **Декоративні (Decorative).** Декоративні шрифти широко використовуються в рекламі, також можуть використовуватися для заголовків.

*За зображенням шрифти поділяються на три групи.*

- До першої групи відносять шрифти **за ознакою нахилу**: прямий, нахилений, курсивний.
- До другої групи відносять **за ознакою контрастності**: грубий, напівгрубий, тонкий.
- До третьої групи шрифти відносять виходячи зі **співвідношення ширини символу до висоти**: вузький, широкий.

## *Шрифти для Веб*

Шрифт є одним із важливих елементів дизайну документа, який може підсилити чи навпаки зменшити ефективність донесення інформації до користувача. Важко переоцінити значення шрифту в оформленні тексту, будь то веб-сторінка чи журнальна стаття. Вірно підібраний шрифт полегшує сприйняття і додає сторінці свого неповторного стилю.

Під час підготовки до створення веб-сайту, необхідно визначитися зі шрифтами, що будуть використані. Часто різними шрифтами дизайнер набирає не лише основний текст сторінки, але і різноманітні заголовки, логотипи, та інші елементи, що прикрашатимуть сторінку.

Досвідчений розробник знає, що браузер може використовувати для відображення сторінки лише ті шрифти, які встановлені на комп'ютері у користувача. З цієї точки зору шрифти можна умовно розподілити до двох категорій:

- Стандартні шрифти, які містяться в інсталяційних пакетах операційних систем, офісних програмах та Інтернет застосуваннях. Вони без проблем відображаються в переважній більшості користувачів.

- Довільні шрифти, які користувач доставляє за власними потребами та уподобаннями. Вони можуть бути відсутніми у значної групи користувачів.

### **Веб-безпечні шрифти**

В Інтернеті історично склалося таке поняття як «безпечні» веб-шрифти. Безпечним шрифтом можна назвати шрифт, який є стандартним для всіх операційних систем. Нажаль, абсолютно безпечних шрифтів не існує. Окремі шрифти можна назвати безпечними з деякими обмовками.

Основою для визначення «безпечних» шрифтів послужили шрифти найбільш поширеної операційної системи Windows, які використовуються в інших операційних системах.

Arial, Arial Black, Comic Sans MS, Courier New, Georgia, Impact,  
Times New Roman, Trebuchet MS, Verdana

Ці шрифти є в кожного користувача Windows, Mac OS і у переважної більшості користувачів Unix/Linux.

### *Нестандартні (довільні) шрифти*

Для відображення довільних шрифтів останнім часом розроблено багато технологій, і розробнику варто знати про всі, щоб обрати зручний для нього підхід.

### **Під'єднання шрифтів @ font-face в файлі стилів**

Властивість CSS @font-face надає можливість втілити на сайт нестандартні шрифти, які можна вибрати в мережі або створити самостійно. Синтаксис CSS для оголошення користувацького шрифту дуже простий. Потрібно лише зазначити назву шрифту, джерело файлу шрифту і його можна застосувати до будь-якого елемента.

```
@font-face {  
    font-family: "MyFontName";  
    src: url('font.ttf');  
}  
body {font-family: "MyFontName";/* використовуємо підключений шрифт */}
```

### **Колекція шрифтів Google Fonts (<https://fonts.google.com/>)**

Google створив онлайн сервіс, де міститься велика колекція нестандартних веб-шрифтів, в тому числі і кириличних та інструкції по підключенню вибраного шрифту до сайту.

Вибраний онлайн шрифт для сайту буде довантажуватися до браузера користувача з серверів Google, що дозволить не створювати додаткового навантаження на сервер власного хостингу, як у випадку, коли шрифти для сайту зберігаються на його хостингу.

Шрифти з Google Fonts API поширюються за вільною ліцензією і даний спосіб їх під'єднання до сайту буде кросбраузерним за замовчуванням.

### *Використання шрифтів на сайті*

**Вибір типу шрифту** - досить відповідальний етап. Шрифт - це частина веб-проекту. Якщо веб-вузол розробляється для банку або серйозної компанії, нереальним буде використання

шрифту зразку **Comic Sans**, шрифт повинен бути підібраний строгим, що відповідає тематиці сайту. Твердо стверджувати, що існує самий кращий шрифт для верстання сайту не можна. Якщо такий шрифт і існує, то для кожного веб-проекту свій.

## Гарнітура шрифту

**Гарнітура шрифту** (*Type family*) – це сукупність зображень, які об'єднані загальними стильовими ознаками, характером графічної побудови символів і рішенням їх елементів.

В гарнітурі присутній комплект шрифтів, що мають схожий малюнок, але різняться за накресленням (звичайний, курсивний, жирний). Наприклад, шрифти «Arial», «**Arial Bold**», «*Arial Italic*» складають одну гарнітуру «Arial».

Деякі гарнітури містять більшу кількість зображень, ніж інші. Завдяки цьому можна побудувати весь документ на одній гарнітурі, використовуючи там, де потрібно різні варіанти зображень.

## Лінійки шрифтів

Для того, щоб текст сторінки міг відображатися однаково за задумом дизайнера в будь-якій операційній системі, варто задавати лінійки шрифтів: комбінації безпечних, довільних і родового шрифтів.

Для категоризації шрифтів використовуються два типи імен:

- **Назва шрифту (Family-name).** Приклад family-name "Arial", "Times New Roman" або "Tahoma".
- **Родовий шрифт (Generic family).** Його можна простіше описати як групу family-names, що мають характерні загальні риси. Наприклад, sans-serif - набір шрифтів без засічок.

При вказуванні шрифтів для сайту першим вказується самий відповідний шрифт, далі перелічуються альтернативні. Рекомендується в кінці списку вказувати родовий шрифт. Тоді сторінка, як мінімум, буде відображена шрифтом того ж сімейства, якщо відсутні всі специфіковані конкретні шрифти.

Список шрифтів може виглядати так:

```
h1 {font-family: arial, verdana, sans-serif;}
h2 {font-family: "Times New Roman", serif;}
```

Заголовки `<h1>` будуть відображатися шрифтом "Arial". Якщо він не встановлений на клієнтській машині, буде використовуватися "Verdana". Якщо недоступні обидва шрифту, для показу заголовків буде використаний шрифт сімейства sans-serif. Зверніть увагу, що назву шрифту "Times New Roman" укладено в лапки. Якщо в назві гарнітури присутні пробіли, тоді цю назву обов'язково потрібно укласти в лапки.

## Розмір шрифту

Розмір шрифту є важливим інструментом представлення тексту на сайті, він визначає ієрархію розмірів заголовків, основного тексту, а також впливає на сприйняття тексту в загальному. Розмір шрифту знаходиться в дуже тісному контакті з іншими властивостями тексту, такими як ширина колонки, висота відступів тощо.

Одиниці виміру можна розділити на дві категорії: абсолютні і відносні.

- **Абсолютні одиниці** є фіксованими і відносяться до будь-яких фізичних одиниць вимірювання. Після того, як вони були задані, розмір не змінюється.
- **Відносні одиниці** не мають фактичного значення. Їх розмір визначається відносно батьківського елемента. Це означає, що розмір шрифту можна змінити шляхом зміни розмірів пов'язаного елемента.

## Піксель

Піксель px - це базова, абсолютна і остаточна одиниця виміру. Піксели поєднують властивості і абсолютних і відносних одиниць вимірювання (рис.3.1).



Рис.3.1 Розміри символів шрифту в пікселях

Відносність визначається тим, що видимий розмір, вказаний в пікселях, залежить від роздільної здатності монітора. На моніторі з невеликою роздільною здатністю видимий розмір буде більшим, ніж той же розмір в пікселях на моніторі з високою роздільною здатністю. Це пов'язано з тим, що розміри пікселів є незмінними, але на великому моніторі для відображення вмісту екрану використовується більша кількість пікселів. Таким чином, при збільшенні роздільності монітора, розміри, що задані в пікселях, зменшуються.

Абсолютність пікселів обумовлено незалежністю даної одиниці вимірювання від налаштувань браузера. Піксель є мінімальною одиницею вимірювання, тобто, немає величини 0.5 піксела. Шрифт, що заданий у розмірі 12px, завжди відобразатиметься в цьому розмірі і його буде неможливо змінити через налаштування браузера. Таким чином, піксели можна назвати відносними одиницями вимірювання, що є незалежними від налаштувань браузера.

## Абсолютні одиниці

Завдання абсолютних одиниць вимірювання не дозволяє відвідувачеві змінити розмір шрифтів через меню браузера і адаптувати сторінку під свої налаштування.

- **cm** Сантиметр
- **mm** Міліметр
- **in** Дюйм (1 дюйм = 2,54 см)
- **pt** Пункт (1 пункт = 1/72 дюйми  $\square$  0.35 мм)
- **pc** Піка (1 піка = 12 пунктам)

В більшості випадків неможливо заздалегідь передбачити, на якому моніторі, якого розміру і роздільності буде відображено сайт. Текст, що нормально сприймається на екрані з невеликою чи середньою роздільною здатністю, може стати абсолютно нечитабельним при високій роздільній здатності.

Абсолютні одиниці вимірювання варто застосовувати лише тоді, коли відомо про точні фізичні розміри пристрою відображення (наприклад, розміри екрану або сторінки, що буде видрукувана).

Якщо сайт передбачає наявність версії для друку, то там доречні будуть pt, mm, cm, але потрібно точно розуміти, для яких елементів задаються значення. В більшості випадків для Інтернет видань використовуються відносні одиниці.

## Відносні одиниці

Відносні одиниці вимірювань застосовуються в верстці набагато частіше. Відносні одиниці визначають розмір елемента через значення розміру іншого елемента. Таблиці стилів, які використовують відносні одиниці, легше масштабуються з одного середовища виведення в інше.

- % Відсоток від висоти шрифту батьківського елемента.
- em Висота найбільшого символу шрифту «М» поточного елемента (m-height).
- rem Висота найбільшого символу шрифту «М» кореневого елемента <html>.
- vw 1% ширини вікна перегляду (viewport width).
- vh 1% висоти вікна перегляду (viewport height).
- vmin Мінімально допустимий відсоток від висоти.
- vmax Максимально допустимий відсоток від висоти.

Використання відносних одиниць вимірювання надає значні переваги щодо вибору розміру шрифту. Відвідувач може самостійно регулювати розмір шрифту через меню браузера і адаптувати сторінку під власні уподобання. У разі, коли користувач змінює розмір шрифту основного тексту в браузері, пропорційно збільшуються чи зменшуються решта текстових елементів сторінки.

У більшості випадків для визначення розмірів шрифту, а також ширини, висоти і відступів елементів слід використовувати такі одиниці вимірювань, як px, em, rem і %. Для мобільних пристроїв добре підійдуть vh, vw або vmin, vmax.

**Використання відсотків % або em** - це вибір верстальника, тому що по суті обидві одиниці відштовхуються від розмірів шрифту батьківського елемента. Найчастіше вживають одиницю **em**, що відповідає 100% розміру шрифту батьківського елемента. Якщо цей розмір змінити, задавши його більше за одиницю, то шрифт збільшиться. Якщо його поставити менше за одиницю, то шрифт зменшиться. Тобто, число перед em є множником для розміру шрифту.

**Одиниця rem (root em)** відповідає розміру "кореневого" елемента, а саме - теги <html>. Для нього не так часто задаються стилі, тому розмір береться з налаштувань розміру шрифту в браузері, найчастіше, це 16px. Саме він береться за "кореневий" - "root" em - rem. Від цього розміру і розраховуються розміри, що зазначені в rem.

## CSS одиниці для мобільних пристроїв

На даний момент зміна розмірів шрифту в залежності від розміру екрану є важливою частиною веб-дизайну, оскільки під різні роздільності екранів шрифт потрібно "перебудувати" - тобто збільшувати або зменшувати його для зручності користувача.

Для цього застосовують одиниці, які беруть відсоток від viewport, тобто вікна перегляду, яке в телефонах буде від 320px до 480px, в планшетах 768-1024px, а на моніторах - від 1024px. Тому, використовувати тут можна такі одиниці, як

- `vw` 1% ширини вікна.
- `vh` 1% висоти вікна.
- `vmin` найменше з (`vw`, `vh`).
- `vmax` найбільше з (`vw`, `vh`).

Дані одиниці також відмінно зарекомендували себе в тих випадках, коли, наприклад, потрібно встановити висоту елемента, що дорівнює висоті екрану.

## Акцент

Для акцентування уваги на важливих фрагментах тексту можна використати кілька підходів до виділення ключових слів:

- **Bold (грубий).** Найпопулярніший метод акцентування уваги. Як правило, грубішим текстом виділяють заголовки або важливі слова в тексті.
- **Italic (нахилений).** Курсивний текст зле виглядає на екранах з невеликою роздільною здатністю. Не слід також застосовувати курсив в формах. Однак він чудово підходить для цитування фраз або складання списку літератури чи періодичних видань.
- **Underline (підкреслений).** Не варто використовувати підкреслений текст для виділення ключових слів. В Вебі прийнято, що підкреслений текст є посиланням, тому, це може ввести відвідувачів в оману.
- **Колір.** Використання кольору для виділення також не є гарною ідеєю. Акцентування на тексті за допомогою кольорових слів було популярним в минулому столітті, в даний час професійні веб-дизайнери не використовують даний метод виділення.
- **Великі букви.** Не слід використовувати довгі речення, що написані великими літерами. Це буде розцінено як недолік в оформленні сторінки, а також зменшить читабельність решти тексту.

## Оптимальні розміри абзаців

Оптимальні розміри абзаців дозволяють оформити текст на сайті таким чином, щоб відображена в них інформація засвоювалася найкращим чином і створювалося враження легкості читання. Найкраще прописувати не більше 5-7 рядків в абзаці.

Довжина рядка не повинна перевищувати 600 рх (16-17 см). Це оптимальний розмір для комфортного переміщення погляду з одного кінця рядка на інший. Дуже широку контентну частину важко читати - часто втрачається зміст після прочитання довгого попереднього рядка. Якщо необхідно розтягнути текстовий блок на 1000 рх і більше за шириною, можна спробувати розподілити текст на дві або більше колонок.

### **Вертикальні відступи (Margin) між абзацами**

Відступи мають вплив на загальне сприйняття сторінки. Вони допомагають розміщувати текст на різних відстанях від суміжних елементів, а також від меж вікна браузера.

На веб-сторінках існують правила оформлення абзаців та інших елементів тексту. Наприклад, на паперових носіях прийнято робити горизонтальні відступи першого рядка, а на веб-сторінках абзаци мають збільшені відступи між собою. Це пов'язано з тим, що сприйняття тексту з екрану є складнішим ніж з паперу. Тому, великі об'єми тексту бажано розділяти на дрібніші фрагменти і розділяти збільшеними вертикальними відступами.

### **Відстань між рядками (Line Spacing)**

Вертикальна дистанція між рядками має значний вплив на чіткість і стиль відображення тексту. Оптимальною відстанню між рядками вважається відстань, більша за висоту символу. Але, занадто велика міжрядкова відстань також може негативно позначитися на загальному сприйнятті тексту.

### **Вирівнювання (Alignment)**

Для веб-тексту можна використати кілька варіантів вирівнювання:

- **За лівим краєм.** Традиційний підхід до вирівнювання основного тексту.
- **По центру.** Здебільшого застосовується для заголовків чи важливих фрагментів.
- **За правим краєм.** Використовується дуже рідко для специфічних цілей, наприклад, епіграфу.
- **За шириною.** Є найбільш популярним в друкованих виданнях, але рідше використовується в веб типографіці. При вирівнюванні за шириною колонки браузер збільшує або зменшує відстані між словами в кожному рядку, що істотно відбивається на естетичному вигляді загального тексту.

### **Відстань між літерами (Letter Spacing)**

Зазвичай, стандартні шрифти у браузерах відображаються з оптимальною відстанню між літерами та словами. Зазвичай, зміну відстані між літерами застосовують для коротких заголовків або важливої інформації.

### **Хороший екранний шрифт**

До хороших екранних шрифтів можна віднести шрифти з такими характеристиками (рис.3.2):

- Низький контраст, прості блоки з відповідною вагою і товщиною.



- Відповідна висота букв.
- Оптимальна ширина символів і відстань між ними.
- Наявність вільного простору в кожному символі.



Рис.3.2. Характеристики читабельного шрифту

#### **Розробник має враховувати наступні правила:**

- Шрифт за стилем повинен відповідати змістовному навантаженню сайту.
- Зручність сприйняття і читабельності текстів відповідними шрифтами: довжина рядків, міжрядковий інтервал (висота рядків), розмір шрифту, співвідношення тексту і вільного простору на сторінці. Підвищення читабельності тексту забезпечує важливі чинники: увагу, швидкість прочитання та осмислення інформації.
- Вибирати шрифти, що є доступними для користувачів більшості операційних систем, або забезпечити надійне використання довільних шрифтів.

Веб-типографіка домоглася значного прогресу за останні кілька років. Сучасні шрифти відмінно сприймаються з екрану, враховують специфіку верстки, розміщення в Інтернеті, і відображення за допомогою браузерів. Саме типографіка підказала дизайнерам цілий ряд прекрасних рішень в плані побудови сайтів.

### *Іконкові шрифти*

Іконковий шрифт є звичайним шрифтом, у якому замість традиційних символів використовуються колекції масштабованих векторних іконок.

Використання іконок у такому форматі є дуже привабливим рішенням.

Символи іконок дуже легко змінювати: розмір, колір та інші параметри. Здійснюється це за допомогою властивостей CSS, які застосовуються для стилізації звичайного тексту (font-size, color та ін.).

Іконковий шрифт є у векторному форматі, тобто символи добре масштабуються і чудово виглядають на екранах різних пристроїв. Шрифт - це стислий формат, а значить, для його завантаження може знадобитися значно менше часу, ніж для завантаження аналогічних векторних зображень.

Іконкові шрифти підтримуються всіма браузерами. Здійснюється це завдяки створенню на базі одного пакета іконок шрифтів у різних форматах: eot, ttf, svg та woff.

Популярні іконкові шрифти:

- Line Awesome <https://icons8.com/line-awesome>
- Font Awesome <https://fontawesome.com/>

### Підключення іконок Line Awesome у форматі шрифту до сторінки

Для встановлення іконкового шрифту необхідно відвідати офіційну сторінку шрифту, уважно ознайомитися з документацією і завантажити архів з потрібними файлами. Зазвичай в архіві міститься файл зі стилями шрифту та папка fonts зі шрифтом в різних форматах. За промовчанням папка зі шрифтами повинна розташовуватися в тому ж каталозі, що й файл CSS. Для того, щоб каталог fonts розташувати по-іншому, необхідно внести відповідні зміни до файлу "fonts.css" (назва файлу зі стилями різниться відповідно до різних шрифтів).

За бажанням вміст файлу "fonts.css" можна включити до свого файлу стилів. Це дозволить скоротити кількість HTTP-запитів та трохи збільшити завантаження сторінки.

Застосування іконок створеного шрифту на сторінці здійснюється за допомогою елемента i, до якого необхідно додати базовий клас icon і клас іконки. В різних шрифтах базовий клас має різні назви різний, наприклад, las чи fas. Слід бути уважним до версії шрифту, оскільки можуть змінюватися як назви базових класів так і класів іконок

#### FontAwesome

```
<!-- fas - базовий клас -->  
<!-- fa-home - клас іконки Cat -->  
<i class="fas fa-cat"></i>
```



#### LineAwesome

```
<!-- las - базовий клас -->  
<!-- la-home - клас іконки Cat -->  
<i class="las la-cat"></i>
```



## Робота з графікою

Веб-дизайн та графічний дизайн мають багато спільного. Обидва вимагають хорошого розуміння типографіки, графіки і принципів дизайну. Але, це різні напрямки з різними поняттями.

Графічний дизайн бере свій початок в друкованій індустрії, будь то макет журналу або дитяча книга. Друковані матеріали, як правило, менш інтерактивні, ніж їх цифрові аналоги. Але вони так само залежать від художньої збірки зображень, тексту та іншої графіки, щоб розповісти історію або надіслати електронний лист. Там повинна бути логіка і навігація по друкованих матеріалах (зміст книги, журналу) тощо.

У веб-дизайн основна увага приділяється взаємодії користувача з елементами сторінки, а не простому споживанню. Веб-дизайнер може працювати з графічним дизайнером для більш чіткого сприйняття окремих візуальних елементів, таких як ілюстрації і графіка. Але в подальшому веб-дизайнер з'єднує ці елементи разом, щоб створити повний інтерактивний дизайн.

## *Особливості графічних форматів*

Інформація, що зберігається в файлі - це сукупність цифрових даних. Спосіб запису інформації за допомогою послідовності байт називають форматом файлу. Тобто, графічний формат - це спосіб запису графічної інформації. Спосіб представлення зображення впливає на можливості його редагування, друку, на обсяг займаній пам'яті. Існують два основних способи кодування графічної інформації: растровий і векторний.

- **Векторний формат** представляє математичний опис об'єктів відносно точки відліку координат. Наприклад, для того, щоб відобразити пряму потрібно вказати координати двох точок, які об'єднуються за коротшим шляхом, для дуги задається радіус тощо. Таким чином, векторне зображення є набором геометричних примітивів. Більшість векторних форматів можуть містити втілені у файл растрові об'єкти. Цей спосіб кодування ідеально підходить для малюнків, які легко уявити у вигляді комбінації найпростіших фігур, наприклад, для іконок.
- **Растровий формат** характеризується тим, що все зображення по вертикалі і горизонталі розбивається на дрібні точки, так звані піксели (залежно від пристрою ~0.2-0.22 мм). У файлі, що містить растрову графіку, зберігається інформація про колір кожного пікселя. Розмір (size) зображення, що зберігається в файлі, задається у вигляді кількості пікселів по горизонталі (width) та вертикалі (height). Растрові формати різняться між собою здатністю містити додаткову інформацію: різні колірні моделі, вектори, Альфа-канали, прошарки різних типів, інтерліньяж (черезрядкове підвантаження), анімацію, можливості стиснення тощо.

## *Растрові веб-формати*

### **GIF (CompuServe Graphics Interchange Format)**

<https://www.w3.org/Graphics/GIF/spec-gif89a.txt>



Апаратно незалежний формат GIF розроблено в 1987 році (GIF87a) фірмою CompuServe для передачі растрових зображень по мережах. У 1989-му формат модифіковано (GIF89a), до нього додано підтримку прозорості та анімації.

GIF використовує LZW-компресію, що дозволяє добре стискати файли, в яких багато ділянок з однорідним заповненням (логотипи, написи, схеми).

У GIF можна застосовувати прозорі ділянки, вони лишаються прозорими в браузерях та інших програмах і через них просвічується фоновий колір. Прозорість забезпечується за рахунок додаткового Alpha-каналу, що зберігається разом з файлом.

Файл GIF спроможні містити кілька растрових картинок, які браузері підвантажують одну за іншою із вказаною у файлі частотою. Так досягається ілюзія руху (GIF-анімація).

## Особливості

- Кількість кольорів в зображенні може бути від 2 до 256, але це можуть бути будь-які кольори з 24-бітової палітри.
- Файл у форматі GIF може містити лише 100% прозорі ділянки. Якщо використовується відмінний від білого кольору фон, він буде просвічуватися крізь прозорі ділянки в зображенні.
- GIF підтримує покадрову зміну зображень, що робить формат популярним для створення банерів і простої анімації.
- Використовує вільний від втрат метод стиснення.

## Область застосування

Текст, логотипи, ілюстрації з чіткими краями, анімовані малюнки, зображення з прозорими ділянками, банери.

## JPEG (Joint Photographic Experts Group)

<https://jpeg.org/>

JPEG це є назва формату та алгоритму стиснення, який засновано не на пошуку однакових елементів, як у LZW, а на різниці між пікселями.



Кодування даних відбувається в кілька етапів. Спочатку графічні дані конвертуються в колірний простір типу LAB, відкидається половина або три чверті інформації про колір (в залежності від реалізації алгоритму).

Далі аналізуються блоки 8x8 пікселів. Для кожного блоку формується набір чисел. Перші кілька чисел представляють колір блоку в цілому, в той час, як подальші числа відображають тонкі деталі.

На наступному етапі, залежно від обраного рівня якості, відкидається певна частина чисел, що представляють тонкі деталі. На останньому етапі використовується кодування за методом Хафмана для ефективного стиснення кінцевих даних. Відновлення даних відбувається в зворотному порядку. Таким чином, чим вище рівень компресії, тим більше даних відкидається, тим нижчою є якість. Використовуючи JPEG можна отримати файл в 2-500 разів менше, ніж BMP.

JPEG краще стискає растрові картини фотографічної якості, ніж логотипи або схеми - в них більше півтонових переходів, в той час серед однотонних заливок з'являються небажані переходи.

Краще і з меншими втратами стискаються великі зображення для Веб з високою роздільністю (200-300 dpi і більше), ніж з низькою (72-150 dpi), оскільки в кожному квадраті 8x8 пікселів переходи виходять м'якшими, за рахунок того, що таких квадратів є більше.

Формат JPEG не підтримує прозорість і при збереженні зображення з прозорими ділянками, вони зафарбовуються в певний колір.

### Особливості

- В зображенні може бути понад 16 мільйонів кольорів, що цілком достатньо для збереження фотографічних зображень.
- Основною характеристикою формату є якість, яка визначає кінцевий розмір файлу. Слід пам'ятати, що формат застосовує стиснення з втратами. Чим вище стиснення, тим менше якість і навпаки.
- Підтримка технології прогресивного JPEG. Спочатку у вікні перегляду з'являється версія зображення з низькою роздільністю, яке при повному завантаженні поступово набуває початкового вигляду.

### Область застосування

Використовується переважно для фотографій. Не є доцільним для зображень з прозорими ділянками, великими одноколірними ділянками.

### **PNG (Portable Network Graphics)**

# PNG

<http://www.libpng.org/pub/png/>

PNG – Інтернет формат, який долає обмеження GIF. Використовує стиснення без втрат Deflate, подібне до LZW. Стиснуті індексовані файли PNG, зазвичай, є меншими за аналогічні GIF.

Глибина кольору може бути любою, до 48 біт. Використовується двовимірний interlacing (не лише рядків, але і стовпців), що, подібно до GIF, дещо збільшує розмір файлу. На відміну від GIF, де застосовується 100% прозорість, PNG підтримує також напівпрозорі піксели (в діапазоні прозорості від 0 до 99%) за рахунок Альфа-каналу з 256 градаціями сірого.

У файл формату PNG записується інформація про гамму. Гамма є певним числом, що характеризує залежність яскравості світіння екрану монітора від напруги на електродах кінескопа. Це число обчислюється з файлу і дозволяє вводити поправку яскравості при відображенні. Воно потрібне для однакового відображення інформації незалежно від апаратної платформи комп'ютера. PNG підтримується у всіх сучасних браузерях.

### Особливості

- Використовує понад 16 млн. кольорів, тому, застосовується для повнокольорових зображень.
- Підтримує багаторівневу прозорість, це дозволяє створювати плавний перехід від прозорої області зображення до колірної, так званий градієнт.
- Алгоритм стиснення зберігає всі кольори і піксели в зображенні незмінними. Якщо порівнювати з іншими форматами, то в PNG-24 кінцевий об'єм графічного файлу виходить найбільшим.

## Область застосування

Фотографії, малюнки, що містять прозорі ділянки, малюнки з великою кількістю кольорів і чіткими краями зображень.

## Векторні веб-формати



### SVG. Масштабована векторна графіка

SVG (Scalable Vector Graphics) - це тип векторних файлів, що описують зображення в форматі XML. Формат з'явився в 2001, однак популярність серед веб-розробників він отримав нещодавно, після впровадження підтримки у сучасні браузері. Формат є відкритим стандартом, на відміну від більшості інших форматів, SVG не є чиеюсь власністю.

Файл із зображенням в цьому форматі є звичайним текстовим файлом, який можна відкрити в блокноті і відредагувати. У цьому форматі можна описати не тільки статичну, а й динамічну картинку (анімація), змішати створені вектори з растровою картинкою. Завдяки тому, що кожна фігура для браузера є елементом DOM, за допомогою JavaScript можна описувати досить складні сценарії, взаємодіяти з користувачем.

Розмір об'єктів SVG є меншим за розмір растрових зображень, а самі зображення не втрачають якості при масштабуванні. На відміну від растрових форматів можна взаємодіяти із зображеннями у форматі SVG - за допомогою CSS можна змінювати параметри графіки: колір, прозорість або межі, а за допомогою JavaScript - анімувати зображення.

У SVG є маса функцій, які роблять цей формат рекомендованим для Вебу, особливо якщо SVG використовується для простих зображень типу логотипів, карт, іконок, маркерів.

### Переваги формату SVG

- SVG часто важать менше за растрові зображення.
- Формат масштабується, що забезпечує чіткість за будь-якої роздільності екрану.
- SVG-код можна помістити в HTML і налаштувати через CSS.
- SVG-зображення можна анімувати, в тому числі окремі частини, як за допомогою CSS, так і JS.

Втім занадто складні SVG-зображення збільшують розмір файлу. SVG не застосовний до фотографій, тут краще підійдуть формати JPG і webP.

## Анімаційні веб-формати



### APNG. Анімований PNG

aPNG (Animated Portable Network Graphics) запропоновано в 2004 році компанією Mozilla. Новий стандарт базується на форматі PNG, додано можливість анімації та покращено алгоритми стиснення. Втім компанія, що підтримує формат PNG не визнала новий стандарт, і aPNG не пішов в реліз. Основна причина відмови звучала так «PNG - це формат для нерухомих зображень».

До 2008 року не було жодних спроб широкого впровадження aPNG. В 2008 році, компанія Mozilla вносить його в свій браузер Firefox, пізніше таку підтримку додано в браузер Opera. З приходом HTML 5, ситуація почала змінюватися. Google, Twitter, Facebook і інші популярні сервіси, стали один за одним відмовлятися від застарілих технологій, і переходити на нові, паралельно, задаючи стандарти розробки.

## WebP



<https://developers.google.com/speed/webp/>

Сучасний формат зображень з відкритим вихідним кодом, розроблений компанією Google спеціально для Інтернету, є відгалуженням відеоформату WebM.

Подібно до APNG, він також підтримує 24-бітні зображення, 8-розрядну прозорість. У ньому є стиснення з втратами та без втрат, що у деяких випадках дозволяє досягти невеликого розміру файлів із забезпеченням високої якості, що робить його універсальним. Формат поєднує переваги JPG і PNG без збільшення розміру файлу. Сьогодні YouTube використовує перетворення мініатюр для відео в формат WebP.

Поки формат підтримується лише браузером Chrome, і має певні недоліки щодо відображення дрібних деталей. Для підтримки в інших браузерах існують способи обходу обмежень (додаткові плагіни або конфігурація), але вони не сприяють використанню формату повсюдно.

## FLIF Free Lossless формат зображення

<http://flif.info/>



FLIF - це новий та інноваційний формат зображення, що підтримує анімацію з альфа-прозорістю, прогресивне завантаження (що дозволяє відобразити зображення в меншій якості, поки він довантажується), а також інші можливості у порівнянні з різними форматами. Формат поки не підтримується всіма браузерами, але розробляються JavaScript-плагіни, що дозволяють використовувати FLIF у будь-якому браузері.

## *Графічні редактори для створення чи редагування зображень*

### **Растровий редактор Photopea (<https://www.photopea.com/>)**

Photopea - онлайн-редактор зображень, призначений для роботи з растровою і векторною графікою. Спроможний до вирішення як простих питань, наприклад зміна розмірів зображень, так і складних - створення веб-сторінок, ілюстрацій і редагування фото (рис.3.3). Ресурс оптимізований для роботи на будь-якому пристрої: комп'ютер, ноутбук, планшет або телефон. У процесі редагування можна відключитися від інтернету і продовжити в автономному режимі.

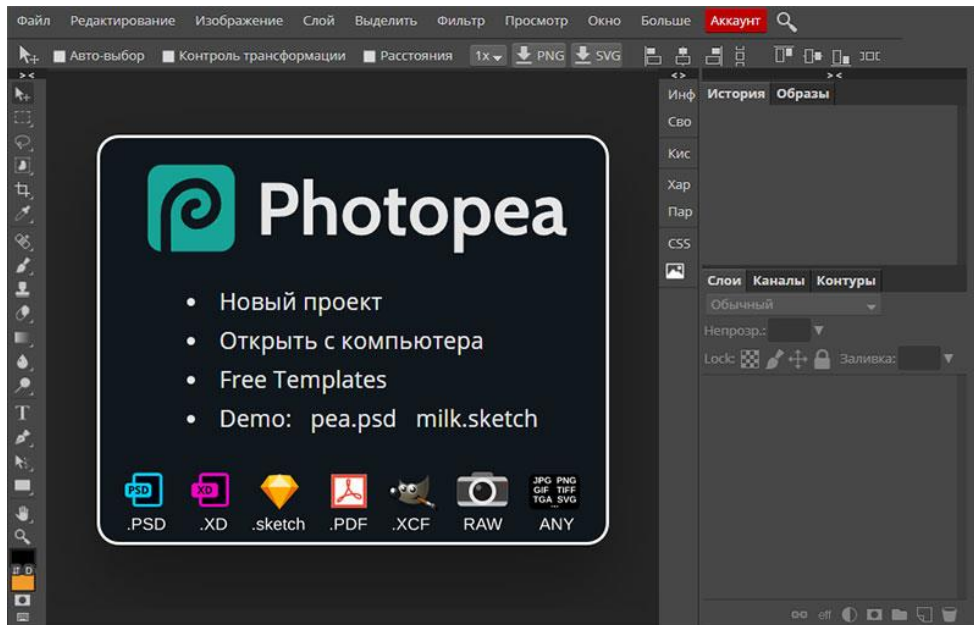


Рис.3.3. Интерфейс редактору Photopea

### Векторный редактор Method Draw (<https://editor.method.ac/>)

Method Draw - Веб додаток для роботи з векторною графікою в режимі онлайн. Сервіс має достатній набір інструментів для створення SVG зображень, також існує можливість відкривати звичайні растрові зображення і створювати на їх основі векторні малюнки. Зовні Method Draw нагадує Adobe Illustrator (рис.3.4).

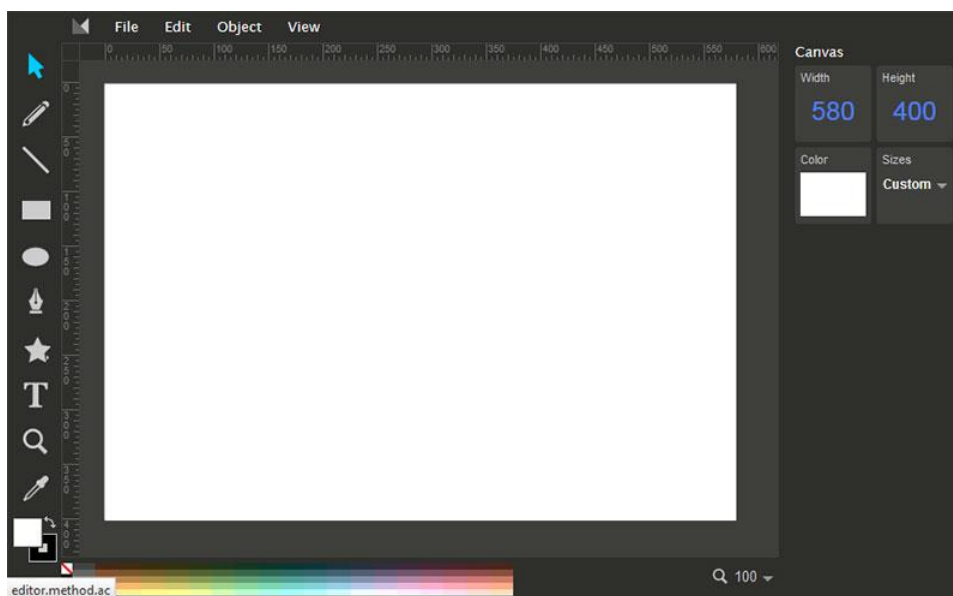


Рис.3.4. Интерфейс редактору Method Draw

### Сервіс для видалення фону з фотографій Remove.bg (<https://www.remove.bg/>)

Remove.bg сервіс, який завдяки штучному інтелекту допоможе скоротити час для видалення фону з фотографій. Сервіс вибирає головний об'єкт з чітко визначеними контурами і видаляє всі решта дрібні і несуттєві деталі (рис.3.5).



# Remove Image Background

100% Automatically and **Free**

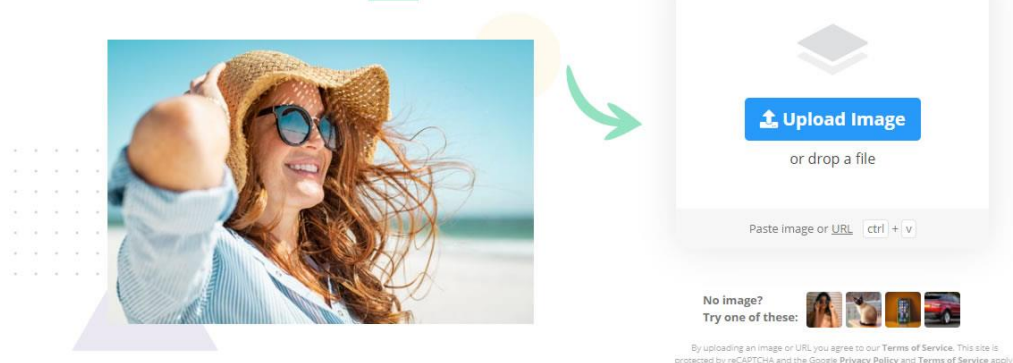


Рис.3.5. Интерфейс сервису Remove.bg

## Сервіс вибору колірної гами Adobe Color Themes (<https://color.adobe.com/>)

Adobe Color - веб-сервіс, що дозволяє створювати пробні і остаточні варіанти колірних тем для використання в проєкті і надавати доступ до них. В Adobe Color CC вбудовано панель «Колір», що дозволяє переглядати і застосовувати колірні теми або помічені як обрані в додатку Adobe Color (рис. 3.6).

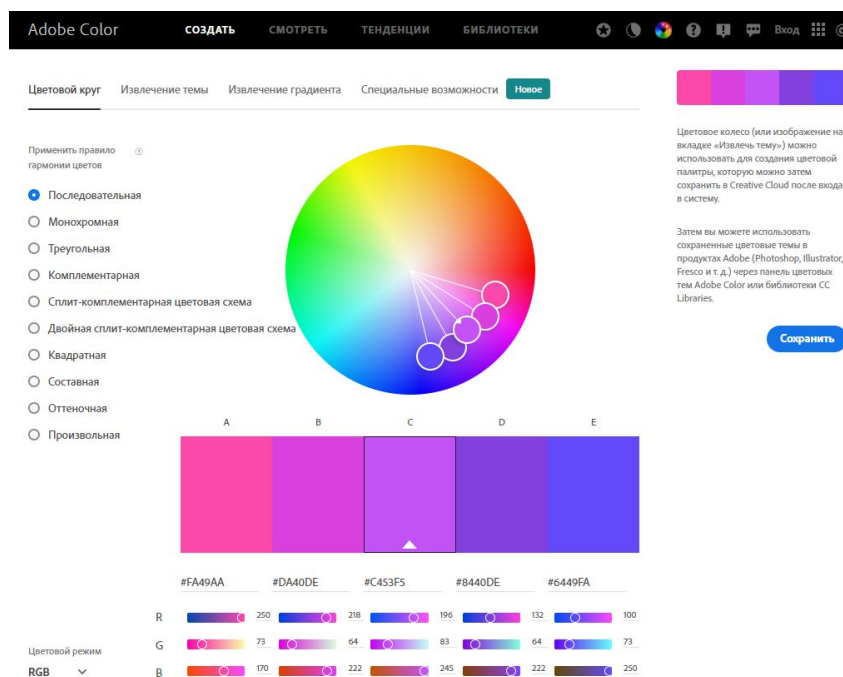


Рис.3.6. Интерфейс сервису Adobe Color

## Сервіс для дизайнерів інтерфейсів Figma

Figma — крос-платформний онлайн-сервіс для дизайнерів інтерфейсів і веб-розробників. У Figma дві ключові особливості: доступ до макету прямо з вікна браузера і можливість спільної роботи над документами (рис.3.7).

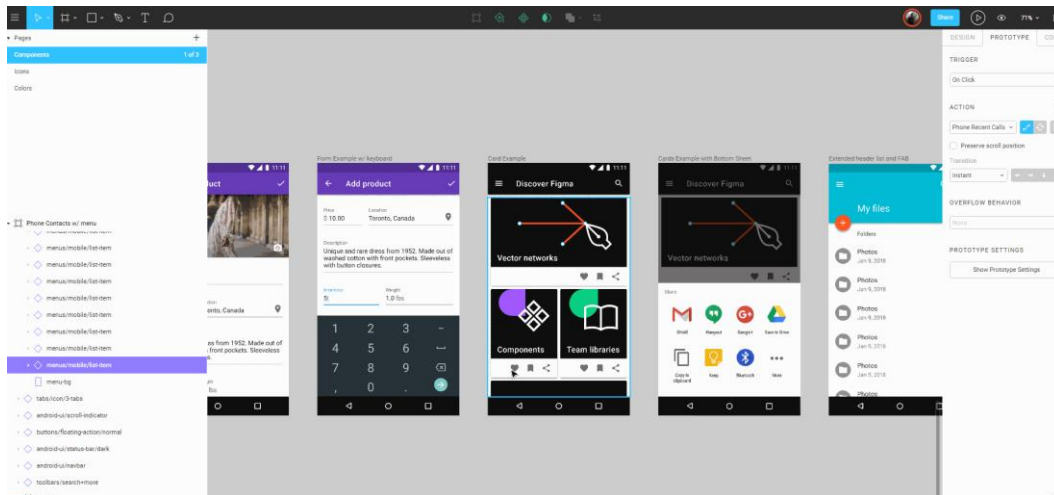


Рис.3.7. Інтерфейс програми Figma

Figma надає можливість кільком дизайнерам одночасно працювати над одним проектом, протоколювати внесені зміни, надавати різні права щодо користування. Це спрощує життя як дизайнеру, так і розробнику. Всі файли автоматично зберігаються в хмарі сервісу Figma.

## *Зображення, що створюють програмним шляхом*

### **SVG зображення**

SVG (Scalable Vector Graphics - масштабована векторна графіка) - графічний формат файлів, який застосовується в веб-дизайні для векторних зображень. SVG є стандартизованим розширенням мови розмітки XML, призначеним для опису векторної графіки переважно на веб-ресурсах. SVG-файли містять XML-розмітку, що описує векторні зображення, а саме всі його деталі у вигляді геометричних примітивів (ліній, кіл, прямокутників тощо) із зазначенням їх розмірів, координат та іншої необхідної інформації.



Можна виділити наступні способи створення векторних зображень у форматі SVG:

- Вручну в текстовому редакторі.
- За допомогою векторних графічних редакторів.
- Через онлайн-сервіси.

Реалізувати анімацію в SVG можна різними способами:

- За допомогою CSS. Цей варіант передбачає маніпулювання геометричними примітивами за допомогою внутрішніх або зовнішніх CSS-властивостей анімації.
- За допомогою JS. Геометричними примітивами в SVG також можна маніпулювати за допомогою JavaScript.
- За допомогою SMIL. Цей спосіб передбачає реалізацію анімації за допомогою спеціальних внутрішніх тегів в SVG-коді.

Найбільш простим способом, що не вимагає знань в JS-програмуванні і SMIL-анімації можна вважати анімацію за допомогою CSS. Для створення складних SVG-анімацій, які потребують певних математичних розрахунків, існують спеціальні JS-плагіни і окремі скрипти.

## CSS зображення

Сучасні можливості CSS дозволяють створювати вражаючі проекти без використання графічних файлів. CSS-зображення це складні фігури, які можна створювати за допомогою псевдоелементів, рамок и тіней на основі html-елементів. Такі проекти є масштабованими, швидкими у завантаженні, простими для редагування.



- Приклади створення css-зображення <https://tproger.ru/translations/pure-css-images-1/>
- Приклад створення 3D-навігації на чистому css <https://codepen.io/katydecorah/pen/HEgw>
- Галерея CSS зображень від Lynn Fisher <https://a.singlediv.com/>

## Програмна анімація

### CSS анімація

CSS анімація дозволяє анімувати переходи від одних CSS стилів до інших. CSS-анімація складається з двох компонентів: стильовий опис анімації і набір ключових кадрів, що визначають початковий, кінцевий і проміжні стани анімованих стилів (рис.3.8).



Рис.3.8. Приклад CSS анімації

CSS-анімація має вагомні переваги перед традиційними способами:

1. Можна створити анімацію, не знаючи і не використовуючи JavaScript.
2. Анімації будуть добре працювати навіть при надмірних навантаженнях системи.
3. Дозволяє браузеру контролювати послідовність анімації, тим самим оптимізувати продуктивність і ефективність браузера.

- Галерея анімованих лоадерів на чистому CSS <https://icons8.com/cssload/en/horizontal-bars>
- Приклади створення css-анімації <https://webdesign.tutsplus.com/ru/articles/pure-css-animation-inspiration-on-codepen--cms-30875>

### JavaScript анімація. Елемент canvas

Елемент Canvas (Холст) — нова особливість в HTML5, що дозволяє малювати в браузері за допомоги технології JavaScript. Його можна використати для створення та анімації форм та фігур, керування елементами на фотографії та відтворення відео (рис.3.9).

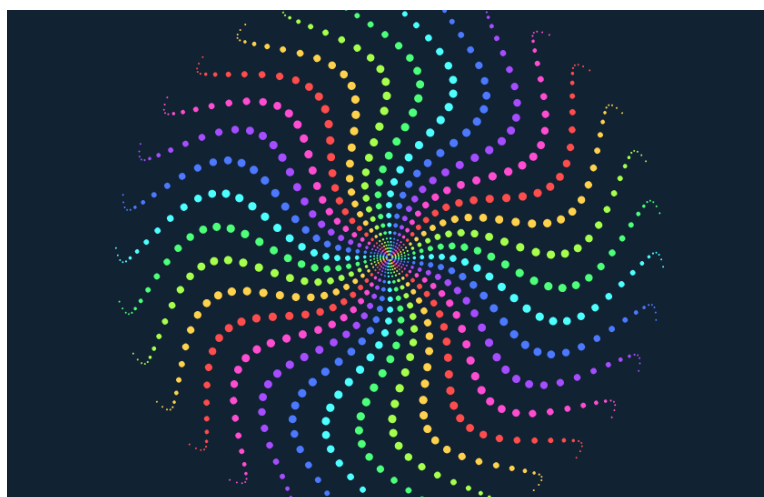


Рис.3.9. Приклад JavaScript анімації

- Колекція js-анімацій <http://beloweb.ru/novichkam/10-krutyih-primerov-raboty-html5-canvas.html>

### SVG анімація

На сьогоднішній день дизайнери для створення анімованих іконок та логотипів частіше використовують SVG-зображення (рис.3.10). Це продиктовано хорошою браузерною підтримкою для анімації SVG і надається більше можливостей для розробників: можна використовувати як власну функціональність SVG-анімації, так і доповнювати JavaScript конструкціями.



Рис.3.10. Приклад SVG анімації

- Колекція анімованих SVG-зображень <https://triu.ru/30-potryasayushhaya-animacziya-svg-dlya-vashego-vdohnoveniya/>

## Порядок роботи

1. Ознайомитися з теоретичними матеріалами щодо веб-типографіки, основними характеристиками шрифтів та особливостями веб-форматів.
2. Завантажити нестандартний шрифт (наприклад, сервіси Fontesk, Fontquirrel ) та під'єднати його до сторінки. Оформити даним шрифтом певні елементи (заголовки, пункти навігації тощо).
3. Під'єднати до сторінки шрифт з сервісу GoogleFonts. Застосувати його до певних елементів (заголовки чи особливі ділянки сторінки).
4. Ознайомитися з ресурсами, що надають у використанні іконкові шрифти. Уважно прочитати документацію щодо уставляння іконок та особливостей надання їм стилів. Створити певну ділянку на сторінці, де розмістити художньо оформлені іконки з відповідною інформацією.
5. Ознайомитися з теоретичними матеріалами щодо комп'ютерної графіки, графічних веб-форматів та їх особливостями. Розуміти відмінності між растровою та векторною графікою
6. Здійснити певні дії в наведених редакторах:
  - Відкрити існуюче зображення, внести зміни (додати текст чи інше зображення, зменшити розмір, повернути елемент тощо).
  - Створити нове полотно певного розміру, зберегти файл. Розробити тематичне зображення: логотип, банер, аватар, мем, листівку, фотоколаж.
  - Створити SVG зображення простої фігури з кольорними елементами для подальшого розгляду його XML-коду.
7. За допомогою HTML-редактора поставити створені графічні елементи у відповідні місця сторінки. Додати стилі до елементів `<img>`, що стосуються розмірів, розташування, тіней, рамок та іншого оздоблення. Створити фавіконку та втілити її на сторінку.
8. Користуючись сервісом Adobe Color Themes підібрати для власного редактора вдалу кольорну гаму. Обирати можна відштовхуючись як від певного кольору так і від фотографії.
9. Створити нескладний дизайн на платформі Figma, відмітити зручності чи складності у використанні. Реалізувати шаблон в HTML-коді та використати згенеровані платформою стилі.
10. На ресурсах безкоштовних колекцій (svgrepo.com, icon-icons.com) обрати певне svg-зображення. Уважно розібрати його html-код, змінити певні значення зображення (розмір, кольори тощо) і вставити на сторінку.
11. Втілити на сторінку зображення (анімації) різних форматів (jpg, png, svg) і способів (css, javascript, svg).
12. Створену сторінку та супутні файли (стилі, зображення) розмістити на хостингу. Перевірити результат виконання через різні браузерери. Результат виконання демонструвати в Інтернеті.

## Зміст звіту

1. Назва та мета виконання лабораторної роботи.
2. Скріншоти послідовності роботи з коротким описом кожного з етапів.

3. Вказати Інтернет-адресу, скріншот та HTML-код створеної тестової сторінки.
4. Скріни етапів виконання роботи та результати втілення шрифтів та іконок на сторінки. HTML код, що було вставлено на сайт.
5. Короткий опис редагування svg-зображення, xml-код та код, що було вставлено на сайт.
6. У висновку оцінити особливості використання нестандартних шрифтів, іконок, растрових, векторних і svg-зображень та різні можливості уставляння.
7. У висновку оцінити особливості роботи зі шрифтами та графікою.

### **Контрольні питання**

1. Перелічити основні типи комп'ютерної графіки.
2. Які графічні формати використовуються у веб? В чому їх відмінності та особливості?
3. Які формати застосовують для зображень з прозорим фоном? Їх особливості?
4. Назвати основні характеристики зображення, яке уставляють на сайт.
5. Що називається фавіконкою? Її призначення та формати?
6. Перелічити основні сімейства комп'ютерних шрифтів. Яке сімейство найкраще використати для сайтів?
7. Зазначити підходи для застосування довільного шрифту на сторінці.
8. Які характеристики шрифту є суттєвими для використання на сайті?
9. Які css-властивості можна застосувати для шрифтів?
10. Що таке іконковий шрифт, яким чином його можна використати на сайті?

### **Використані джерела**

1. Колекція веб-шрифтів <https://webfonts.pro/>
2. Google Fonts <https://fonts.google.com/>
3. Font Awesome <https://fontawesome.com/>
4. Line Awesome <https://icons8.com/line-awesome>
5. Під'єднання нестандартних шрифтів <https://htmlacademy.ru/blog/boost/frontend/fonts-loading>
6. Відеокурс по Figma <https://awdee.ru/figma/>
7. Figma для початківців <https://tilda.education/articles-figma>
8. Видалення фону Remove.bg <https://www.remove.bg/ru>
9. Adobe Color Themes <https://color.adobe.com/ru/>
10. Гід по форматах графічних файлів <https://texterra.ru/blog/gid-po-formatam-graficheskikh-faylov-dlya-nedizaynerov-kogda-i-dlya-chego-ispolzovat-kazhdyy-tip.html>
11. Робота з форматом SVG <https://ru.hexlet.io/blog/posts/kak-rabotat-s-formatom-svg-rukovodstvo-dlya-nachinayuschih-veb-razrabotchikov>
12. Найкращі SVG - PNG іконки для вільного використання <https://icon-icons.com/>
13. Колекція SVG-зображень <https://www.svgrepo.com/>

## Лабораторна робота №4. Втілення додатків з зовнішніх джерел

Мета роботи. Ознайомитися з програмними інтерфейсами веб-сервісів, можливостями щодо втілення виконавчого коду на веб-сторінку. Набути практичних навичок щодо налаштування елементів коду, корекція стилів для візуальної однаковості, тестування втіленого додатку у різних браузерах та пристроях.

### Втілення додатків з зовнішніх джерел

Інтерфейс програмування додатків (API, application programming interface) це набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою, сервісом) для використання у зовнішніх програмних продуктах. Використовується програмістами для написання різноманітних додатків.

API визначає функціональність, яку надає програма (модуль, бібліотека), при цьому API дозволяє абстрагуватися від того, як саме ця функціональність реалізована. Якщо програму (модуль, бібліотеку) розглядати як чорну скриньку, то API - це множина «важелів», які доступні користувачеві даної скриньки, які він може повертати.

Програмні компоненти взаємодіють один з одним за допомогою API. При цьому, зазвичай, компоненти утворюють ієрархію - високорівневі компоненти використовують API низькорівневих компонентів, а ті, в свою чергу, використовують API більш низькорівневих компонентів.

API для сайту - це скрипт, який приймає запити (за методами GET (site.ru / api.php? A = b), POST) і повертає не звичайний HTML для браузерів, а результат запиту у певному форматі (XML, JSON, php serialize (-ed)).

API призначений не для користувачів, а для скрипту зі стороннього сайту / сервісу / програми, який надсилає ці GET/POST запити, отримує результат і використовує дані. Запити надсилає скрипт, щоб виконати певну дію (наприклад, як дію, яку виконують користувачі сайту через браузер). Розробникам-програмістам API потрібен для інтеграції з іншими програмами або автоматизації певних дій.

### Онлайнві картографічні сервіси

Популярні геоінформаційні сервіси пропонують окрім перегляду карт пошук об'єктів, прокладання маршрутів і багато соціальних функції. Кожна з популярних систем має свої сильні і слабкі сторони, зумовлені спеціалізацією або історією розвитку.

#### *Google Maps (<https://www.google.com/maps>)*

Google Maps є популярною та функціональною картографічною системою в Інтернеті. Карти Google охоплюють всю земну кулю, однак їх деталізація від регіону до регіону відрізняється.



Крім стандартної карти місцевості і досить деталізованих супутникових знімків, карта має режим тривимірного відображення земної кулі. Для віртуального глобусу можна відвідати сервіс Google Earth (<https://earth.google.com/web/>).

Картографічний сервіс Google підтримує панорамний перегляд вулиць українських міст. Режим Street View дозволяє уважно розглянути окремі ділянки вулиць і будинків. Вивчення панорами вулиць можна поєднати з навігацією по звичайній карті, а також з переглядом користувацьких фотографій (Panoramio) та відеозаписів (YouTube), територіально пов'язаних з місцем розташування на карті.

В картах добре реалізована функція прокладання маршрутів. Оскільки карта в Google є глобальною, то можна прокласти маршрут не лише в межах одного населеного пункту, а й між різними містами і навіть різними країнами. Картографічний сервіс вміє прокладати маршрут і для подорожі автомобілем, і для прогулянок пішки або поїздки громадським транспортом.

Створювати і редагувати власні карти також можна вручну. Приміром, додавати текстові мітки і об'єкти з коментарями на карту, малювати лінії і деякі інші геометричні фігури. Для створення власної карти можна запрошувати співавторів, а результати роботи публікувати у відкритому або обмеженому доступі.

Для уставка на сайт потрібно на сервісі Google Maps налаштувати потрібну місцевість, масштаб та відмітити потрібну точку на карті. У бічній панелі натиснути посилання Поділитися (рис.4.1).

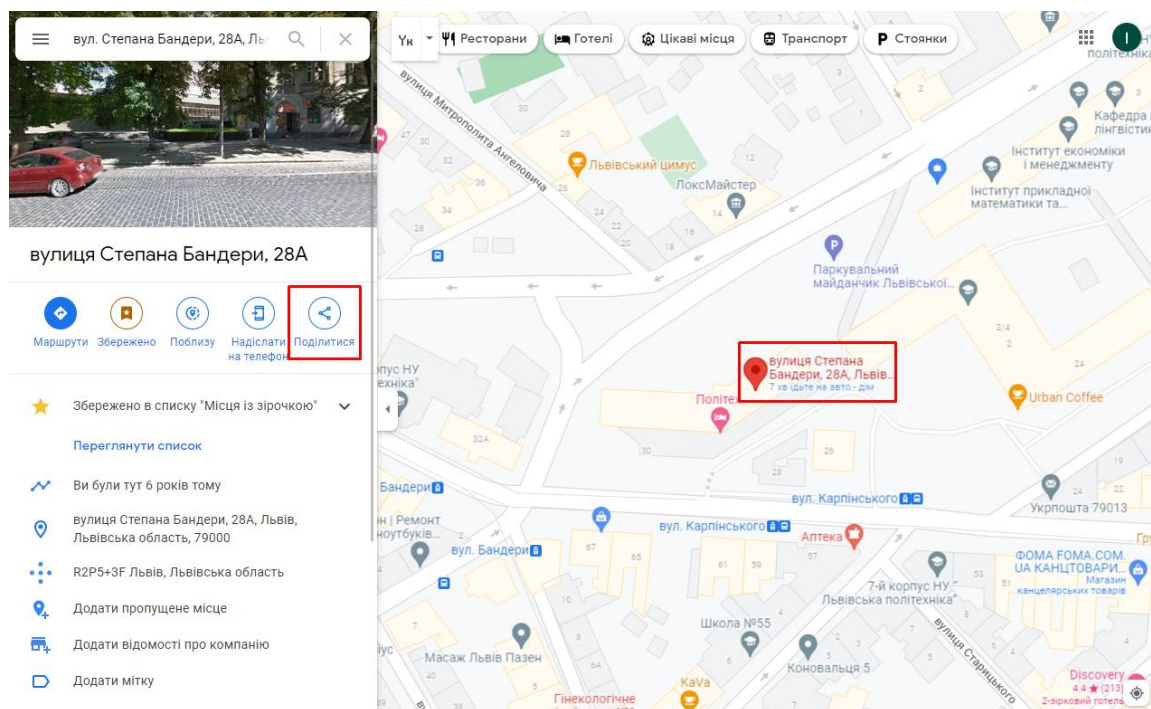


Рис.4.1. Вибір місцевості для уставка карти на сайт

У модальному вікні, що з'явиться потрібно обрати пункт Вставити карту і скопіювати HTML-код, який уставляється на сторінку сайту.



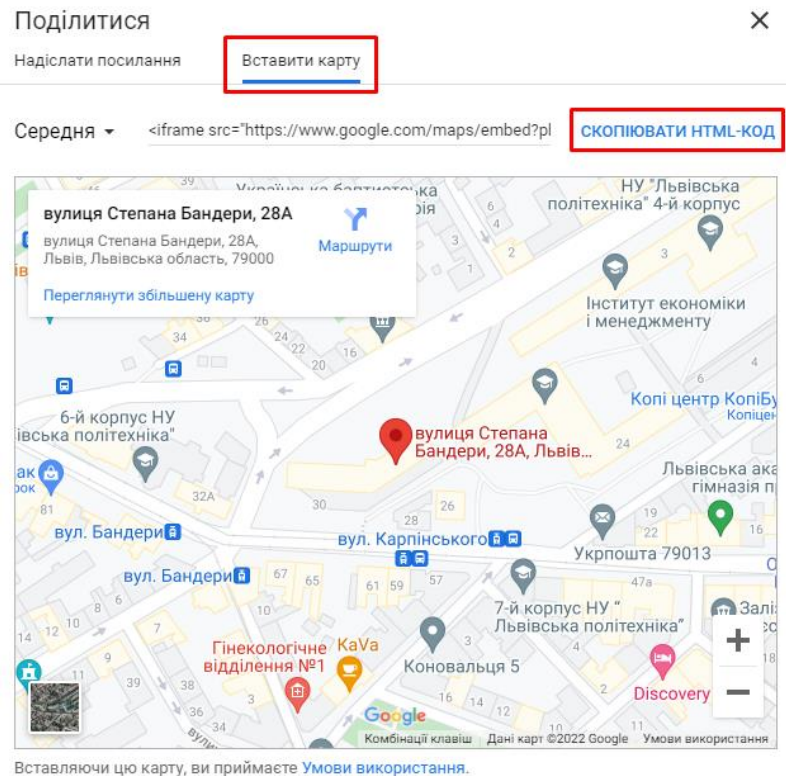


Рис.4.2. Модальне вікно з HTML-кодом

## Перегляд вулиць на Google Maps

На сьогодні сервіс Перегляд вулиць надає змогу не лише переглядати панорами відкритого простору, що доступні через зйомку з автомобілів Google, а й відвідувати різні приміщення (музеї, офіси, готелі, ресторани). Цей сервіс доступний на Картах Google, у галереї Перегляду вулиць (зображення маленького чоловічка у правому нижньому куті карт).

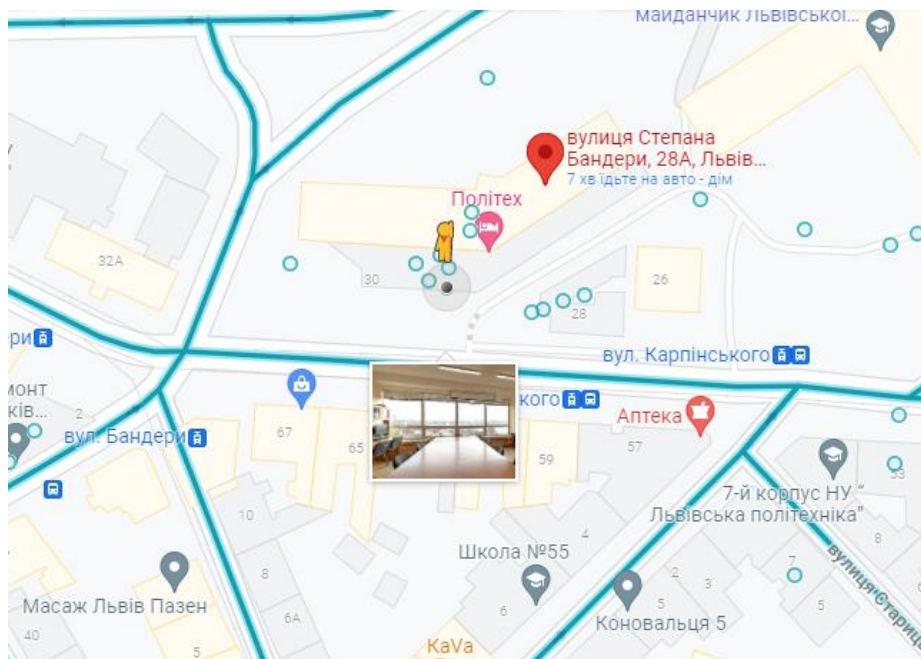


Рис.4.3. Сервіс Перегляд вулиць

Якщо захопити чоловічка мишкою, на карті позначаються місця, що доступні для перегляду: відкритий простір – бірюзові лінії, закритий простір – кружечки (рис.4.3). Потрібно піднести іконку чоловічка до доступного місця і автоматично підвантажиться панорама.

Втілення панорами на сайт відбувається аналогічно до втіленню карт (рис.4.4).

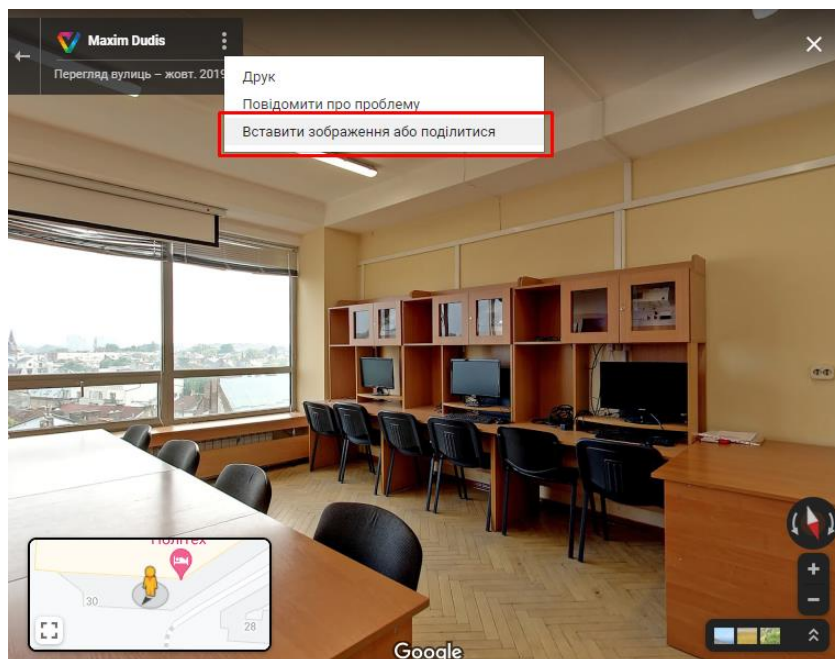


Рис.4.4. Втілення Панорами Google Maps на веб-сторінку

### Карту Bing (<https://www.bing.com/maps/>)

За допомогою сервісу Карти Bing можна додати на сторінку інтерактивну карту. Для уставляння потрібно обрати тип карти, місцевість і масштаб (рис.4.5).

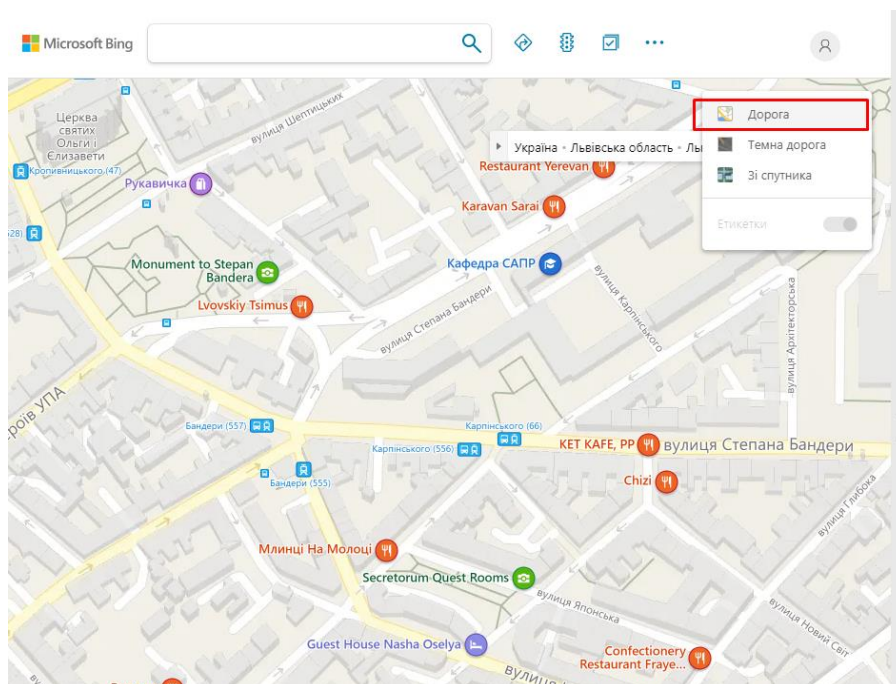


Рис.4.5. Вибір типу карти

На інформативній панелі обрати пункт Вставити карту (рис.4.6).

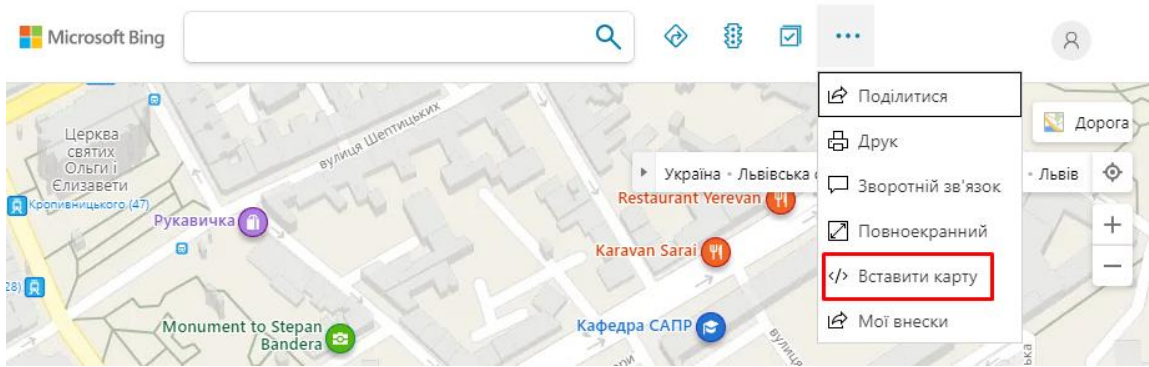


Рис.4.6. Вибір способу використання карти

У новому вікні надається можливість налаштувати вигляд карти, в якому вона буде відображатися на сторінці сайту (рис.4.7).

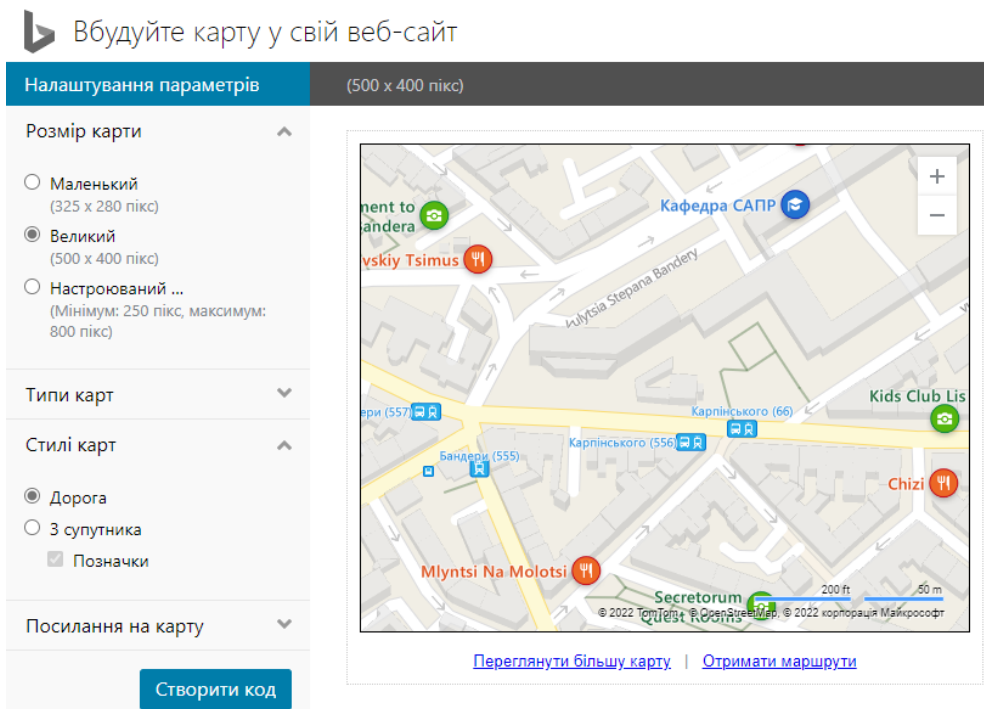


Рис.4.7. Налаштування вигляду карти перед уставлянням на сторінку

### Карти «Мета» (<https://map.meta.ua/>)



Сервіс «Мета» є продуктом українського виробництва. Зареєструвавшись на сервісі, можна додавати власні об'єкти, які будуть відзначені як мітки на карті з коментарями користувача. Є можливість складання маршруту між різними точками на карті. В картах від «Мети» маршрут може бути складений за необмеженою кількістю точок, але маршрут між двома сусідніми точками прокладається не по автомобільним магістралям, а навпростець, незважаючи на будинки та інші перешкоди.

Для уставляння карти на сайт потрібно зареєструватися на сайті meta.ua, перейти до пункту API, отримати ключ і відповідні вказівки для уставляння (HTML, CSS та JavaScript код) (рис.4.8).

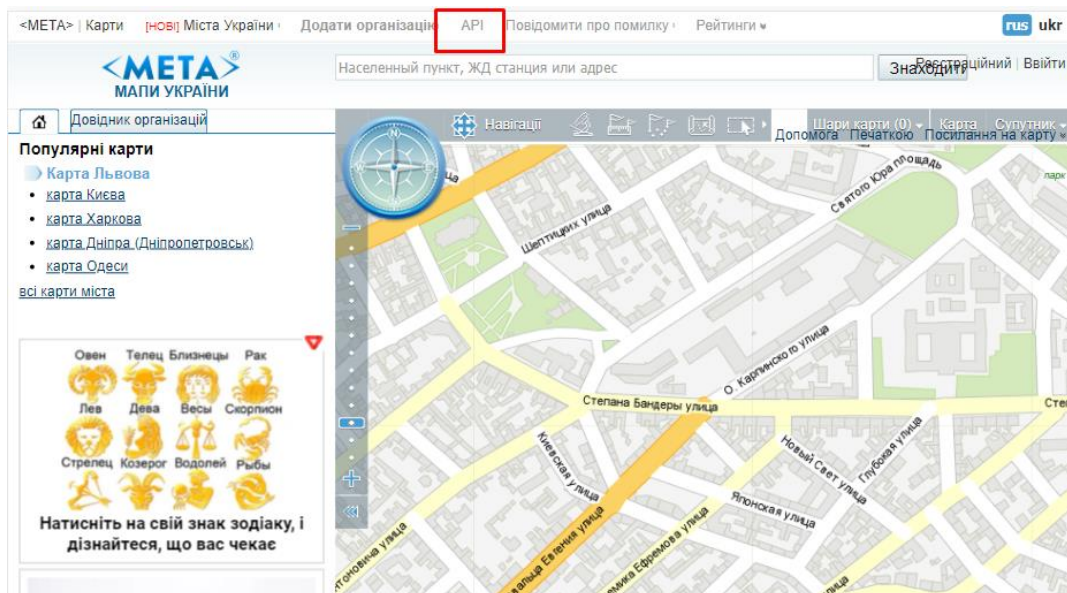


Рис.4.8. Уставляння Карты «Мета» на сайт

## Українська спільнота OpenStreetMap (<https://openstreetmap.org.ua/>)

OpenStreetMap — міжнародний проект, метою якого є створення вільної, відкритої мапи світу. Велика кількість учасників по всьому світу прямо зараз додають дані про місцевість, в якій вони проживають, допомагають наповнити мапу даними про об'єкти інфраструктури поруч із ними, беруть участь у створенні мапи для допомоги постраждалим у зонах стихійних лих, катастроф і таке інше.



Дані, зібрані учасниками проекту, є вільними, відкритими та безкоштовними. Вони отримані з відкритих джерел, передані урядовими та комерційними організаціями, а також зібрані в ході досліджень місцевості рядовими учасниками проекту.

На протигагу власницьким наборам даних, таким як Google Map Maker, ліцензія OpenStreetMap гарантує вільний доступ до всіх наявних даних. Геопросторові дані ліцензуються на умовах Open Database License, яка дозволяє їх використання з будь-якою метою, за умови зазначення походження даних. Проект охоплює всю поверхню Землі.

Дані мап, як всієї Землі, так і окремих її ділянок, можуть бути завантажені як у внутрішньому форматі OSM, так і конвертованими в графічні формати або формати GPS-навігаторів. Можна використовувати Javascript-бібліотеку Google Map для відображення шарів OpenStreetMap.

Можуть бути використані власні фотографії та мапи, вільні від ліцензійних обмежень. Інтерфейс головної сторінки проекту дозволяє додавати у певні місця мапи нотатки, як зареєстрованим, так і анонімним користувачам. Ця інформація може бути використана згодом іншими мапперами в процесі правки даних.

## Мультимедійні сервіси

Популярною на сьогодні областю веб-дизайну є втілення відео на веб-сторінках. Застосування відео в Інтернет постійно зростає, тому стрімко вдосконалюються технології, що

дозволяють швидко завантажувати відеофайли. Вміло застосовані відео можуть зробити веб-сайт більш привабливим для користувачів.

*YouTube* (<https://www.youtube.com/>)



YouTube-сервіс, що надає послуги відеохостингу.

Користувачі можуть додавати, переглядати і коментувати відеозаписи. На сайті представлено як професійно зняті фільми і кліпи, так і любительські відеозаписи, включно з відеоблогами.

#### *Переваги сервісу YouTube*

- Величезна аудиторія ресурсу. Безкоштовність користування.
- Немає обмежень на завантаження відео.
- Налаштування відеоканалу.

#### *Недоліки сервісу*

- Не дуже хороша статистика, яка не дозволяє детально аналізувати аудиторію.
- Недостатня захищеність авторства контенту.
- Багато реклами в відео роликах.

Переглядати відео можуть всі користувачі, включаючи незареєстрованих. Для перегляду деяких зазначених відео необхідно мати зареєстрований акаунт (для чого необхідно мати акаунт в системі Google) і підтвердити повноліття натисканням кнопки.

#### **Завантаження відео на YouTube**

Завантажувати відеозаписи можуть тільки зареєстровані користувачі. Для завантаження можна використовувати ряд поширених форматів, які автоматично і стають доступними для перегляду на YouTube. Збереження записів безпосередньо з YouTube недоступно, але є сайти і програми, що дозволяють це робити.

У зареєстрованих на сайті користувачів з'являється власна сторінка ( «канал»). На ній можна побачити ім'я користувача, дату реєстрації, кількість переглядів каналу, вік, останній вхід і країну проживання (кожен з цих елементів власник каналу може приховати); нові дії користувача (завантаження нових відео та додавання чужих записів в обране); на каналах можуть залишати коментарі глядачі. Також після реєстрації на головній сторінці сайту з'являються додаткові модулі, такі як «Рекомендовані вам відео».

Ролик перед завантаженням варто підготувати: обрізати чи додати фрагменти, накласти музику. Якщо використовувати аудіо трек, захищений авторськими правами, контент може бути заблокованим.

YouTube підтримує такі формати, як MOV, MPEG4, MP4, AVI WMV, MPEGPS, FLV, 3GP, WebM, DNxHR, ProRes, CineForm та HEVC (H.265). Якщо ролик має специфічне розширення, спочатку доведеться перетворити його на відповідний формат.

Для завантаження відео потрібно на лівій інформаційній панелі обрати пункт *Ваші відео*. Відкриється вікно Studio.Youtube з вибором інструментів і кнопкою *Створити* (рис.4.9).

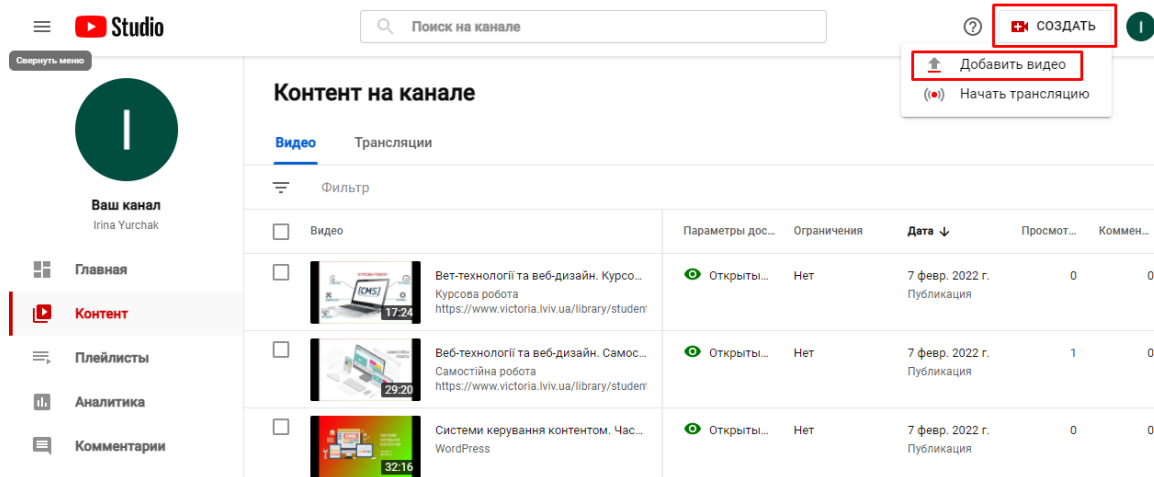


Рис.4.9. Интерфейс Studio.Youtube

У модальному вікні потрібно зазначити відомості, що стосуються відео і опублікувати його (рис.4.10). Після перевірки від сервісу щодо якості та легальності контенту, відео буде опубліковано, і надано постійне посилання на даний ролик.

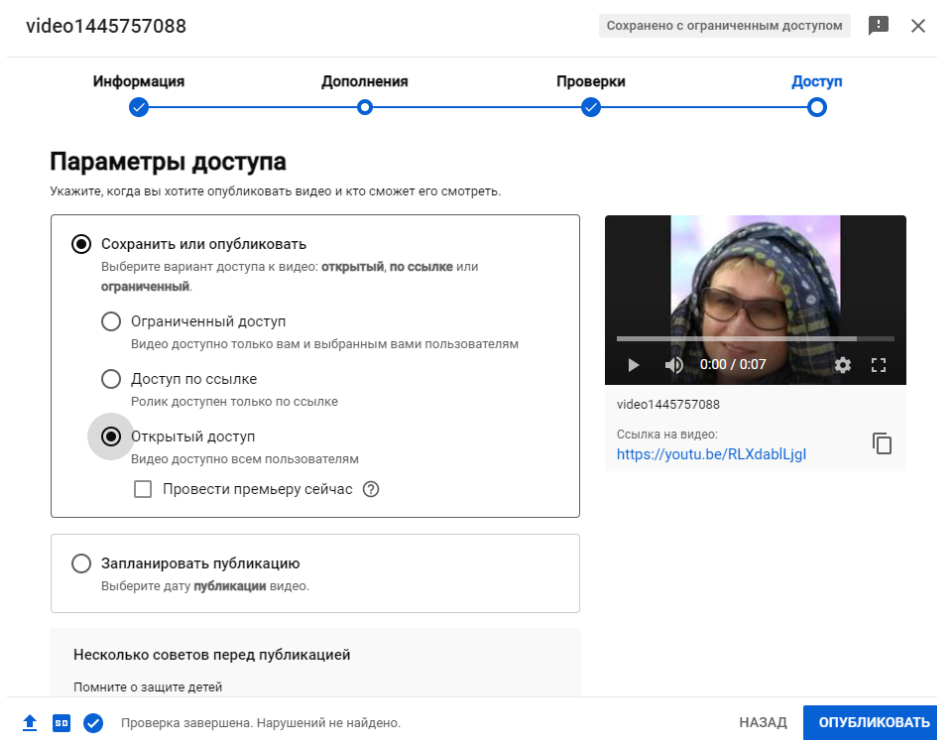


Рис.4.10. Вікно для внесення відомостей про публікацію

Відеозаписи зберігаються вічно. Вони можуть бути видалені власником і за запитом від правовласника в разі порушення авторських прав. На YouTube є система автоматичного розпізнавання нелегального контенту. Якщо в ролик використовується чужа звукова доріжка, може бути тільки відключений звук.

На завантажені відео накладаються обмеження: його розмір повинен бути не вище 2 ГБ (при завантаженні відео за допомогою Java ліміт збільшується до 20 ГБ).

## Втілення відео на сторінку

Відео або список відтворення YouTube можна вставити в блог чи розмістити на веб-сайті. Для цього потрібно натиснути кнопку Поділитися, що знаходиться під роликом (рис.4.11).

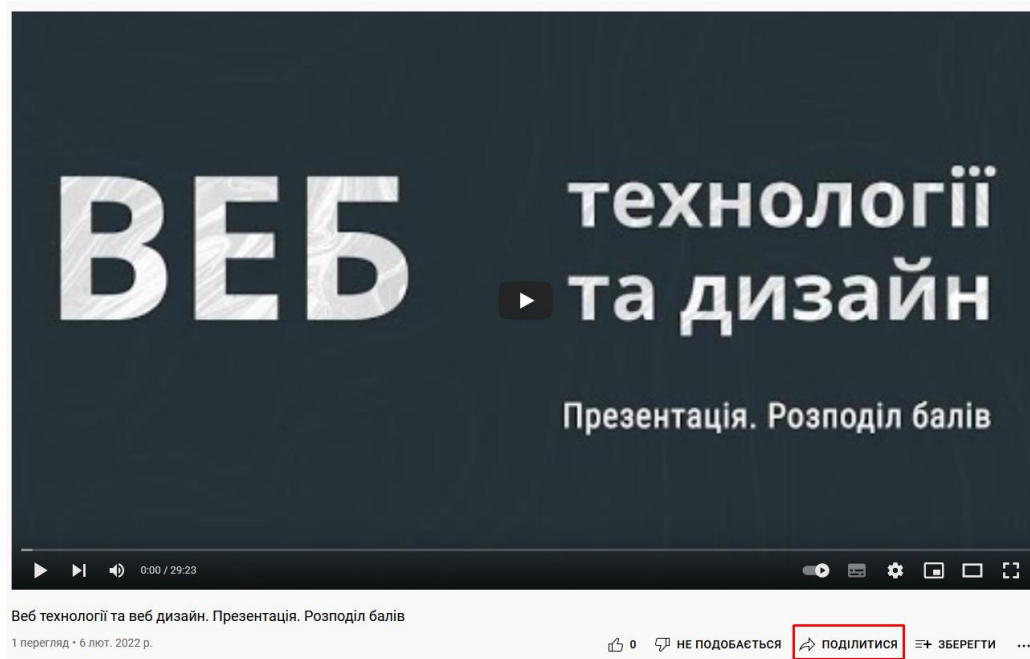


Рис.4.11. Можливості для користувача щодо опублікованого відео

У модальному вікні обрати пункт Вставити, скопіювати наданий HTML-код і вставити його на сторінку сайту (рис.4.12)

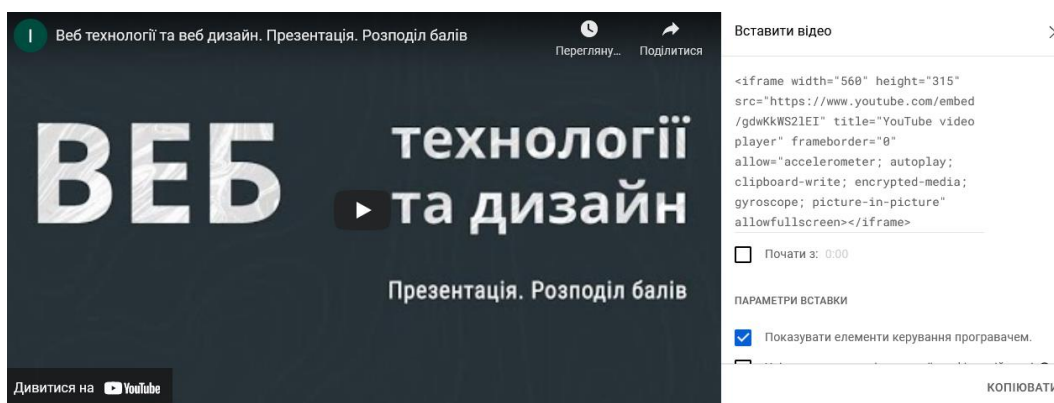


Рис.4.12. HTML-код для уставляння відео на сторінку.

## Vimeo (<https://vimeo.com/>)

Vimeo video - це онлайн сервіс, що надає безкоштовні і платні послуги відіохостінгу. Основною перевагою є можливість завантажувати відео ролики високої якості. Сайт належить корпорації InterActiveCorp (IAC), є основним конкурентом YouTube. Не зареєстровані користувачі на сайті можуть лише переглядати відео, додавати і коментувати відеозаписи дозволено після проходження процедури реєстрації.



### *Переваги*

- Переваги за умови платної підписки (оформлення річної передплати 60 уе). Можна публікувати до 5 Гб відео в тиждень. Завантаження в HD форматі. Завантаження відео здійснюється дуже швидко.
- Немає реклами на відео. Немає обмежень на створення каналів, груп і альбомів.
- Розширена статистика за переглядами відео.
- Конфіденційність відео. Можна вказати на яких сайтах буде програватися відео (вказуються домени), заборонити пошук відео, перегляд відео за паролем.

### *Недоліки*

- У безкоштовній версії можна завантажувати відео до 500 мегабайт в тиждень.
- На відео з безкоштовних акаунтів може бути присутньою реклама.
- Публікація відео на сервісі займає 1-1,5 години.
- Обмеження на створення каналів і груп і альбомів.
- Вбудоване відео на сайті не буде показуватися в HD якості.

Для втілення відео з сервісу Vimeo video потрібно переконатися, що воно доступне для поширення. Про це свідчить іконка Share, при натиску на яку з'явиться модальне вікно з різними способами поширення ролику (рис.4.13).



Рис.4.13. Поширення відео з сервісу Vimeo video

### *Відтворення музики на сайті*

Веб довгий час був безмовним середовищем, але на сьогодні звук є дуже популярним. Продуманий звуковий супровід допоможе відвідувачам сайту отримати максимум вражень від сторінки і утримати їх на сайті.



Перші звукові формати дозволяли створювати файли з настільки великими розмірами, що на завантаження хвилинного музичного фрагмента потрібно було очікувати хвилинами, а то і годинами.

Технології розвиваються і веб-розробники розуміють, наскільки доречним є звук. Використання звуку внесло зміни до дизайну і конструкції сайтів. Наприклад, тепер на деяких сайтах ведеться радіомовлення (webcast — співзвучно до broadcast - віщати, передавати теле- або радіосигнали). Тисячі сайтів в Веб віщають в прямому ефірі. Webcast широко використовують - від музики до прямих трансляцій президентських мов і спортивних ігор.

Якщо сайт має звуковий супровід, слід передбачити кнопку, яка дозволить користувачеві швидко вимкнути музику або звук.

Подібно до графічних файлів, аудіофайли теж мають власні формати. Загальноприйнятими форматами звукових файлів, які добре працюють на платформі Windows є формати WAV, AU, SND, MP3 і MIDI. На платформі Macintosh застосовують формати AIF, AIFF, SND, AU, MP3 і MIDI. Деякі з форматів (AU, SND, MP3 і MIDI) є універсальними і можуть використовуватися як в Windows, так і в Macintosh.

## HTML5 audio

Поява тегу <audio> в HTML5 надало можливості легко відтворювати звук.

Найпростіший спосіб додати на сторінку аудіоплеєр. Це класичний варіант, коли застосовується тег audio, всередині якого:

- Прописується шлях до музичного файлу за допомогою тега source
- Вказується атрибут controls, що відповідає за показ панелі управління відтворенням
- Рядок "Ваш браузер не підтримує аудіоплеєр!" дозволяє користувачам, чії браузери не підтримують стандарти HTML5, побачити, що на цьому місці мав бути аудіоплеєр.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Audio player в html: музикальний потік на сайті</title>
</head>

<body>
  <audio src="music.ogg" controls="controls">
    Ваш браузер не підтримує аудіоплеєр! Оновіть версію браузера!
  </audio>
</body>

</html>
```

На тепер існує 3 вживаних формати для <audio> - ogg, .mp3, wav.

## *Віджети*

Веб-віджет (web widget) - це фрагмент коду, який може бути вбудований користувачем в HTML сторінку і використовуватися без значної модифікації. Як правило, при створенні веб-віджетів використовуються технології JavaScript.

*Веб-віджети можна умовно розділити на:*

- Інтерактивні з якими користувач може взаємодіяти, наприклад, відправляти SMS або шукати маршрут на карті.
- Неінтерактивні, вміст і робота яких не залежить від дій користувача, що переглядає сторінку. Неінтерактивні віджети ще часто називають інформерами. Класичний приклад інформеру - погодний інформер.

### **Кнопки соціальних мереж**

У сучасному інтернет-просторі соціальні мережі є надто популярними. Різні майданчики зосередили мільйони людей, які щодня гортають стрічку, роблять репости, ставлять лайки і ведуть активне онлайн-життя.

Існують особливі кнопки соціальних мереж, установка яких на сайті дозволяє побачити приріст трафіку, а заодно сприяє розвитку PR-кампанії свого ресурсу. Поширення матеріалів забезпечує охоплення гігантської аудиторії, яка складається з потенційних клієнтів, читачів і передплатників.

З установкою інструментів поширення соціальних мереж підвищується рівень довіри до сайту з боку пошукових систем. Поширення за допомогою кнопок соціальних мереж дозволяє збільшити відвідуваність сайту до 20%.

### *Переваги та недоліки сервісів*

Ефективні інструменти для поширення можуть бути додані двома способами:

- Вручну (за допомогою скриптів і плагінів, утворених API самих мереж);
- За допомогою популярних сервісів для додавання кнопок «поділитися в соціальних мережах».

Множинні багатофункціональні сервіси пропонують користувачам широкі можливості в налаштуванні і оформленні, не вимагають наявності спеціальних навичок, мають інтуїтивно зрозумілий інтерфейс для створення красивих елементів. На сервісах, зазвичай, після введення даних генерується код, який слід прописати на сторінці веб-сайті.

Самі соціальні мережі надають API та детальні інструкції для уставляння на сайт низки віджетів.

- Facebook <https://developers.facebook.com/docs/plugins>
- Instagram <https://elfsight.com/widgets/social/>
- Pinterest <https://business.pinterest.com/ru/save-button>
- Youtube <https://developers.google.com/youtube/subscribe/>
- Twitter <https://about.twitter.com/ru/resources/buttons>

## **Інформери**

Інформери - спеціальний блок (відображається у фреймі), який встановлюється на сайті користувача для надання відвідувачам додаткового контенту (оперативної інформації з якоїсь області) від постачальника цієї інформації. Інформація в інформерах оновлюється автоматично.

Існують три типи інформерів - графічні, текстові і інтерактивні.

Найбільш широкого поширення набули валютні, погодні, спортивні та новинні інформери. Деякі інформери містять телевізійну програму передач, анекдоти, різноманітні хіт-паради і подібне. Інтерактивні інформери відрізняються від інших тим, що не "відправляють" відвідувача з сайту, де розміщено інформер, а залишаються для отримання інформації. Крім того, даний вид інформерів може виконувати різні функції: опитувальники, консультації тощо.

### *Порядок роботи*

1. Ознайомитися з наведеними картографічними сервісами та можливостями їх API.
2. Користуючись сервісом Google Maps втілити карту, що відповідає певному місцю. На карті проставити мітки, прокласти маршрути, змінити фон, схему відображення. Додатково розмістити панорамний перегляд цієї або іншої місцевості.
3. Втілити карту від інших картографічних сервісів. Оцінити їх особливості.
4. Ознайомитися з наведеними мультимедійними ресурсами. Завантажити на YouTube та опублікувати власне відео. Додати звукове супроводження, ефекти, додати обкладинку. Втілити відредаговане відео у відповідну сторінку сайту.
5. Втілити на сторінку відео з сервісу Vimeo.
6. За допомогою тегу <audio> додати музичний файл на сайт.
7. Поставити на сторінку кнопки плагінів соціальних мереж та інформери.
8. Протестувати сторінку на різних браузерях та пристроях.
9. Розмістити тестову сторінку на хостингу. Результат виконання демонструвати в Інтернеті.
10. Під час виконання роботи робити скріншоти основних етапів роботи. Оформити звіт.

### *Зміст звіту*

1. Назва та мета виконання лабораторної роботи.
2. Скріни етапів виконання роботи та результати втілення карт на сторінки. HTML код, що вставлено на сайт.
3. Короткий опис уставляння відео зі скрінами. HTML код, що було вставлено на сайт.
4. Короткий опис уставляння аудіо зі скрінами. HTML код, що було вставлено на сайт.
5. У висновку оцінити зручність використання API картографічних та мультимедійних сервісів та різні можливості уставляння.

## Контрольні питання

1. Для чого розробляються API додатків, яку функціональність вони надають?
2. Яким чином можна використати сторонній сервіс з доступним API на власному сайті?
3. В чому переваги картографічних сервісів? Яким чином можна вставити онлайн карту на сайт?
4. В чому проявляються відмінності різних картографічних сервісів?
5. Які обмеження у використанні накладено картографічним сервісом GoogleMaps?
6. Назвіть вагомні відмінності відеохостингів Youtube та Vimeo. Яким чином можна вставити відео з сервісу на користувацький сайт?
7. На які особливості відтворення аудіо файлів на сайті слід вважати: спосіб уставляння, формати файлів, режими відтворення?
8. Які особливі елементи присутні в мові HTML5, для уставляння аудіо та відео. Чи цей підхід є зручним?
9. Зазначити особливості уставляння віджетів соціальних мереж на користувацький сайт. Які переваги надає їх наявність?
10. Які послуги надають інформери, як їх можна підключити?

## Використані джерела

1. Google Планета Земля. - <http://www.google.com/intl/uk/earth/>
2. Як поділитися картою чи маршрутами - <https://support.google.com/maps/answer/144361>
3. Як вставляти відео та списки відтворення - <https://support.google.com/youtube/answer/171780>
4. Vimeo – платформа перегляду і обміну відео особливої якості - [https://webznam.ru/blog/vimeo\\_platforma/2019-10-02-1268](https://webznam.ru/blog/vimeo_platforma/2019-10-02-1268)
5. HTML-елемент <audio> - <https://developer.mozilla.org/ru/docs/Web/HTML/Element/audio>
6. Віджети соціальних мереж на сайт <https://elfsight.com/widgets/social/>
7. Інформери на сайт - <http://coderhs.com/informers/>

## Лабораторна робота №5. Бібліотека jQuery та плагіни

**Мета роботи.** Ознайомитися з можливостями JavaScript бібліотеки jQuery та плагінами, що працюють на її ядрі. Набути практичних навичок щодо втілення готових динамічних елементів на сторінки сайту: слайдер, фотогалерея, випадне меню, календар, лічильник часу тощо. Спромогтися змінити параметри коду або стилів скрипту, для налаштування під стиль сторінки.

### Бібліотека jQuery та JS-плагіни

#### *Мова програмування JavaScript*

JavaScript — це мова програмування, яка приносить інтерактивність до веб-сайту (наприклад: ігри, реакція на натиски кнопок, введення даних через форми, динамічна зміна стилів, анімація). JavaScript спочатку створювався для того, щоб зробити сторінки «живими». Програми на цій мові називаються скриптами. Вони під'єднуються безпосередньо до HTML коду і, як тільки завантажуються сторінка в браузері - тут же виконуються.



Програми на JavaScript є звичайним текстом і не вимагають якоїсь спеціальної підготовки.

JavaScript може виконуватися не лише в браузері, а де завгодно, потрібна лише спеціальна програма - інтерпретатор. Процес виконання скрипта називають «інтерпретацією».

#### **Компіляція і інтерпретація**

Для виконання програм, не має значення якою мовою, існують два способи: «компіляція» і «інтерпретація».

- Компіляція - вихідний код програми, за допомогою спеціального модулю «компілятора», перетворюється в машинний код, який далі поширюється і запускається. При цьому вихідний код програми залишається у розробника.
- Інтерпретація - це коли вихідний код програми отримує інший модуль «інтерпретатор», і виконує його «як є». При цьому поширюється саме сам вихідний код (скрипт). Цей підхід застосовується в браузерах для JavaScript.

Сучасні інтерпретатори перед виконанням перетворюють JavaScript в машинний код або подібний до нього, оптимізують, а потім виконують. І навіть під час виконання намагаються оптимізувати. Тому, JavaScript працює дуже швидко.

У всі основні браузери вбудований інтерпретатор JavaScript, саме тому вони можуть виконувати скрипти на сторінці. Звісно, що JavaScript можна використовувати не лише в браузері. Це повноцінна мова, програми на якій можна запускати і на сервері, і в інших додатках, якщо там встановлено відповідний інтерпретатор.

Сучасний JavaScript - це «безпечна» мова програмування загального призначення. Він не надає низькорівневих засобів роботи з пам'яттю, процесором, і від початку був орієнтований на браузери, в яких це не потрібно.

Інші можливості залежать від оточення, в якому запущено JavaScript. У браузері JavaScript вміє робити все, що відноситься до маніпуляції зі сторінкою, взаємодії з відвідувачем і, в певній мірі, з сервером:

- Створювати нові HTML-теги, видаляти існуючі, змінювати стилі елементів, приховувати та показувати елементи і т.п.
- Реагувати на дії відвідувача, обробляти кліки мишки, переміщення курсору, натискання на клавіатуру і т.п.
- Надсилати запити до серверу і завантажувати дані без перезавантаження сторінки (технологія AJAX).
- Отримувати і встановлювати cookie, запитувати дані, виводити повідомлення і багато іншого.

JavaScript - швидка і потужна мова, але браузер накладає на його виконання деякі обмеження. Це зроблено для безпеки користувачів, щоб зловмисник не міг за допомогою JavaScript отримати особисті дані або нашкодити комп'ютеру користувача.

Цих обмежень немає там, де JavaScript використовується поза браузером, наприклад на сервері. Сучасні браузери надають власні механізми з установки плагінів і розширень, які мають розширені можливості, але вимагають спеціальних дій по установці від користувача.

Більшість можливостей JavaScript в браузері обмежено поточним вікном і сторінкою (рис.5.1).

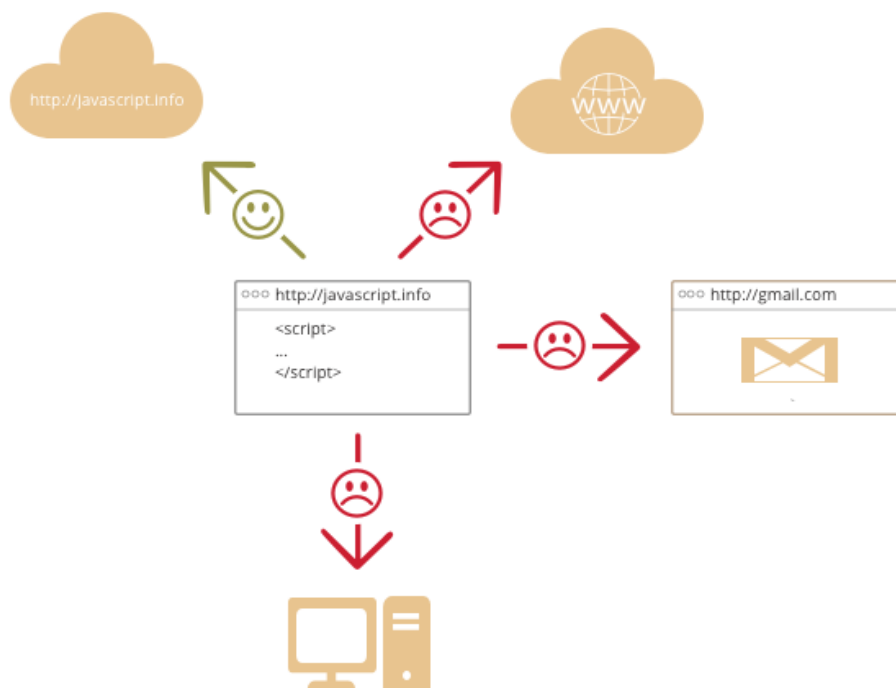


Рис.5.1. Логіка поведінки JavaScript

- JavaScript не може читати/записувати довільні файли на твердий диск, копіювати їх або викликати програми. Він не має прямого доступу до операційної системи.
- Сучасні браузери можуть працювати з файлами, але ця можливість обмежена спеціально виділеної директорією - «пісочницею». Можливості щодо доступу до

пристроїв також опрацьовуються в сучасних стандартах і частково доступні в деяких браузерах.

- JavaScript, що працює в одній вкладці, не може спілкуватися з іншими вкладками і вікнами, за винятком випадку, коли він сам відкрив це вікно або кілька вкладок з одного джерела (однаковий домен, порт, протокол).
- З JavaScript можна легко надсилати запити до серверу, з якого надійшла сторінка. Запит на інший домен теж можливий, але менш зручний, оскільки є обмеження безпеки.

*Особливості JavaScript для веб-сторінок:*

- Повна інтеграція з HTML/CSS.
- Прості речі робляться просто.
- Підтримується всіма поширеними браузерами і включений за замовченням.

Таких речей одночасно немає в жодній браузерній технології. Тому, JavaScript є найпоширенішим засобом створення браузерних інтерфейсів.

## *JavaScript-бібліотека jQuery*

jQuery — популярна JavaScript-бібліотека з відкритим сирцевим кодом, якою користується сотні тисяч сайтів. jQuery є вільним програмним забезпеченням під ліцензією MIT. Синтаксис jQuery розроблено, щоб зробити зручнішим орієнтування у навігації завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків.



jQuery надає можливості для створення плагинів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок.

Основним завданням jQuery є надання легкого та гнучкого інструментарію кросбраузерної адресації DOM об'єктів за допомогою CSS та XPath селекторів. Даний фреймворк також надає інтерфейси для Ajax-застосунків, обробників подій і простої анімації. Принцип роботи jQuery полягає в використанні класу (функції), який при звертанні до нього повертає сам себе. Таким чином, це дозволяє будувати послідовний ланцюг методів.

Бібліотека jQuery є JavaScript файлом, яка містить DOM, події (events), ефекти (effects) і Ajax функції. Бібліотеку можна додати до сторінки посиланням на локальну копію, або на одну з копій доступних на публічному сервері (наприклад Google або Microsoft CDN).

Для використання бібліотеки jQuery потрібно:

1. Завантажити js-файл з офіційного сайту. Існують дві версії jQuery: для використання в готових додатках (production - jquery.min.js) і для розробки (development - jquery.js). Версія для розробки містить коментарі і структурований код. У скороченій версії немає коментарів та стиснутий код, вона займає менше місця і сторінки будуть завантажуватися швидше.

2. Додати js-файл на сторінку в секцію head або в кінці сторінки перед тегом </body>:  
<script type="text/javascript" src="папка\_зі\_скриптами/jquery.js"> </script>
3. За потреби ініціалізувати плагін, що базується на бібліотеці jQuery. Як приклад, певний код ініціалізації наведено нижче:

```
<script>
  $(document).ready(function() {
    $("input").click(function(event) {
      alert("Поздравляем! Вы починили код!");
    });
  });
</script>
```

## Бібліотека jQuery UI

jQuery UI - це бібліотека на основі jQuery, що реалізує багато плагінів, що організують різну поведінку (перетягування або розтягування елементів), віджети (календар, діалогові вікна, система вкладок) і анімаційні ефекти. UI має багато тем оформлення, за допомогою яких оформляються віджети і набір корисних іконок. Тему оформлення можна скоректувати прямо на сайті jQuery UI, безпосередньо перед завантаженням.



### Під'єднання jQuery UI

Для того, щоб скористатися можливостями плагінів jQuery UI їх необхідно спочатку під'єднати до сторінки, на якій вони будуть використовуватися.

- **Локальне під'єднання.** Даний спосіб вимагає завантаження файлу з офіційного сайту.
- **Віддалене під'єднання.** Даний спосіб не вимагає завантаження файлу, а замість цього використовує його віддалено.

Завантажити бібліотеку можна з сайту розробників <https://jqueryui.com/>. У будь-якому випадку потрібно під'єднати стилі бібліотеки, скрипти бібліотек jQuery та jQuery UI.

#### Приклад підключення

```
<link rel="stylesheet" href="//code.jquery.com/ui/1.13.1/themes/base/jquery-
ui.css">
<script src="https://code.jquery.com/jquery-3.6.0.js"></script>
<script src="https://code.jquery.com/ui/1.13.1/jquery-ui.js"></script>
```

Будь-який плагін чи віджет бібліотеки jQuery UI потребує ініціалізації, що прописується або в HTML-кодї сторінки або вноситься в окремий файл. Як приклад, ініціалізація взаємодії Draggable буде виглядати наступним чином:

```
<script>
$( function() {
  $( "#draggable" ).draggable();
} );
</script>
```



Можливості бібліотеки jQuery UI описуються чотирма блоками: взаємодії (Interactions), віджети / елементи управління (Widgets), ефекти (Effects), утиліти (Utilities). Кожен з блоків містить список можливих застосувань.

#### *Взаємодії (Interactions) jQuery UI*

- Draggable - перетягувати HTML-елементи мишкою.
- Droppable - призначити HTML-елемент, на який буде розміщено елемент, що перетягується.
- Resizable - змінює розмір HTML-елемента за допомогою мишки.
- Selectable - використовується мишка для виділення елемента або групи елементів.
- Sortable - дозволяє сортувати елементи на сторінці.

#### *Віджети / елементи управління (Widgets) jQuery UI*

- Accordion - набір елементів, що розгортаються/згортаються.
- Autocomplete - автодоповнення введення по перших символах.
- Button - кнопки на сторінці.
- DatePicker - вибір дати з календаря.
- Dialog - створення діалогових вікон.
- Menu – випадне меню.
- Progressbar - прогрес виконання деякого процесу.
- Selectmenu - розширює функціональність базового HTML-елемента.
- Slider - вибір числа за допомогою повзунка.
- Spinner - для зміни значень в текстовому полі введення.
- Tabs - організація вкладок на сторінці.
- Tooltip - організація спливаючих підказок.

#### *Ефекти (Effects) jQuery UI*

- Add Class - додає CSS-клас елементу.
- Color Animation - розширення анімації jQuery.
- Easing - зміна темпу анімації.
- Effect - анімаційні ефекти.
- Hide - приховування елемента з використанням анімаційних ефектів.
- Remove Class - видалення CSS-класу.

- Show - поява елемента з використанням анімаційних ефектів.
- Switch Class - перемикання класів.
- Toggle - по черзі відображає і ховає елемент з використанням анімаційних ефектів.
- Toggle Class - по черзі додає і видаляє клас з елемента (на відміну від Switch Class працює з одним класом).

### *Утиліти (Utilities) jQuery UI*

- Position - управління позиціонуванням елемента щодо документа, вікна браузера, інших елементів.
- Widget Factory - дозволяє використовувати всі віджети jQuery UI на сторінці.

### *Оформлення jQuery UI*

Для оформлення бібліотеки можна використовувати один з готових наборів (тем) або сформувати власну тему на сторінці оформлювача тем (ThemeRoller). Створена тема представляє папку з CSS, яку буде потрібно розмістити поруч з основним кодом бібліотеки.

## Популярні плагіни, що базуються на jQuery

### *Слайдер Owl Carousel (<https://owlcarousel2.github.io/OwlCarousel2/>)*

Owl Carousel - це популярний слайдер, що має широкі можливості для налаштування, сумісний зі всіма браузерами, підтримує дотики та жести на мобільних пристроях.

### **Встановлення Owl Carousel**

1. Завантажити плагін з офіційного сайту
2. Під'єднати необхідні файли плагіну (стилі і скрипти)
  - `<link rel="stylesheet" href="owlcarousel/owl.carousel.css">`
  - `<link rel="stylesheet" href="owlcarousel/owl.theme.default.css">`
  - `<script src="owlcarousel/owl.carousel.js"> </script>`
3. Переконайтеся, що під'єднано файл бібліотеки jQuery
  - `<script src="jquery.js"> </script>`
4. Ініціалізувати плагін. Цей код потрібно розмістити в кінці сторінки перед тегом `</body>`

```
<script>
    $(document).ready(function() {
        $('.owl-carousel').owlCarousel();
    });
</script>
```

### **УВАГА!!!! Перевірити правильність написання шляхів до файлів**

Далі потрібно написати простий код для слайдера. В основний блок `div` поміщаються блоки з різним вмістом: текст, зображення, блоки інформації.

```

<div class="owl-carousel owl-theme">
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
  <div></div>
</div>

```

Після цього буде створено карусель «базової комплектації» (рис. 5.2)



Рис.5.2. Базовий вигляд слайдера Owl Carousel

Плагін містить багато опцій до налаштувань під власні потреби: зациклити, додати кнопки навігації, пришвидшити прогортання, встановити число слайдів для одночасного показу тощо.

## Галерея *UniteGallery*

Unite Gallery - це багатоцільова галерея javascript, що базується на бібліотеці jquery. Параметри галереї легко налаштувати, змінити тему за допомогою CSS, додавати різні елементи. Галерея дуже потужна, швидка та має більшість сучасних функцій: адаптивність, сенсорне управління, масштабування. Галерея відтворює зображення та відео з Youtube чи Vimeo.

### Встановлення *Unite Gallery*

1. Завантажити плагін з офіційного сайту
2. Під'єднати загальні файли плагіну (стили і скрипти)
  - `<link href="unitegallery/css/unite-gallery.css" rel="stylesheet">`
  - `<link href="unitegallery/themes/default/ug-theme-default.css" rel="stylesheet">`
  - `<script src="unitegallery/js/unitegallery.js"></script>`
3. Переконайтеся, що під'єднано файл бібліотеки jQuery
  - `<script src="jquery.js"> </script>`
4. Під'єднати js-файл, наприклад Tiles Nested
  - `<script src="unitegallery/themes/tiles/ug-theme-tiles.js" type="text/javascript"></script>`
5. Ініціалізувати плагін. Цей код потрібно розмістити в кінці сторінки перед тегом `</body>`

```

<script>
  jQuery(document).ready(function() {
    jQuery("#galleryTiles").unitegallery({

```

```

        gallery_theme: "tiles",
        tiles_type: "nested"
    });
});
</script>

```

**УВАГА!!!!** Перевірити правильність написання шляхів до файлів

Далі потрібно написати код для галереї. В основний блок div поміщаються інші блоки з різним вмістом: текст, зображення, блоки інформації.

```

<div id="galleryTiles" style="display:none;">


</div><!--./galleryTiles-->

```

**УВАГА!!!!** ID галереї має збігатися з id в скрипті ініціалізації (в даному випадку id="galleryTiles").

При правильному виконанні буде відтворена галерея зображень у нерівномірній матриці (рис.5.3).

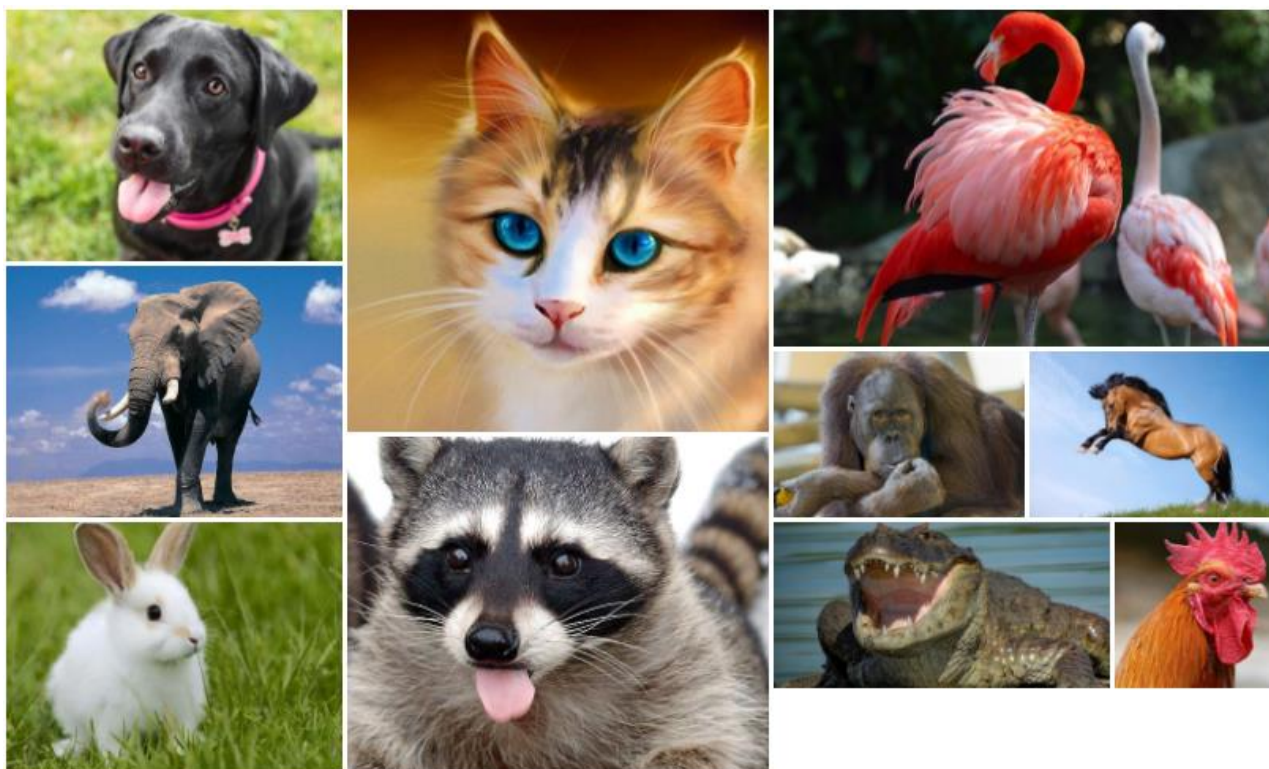


Рис.5.3. Вигляд галереї Unite Gallery Tiles

Для зміни теми, потрібно додати відповідні до теми css та js-файли і за потребою скорегувати код ініціалізації. Нижче наведено код для відтворення відео-слайдера (рис.5.4)

```

<script src="unitegallery/themes/slider/ug-theme-slider.js"
type="text/javascript"></script>
<script src="unitegallery/themes/video/ug-theme-video.js"
type="text/javascript"></script>
<link href="unitegallery/themes/video/skin-right-no-thumb.css" rel="stylesheet"
type='text/css'>
<link href="unitegallery/themes/video/skin-right-thumb.css" rel="stylesheet"
type='text/css'>
<link href="unitegallery/themes/video/skin-right-title-only.css"
rel="stylesheet" type='text/css'>
<script>
    jQuery(document).ready(function(){
        jQuery("#gallerySlider").unitegallery({
            gallery_theme: "video"
        });
    });
</script>

```



Рис.5.4. Відео-слайдер Unite Gallery Video Slider

### Порядок роботи

1. Під'єднати до сторінки скрипт jQuery `<script src="js/jquery.js"></script>`
2. Ознайомитися бібліотекою jQuery UI. Користуючись документацією бібліотеки додати на сторінку 2-3 динамічних елементи
3. Додати до сторінки кілька наведених плагінів, що допускають налаштування опцій.
4. Розглянути коди скриптів і, якщо це можливо, змінити розміри, колір шрифту для стильової сумісності дизайну.
5. Розмістити тестову сторінку на хостингу. Результат виконання демонструвати в Інтернеті.

## *Зміст звіту*

1. Назва та мета виконання лабораторної роботи.
2. Скріни етапів виконання роботи та результати втілення скриптів, динамічних елементів, плагінів.
3. Короткий опис редагування та уставляння елементів з наведенням HTML-коду зі сторінки.
4. У висновку оцінити різні можливості уставляння плагінів.

## *Контрольні питання*

1. Чому js-скрипти називають мовою клієнтського програмування? Чим вони відрізняються від мов серверного програмування?
2. Що собою представляє бібліотека jQuery? Чи можна її використовувати у чистому вигляді?
3. Яким чином можна під'єднати до сайту бібліотеку jQuery?
4. Які можливості надає бібліотека jQuery UI? Яким чином можна уставити динамічні елементи на сайт?
5. Яким чином під'єднати до сайту зовнішній плагін? На які файли потрібно звертати увагу?
6. Для чого застосовують ініціалізацію скрипта? Яким чином можна підключити потрібні опції до скрипту?
7. Оцінити плагіни, які використані для виконання даної лабораторної роботи. Які завдання він вирішує? Чи можливі його модифікації?
8. Перелічити плагіни, які можна застосувати для реалізації слайдера.

## *Використані джерела*

1. Офіційна сторінка jQuery - <http://jquery.com/>
2. Документація jQuery - <http://jquery-docs.ru/>
3. Уроки jQuery для почаківців - <http://ruseller.com/lessons.php?rub=32>
4. jQuery для початківців - <http://anton.shevchuk.name/jquery-book/>
5. Відеоуроки jQuery - <http://www.sdelaysite.com/jquery/jquery-1>
6. jQuery документація - <http://jquerybook.ru/api/>
7. Офіційна сторінка jQuery UI - <http://jqueryui.com/>
8. Зустрічайте – jQuery UI! - <http://coolcoder.co.ua/zustrichajte-jquery-ui/>

## Лабораторна №6. Фреймворк Bootstrap

**Мета роботи.** Ознайомитися з можливостями фреймворку Bootstrap. Набути практичних навичок щодо верстання сторінок сайту за базовими шаблонами. Додати до базових сторінок різні елементи: навігація, кнопки, таблиці, форми. Спромогтися змінити параметри коду або стилів скрипту, для налаштування під стиль сторінки.

### Bootstrap 5

Bootstrap - популярна бібліотека для розробки сучасних, адаптивних і динамічних інтерфейсів веб-сторінок. Це безкоштовна колекція з відкритим вихідним кодом CSS і JavaScript для будь-якого типу сайту чи веб-додатку. Bootstrap надає готове рішення з сотнями сторонніх компонентів, які можна інтегрувати. Це дозволяє швидко створювати прототип для реалізації ідеального веб-сайту, не витрачаючи багато часу.

Bootstrap це набір інструментів для створення сайтів і веб-додатків. Містить HTML/CSS шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення.

#### Основні складові Bootstrap:

- Інструменти для створення макета (обгорткові контейнери, потужні системи сіток, гнучкі медіа-об'єкти, адаптивні утилітні класи).
- Класи для стилізації базового контенту: тексту, зображень, коду, таблиць та зображень.
- Готові компоненти: кнопки, форми, горизонтальні і вертикальні навігаційні панелі, слайдери, списки, акордеони, модальні вікна, спливаючі підказки тощо.
- Утилітні класи для вирішення традиційних завдань, що найбільш часто виникають перед веб-розробниками: вирівнювання тексту, відображення та приховування елементів, вибір кольору, фону, зовнішніх та внутрішніх відступів.

#### Переваги фреймворка Bootstrap

- Економія часу - використовуються вже готові класи і дизайн.
- Адаптивність, висока швидкість і оптимізація, стандартизація інтерфейсів - динамічні макети Bootstrap якісно відображаються на самих різних пристроях без необхідності внесення змін до розмітку.
- Дизайн - єдині шаблони і стильове оформлення елементів макета і всіх сторінок на сайті в цілому. Bootstrap-сторінки кросбраузерні і добре відображається у всіх основних браузерах. Регулярне оновлення і доповнення фреймворку найсучаснішими можливостями HTML і CSS вносить деякі обмеження у використанні з IE7 і IE8.
- Простота і відкритість - використовувати Bootstrap настільки просто, що з ним впораються навіть початківці, а відкритий вихідний код дозволяє брати участь в розробці, модифікувати під власні потреби або просто користуватися хорошим безкоштовним рішенням.

- Код HTML, JavaScript і CSS в Bootstrap продуманий і протестований сотнями розробників зі всього світу.
- Динамічна мова стилів LESS, яка розширює можливості CSS: розробники можуть керувати кольорами, створювати вкладені колонки і змінні.

Розвиток Інтернету і веб-сайтів показало, що половина користувачів Інтернету відвідує сайти з мобільних пристроїв і маленьких екранів. Тому, сучасний веб-розробник повинен враховувати про функціонування сайту не лише на десктопах, але і на смартфонах та планшетах з тачскріном.

Концепція чуйного веб-дизайну, втіленого в Bootstrap, вирішує саме цю проблему: сайт однаково «відгукується» і відображає інформацію кращим чином незалежно від типу екрану і розміру пристрою. Зміст і колірні гамма не змінюються, змінюється лише форма і спосіб групування інформаційних та навігаційних блоків сайту найбільш зручним для користувача способом.

На сьогоднішній день Bootstrap є визнаним фаворитом серед розробників адаптивних сайтів.

### Переваги сайту на Bootstrap

- **Швидкість розробки.** Завдяки великій кількості встановлених компонентів розробнику немає потреби починати розробку проекту з нуля.
- **Адаптивність.** Фреймворк Bootstrap використовує 12 колонкову сітку. При розробці сайту розробник приймає рішення, скільки колонок відвести на певний компонент в залежності від розміру екрана користувача.
- **Популярність.** Мільйони сайтів створено на Bootstrap. Відповідно, розроблено багато шаблонів і різних варіантів оформлення компонентів.

### Шаблони на Bootstrap

- **Безкоштовність.** Bootstrap можна використовувати абсолютно безкоштовно як при самостійній розробці сайту, так і при комерційної діяльності.
- **Гнучке налаштування.** Bootstrap підтримує можливість гнучкого налаштування компонентів. Можна вибрати лише ті компоненти CSS і JavaScript, які знадобляться в поточному проекті.
- **Документованість.** Офіційний сайт Bootstrap надає вичерпну документацію з великою кількістю прикладів. Багато інформації можна знайти на спеціалізованих форумах.
- **Підтримка.** Bootstrap активно розвивається, виходять нові версії, виправляються помилки і з'являється новий функціонал.
- **Еволюція.** Bootstrap відповідає найсучаснішим вимогам дизайну.



## *Bootstrap 5: нововведення, основні стилі*

Bootstrap 5 - це остання версія широко поширеного фреймворку для front-end розробників. У Bootstrap 5 є багато нових цікавих функцій і можливостей, які спрощують процес роботи над проектом.

### *1) Відмова від jQuery на користь нативного JavaScript*

Бібліотека jQuery пережила свої найкращі роки, віддавши пальму першості таким фреймворкам, як Angular, Vue та React. Завдяки прогресу в нативному JavaScript стало можливим відмовитися від підключення бібліотеки jQuery. Це означає, що сайти, створені на Bootstrap 5, швидше завантажуватимуться. Відмова від використання jQuery стала найбільшою зміною за останні роки. Адже всі інтерактивні компоненти Bootstrap працювали на jQuery протягом восьми років, розробникам довелося виконати величезну роботу і все переписати наново.

### *2) Відмова від підтримки Internet Explorer*

Команда розробників відмовилася від підтримки IE 10/11, що виражатиметься у відсутності в розмітці або CSS коді, окремих стилів для підтримки цих браузерів. Це допоможе розробникам не перейматися сумісністю коду зі старими браузерами.

### *3) Додавання CSS змінних*

Bootstrap 4 переважно використовував змінні від препроцесора Sass і кілька кореневих CSS змінних для кольору та шрифтів. Але завдяки припиненню підтримки IE 10/11 (там css-змінні не працювали), змінних стало значно більше.

### *4) Вдосконалення у системі сіток*

Пам'ятаючи труднощі переходу з v3 (float) на v4 (flexbox), команда зберегла основну частину системи сітки, поки залишивши її у флексбоксах.

- Додано новий клас XXL для широкоформатних моніторів, понад 1400 пікселів.
- Замість класу gutter ввели нові класи g\* для вказування відступів між блоками.

### *5) Власні SVG іконки*

У Bootstrap 5 з'явилася нова система іконок SVG, код яких можна легко копіювати і уставляти в будь-який проект, навіть якщо він не створений на Bootstrap.

### *6) Додавання та видалення класів*

Список видалених класів:

- form-row
- form-inline
- list-inline
- card-deck

Список доданих класів:

- `gx*`- простір між блоками по горизонталі.
- `gy*`- простір між блоками по вертикалі.
- `g*`- простір між блоками по горизонталі та вертикалі.
- `row-cols-auto` - автоматичне розподілення колонок сітки.

### Інші нововведення

- Змінився зовнішній вигляд домашньої сторінки та логотип.
- Відбулися покращення у документації – з'явилося більше пояснень.
- Істотно розширено палітру кольору, всі кольори стали доступні у вигляді змінних Sass.
- Оновлено документацію за формами та компонентами.
- З'явилося кілька варіантів для класу `container`, що залежить від ширини екрану.
- Додано більше варіантів стилізації чекбоксів та радіокнопок.
- Можливість створення власних колірних тем.
- Можливість створювати свої класи на основі SASS-змінних за допомогою API утиліт.

## Початок роботи з Bootstrap

### Завантаження бібліотеки

Створення веб-проекту, в основу якого покладено Bootstrap потребує завантаження файлів бібліотеки та їх підключення. Звісно, це краще зробити з офіційного сайту <https://getbootstrap.com/>. (рис.6.1)



Рис.6.1. Інтерфейс офіційної сторінки Bootstrap 5

На сторінці Download присутні три кнопки для завантаження

- **Compiled CSS and JS.** Завантажити скомпільовані CSS та JS файли. Тут містяться лише базові стилі та скрипти, що готові до використання.
- **Source files.** Містить фреймворк у вихідних кодах.
- **Examples.** Містить приклади сторінок з різною структурною розміткою.

## Файлова структура сайту

Після завантаження бібліотеки потрібно створити базову файлову структуру, що на початку буде мати наступний вигляд (рис.6.2):



Рис.6.2. Типова файлова структура для Bootstrap-сайту

На зовнішній вигляд будь-якої веб-сторінки впливають дефолтні стилі браузера, стилі різних фреймворків таких, як Bootstrap, і CSS-стилі, що написані розробником. Бібліотека Bootstrap містить CSS-файл, в якому описано велику кількість корисних класів, гнучких модулів і загальних CSS-властивостей. Звісно, що розробник буде писати власні стилі, які варто розміщати в окремій таблиці стилів (наприклад, style.css), щоб таблиця стилів Bootstrap залишалася оригінальною.

Отже, в папці [css] мають міститися як мінімум 2 файли:

- bootstrap.css
- style.css (\*.css)

В папці [js] розміщують плагіни для надання динамічності елементам сайту. Попередні версії Bootstrap для реалізації деяких власних плагінів використовували функції бібліотеки jQuery, тому обов'язковим було наявність js-файлів jQuery та дотичних бібліотек. У версії Bootstrap5 відмовилися від бібліотеки jQuery на користь нативного JavaScript (bootstrap.js). Для реалізації певних компонентів (tooltip, popover) потрібно підключити зовнішній скрипт Popper або скомпільований скрипт bootstrap.bundle.js, що містить відповідні функції і Bootstrap і Popper.

Отже, в папці [js] мають міститися як мінімум такі файли:

- popper.js
- bootstrap.js

або

- bootstrap.bundle.js (об'єднує файли popper.js та bootstrap.js)

Якщо для реалізації зовнішніх плагінів потрібно використати jQuery, то після її під'єднання, Bootstrap автоматично додасть всі компоненти jQuery у систему плагінів.

Для розгортання веб-проекту, потрібно додати папки [images] та [fonts]. Там будуть міститися відповідно зображення, що відносяться до сайту проекту та використані шрифти, зокрема іконкові. У версію Bootstrap5 додано власну бібліотеку іконок (зображення SVG), що містить

більш як 1300 елементів, які можна масштабувати, модифікувати та стилізувати за допомогою CSS.

### *Під'єднання файлів на сторінці*

Початковий шаблон HTML-сторінки, з підключеними стилями та скриптами виглядатиме так

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
  <link rel="stylesheet" href="css/bootstrap.css">
  <link rel="stylesheet" href="css/style.css">
  <title>Hello, world!</title>
</head>
<body>
  <h1>Hello, world!</h1>
  <!-- Під'єднання JavaScript; Вибрати один підхід з наведених -->
  <!-- Підхід 1: Bootstrap Bundle з Popper -->
  <script src="js/bootstrap.bundle.js"></script>
  <!-- Підхід 2: Окремі Popper та Bootstrap JS -->
  <!--
  <script src="js/popper.js"></script>
  <script src="js/bootstrap.js"></script>
  -->
  <!-- Підхід 3: Використання jQuery -->
  <!--
  <script src="js/jquery.js"></script>
  <script src="js/popper.js"></script>
  <script src="js/bootstrap.js"></script>
  -->
</body>
</html>
```

### *Контрольні точки*

Контрольні точки – це тригери в Bootstrap, які визначають, як макет реагує на зміни залежно від пристрою або розмірів області перегляду.

- Контрольні точки – це будівельні блоки адаптивного дизайну. Їх використовують, щоб контролювати адаптацію макету до певного вікна перегляду або розміру пристрою.

- Для створення CSS по контрольних точках використовують медіа-запити, що надають можливість використати визначені стилі.
- CSS Bootstrap націлений на застосування мінімуму стилів, щоб макет працював з найменшою контрольною точкою, а потім накладання стилів для налаштування цього дизайну для більших пристроїв. Це оптимізує CSS та скорочує час рендерингу.

## Доступні контрольні точки

Bootstrap містить шість контрольних точок за замовчуванням (рівні сітки) для адаптивної верстки сторінки.

Контрольна точка	Префікс класу	Розміри
Екстра-малий	<i>немає</i>	0–576 пікселів
Малий	<code>sm</code>	≥576 пікселів
Середній	<code>md</code>	≥768 пікселів
Великий	<code>lg</code>	≥992 пікселів
Дуже великий	<code>xl</code>	≥1200 пікселів
Екстра великий	<code>xxl</code>	≥1400 пікселів

Розмір кожної контрольної точки є кратним до 12 і представляє підмножину загальних розмірів пристроїв та розмірів області перегляду. Вони спеціально не націлені на всі варіанти використання, але пропонувані діапазони забезпечують підлаштування до будь-якого пристрою.

## Контейнери

Контейнер - це фундаментальний будівельний блок Bootstrap, який містить, доповнює та вирівнює контент до розмірів вікна перегляду. Контейнер потрібний при використанні сіткової системи за замовченням.

Bootstrap надає до використання три класи контейнерів:

- `.container`, який встановлює максимальну ширину в кожній контрольній точці.
- `.container-fluid`, який надає 100% ширину у всіх контрольних точках.
- `.container- { breakpoint}`, надає 100% ширину до вказаної контрольної точки.

У таблиці нижче показано максимальну ширину контейнера в кожній контрольній точці.

	Екстра малий <576 px	Малий ≥576 px	Середній ≥768 px	Великий ≥992 px	X-великий ≥1200 px	XX-великий ≥1400 px
<code>.container</code>	100%	540 px	720 px	960 px	1140 px	1320 px

	Екстра малий <576 px	Малий ≥576 px	Середній ≥768 px	Великий ≥992 px	X-великий ≥1200 px	XX-великий ≥1400 px
<code>.container-sm</code>	100%	540 px	720 px	960 px	1140 px	1320 px
<code>.container-md</code>	100%	100%	720 px	960 px	1140 px	1320 px
<code>.container-lg</code>	100%	100%	100%	960 px	1140 px	1320 px
<code>.container-xl</code>	100%	100%	100%	100%	1140 px	1320 px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320 px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

## Бутстрап сітка

Система сіток Bootstrap використовує серію контейнерів, рядків та стовпців для створення макетів різних розмірів. Система складається з дванадцяти стовпців, шести адаптивних рівнів за замовчуванням, та допоміжних класів. Сітка розроблена з використанням Flexbox, повністю адаптивна, орієнтована на мобільні пристрої.

### Приклад

Один із трьох стовпців	Один із трьох стовпців	Один із трьох стовпців
------------------------	------------------------	------------------------

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      Один из трех столбцов
    </div>
    <div class="col-sm">
      Один из трех столбцов
    </div>
    <div class="col-sm">
      Один из трех столбцов
    </div>
  </div>
</div>
```

У наведеному вище прикладі створюються три стовпці однакової ширини на всіх пристроях та у вікнах перегляду з використанням наших визначених класів сітки. Ці стовпці центруються на сторінці разом із батьківським елементом `.container`.

## Особливості сітки

- Сітка підтримує шість контрольних точок. Відповідні класи орієнтуються на цю контрольну точку і все, що вище (наприклад, `.col-sm-4` відноситься до `sm`, `md`, `lg`, `xl`, і `xxl`). Можете керувати розміром та поведінкою контейнера та стовпця за допомогою кожної контрольної точки.
- Контейнери розташовуються по центру та вирівнюють вміст по горизонталі. Використовують `.container` для гнучкої ширини в пікселях `.container-fluid` для 100% ширини всіх вікон перегляду та на всіх пристроях або для адаптивного контейнера (наприклад, `.container-md`) з комбінацією гнучкої ширини та пікселя.
- Рядки – це оболонки для стовпців. Кожна колонка має горизонтальні внутрішні відступи (`padding`) для контролю відстані між ними.
- У кожному рядку є 12 стовпців, що дозволяє створювати різні комбінації елементів, що охоплюють будь-яку кількість стовпців. Класи стовпців вказують кількість стовпців шаблону (наприклад, `col-4` чотири).
- Внутрішні відступи (`padding`) також адаптивні та налаштовуються різними класами. `.gx-*` класи для горизонтальних відступів, `.gy-*` для вертикальних, `.g-*` для всіх відступів. Наприклад, клас `.g-0` забирає всі внутрішні відступи.

## Параметри сітки

Система сіток Bootstrap може адаптуватися до всіх шести контрольних точок за замовченням і контрольних точок, що вказує розробник. Кожна з контрольних точок має власний контейнер, унікальний префікс класу та модифікатори.

	xs <576 px	sm ≥576 px	md ≥768 px	lg ≥992 px	xl ≥1200 px	xxl ≥1400px
Контейнер <code>max-width</code>	Авто	540 px	720 px	960 px	1140 px	1320 px
Префікс класу	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>	<code>.col-xxl-</code>

## Доступні кольори

Кольори - це візуальна мова, у сфері дизайну кольори можуть бути застосовані до великої кількості елементів: тексту, заднього фону, меж та багато чого іншого.

На даний момент існує понад 100 назв кольорів в CSS, що створює складності в запам'ятовуванні цих назв. Кольори в Bootstrap розподілені трохи інакше. В Bootstrap існує кілька ключових слів або певних назв для кольорів, яким відповідають відповідні кольори.

Наприклад, ключове слово `primary` завжди відповідає синьому кольору, `success` зеленому, `danger` червоному і так далі (рис.6.3).

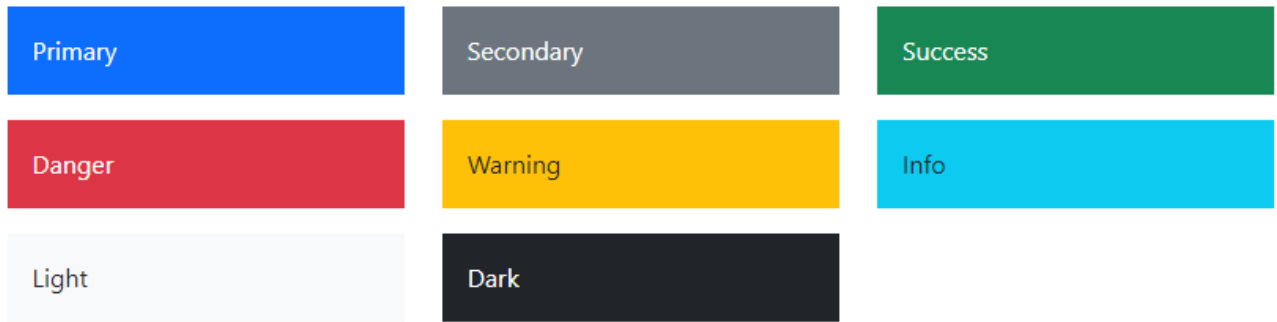


Рис.6.3. Базові кольори Bootstrap

### Створення кольорового тексту

Для того, щоб відображати текст в певний колір необхідно використовувати клас `.text-*`, де "\*" це, відповідне до кольору ключове слово. Наприклад:

```
<p class="text-primary">Hello World</p>
```

Hello World

```
<p class="text-secondary">Hello World</p>
```

Hello World

```
<p class="text-success">Hello World</p>
```

Hello World

```
<p class="text-danger">Hello World</p>
```

Hello World

```
<p class="text-warning">Hello World</p>
```

Hello World

```
<p class="text-info">Hello World</p>
```

Hello World

```
<p class="text-light">Hello World</p>
```

Hello World

```
<p class="text-dark">Hello World</p>
```

Hello World

Застосування класу `text-light` надає світлий колір. Такий ефект відмінно підходить, якщо є темний задній фон.

### Типографіка

Bootstrap встановлює основні глобальні стилі відображення, типографіки та посилання.

Доступні всі заголовки HTML `<h1>` до `<h6>`.

**h1. Заголовок Bootstrap**

**h2. Заголовок Bootstrap**

**h3. Заголовок Bootstrap**

**h4. Заголовок Bootstrap**

**h5. Заголовок Bootstrap**

**h6. Заголовок Bootstrap**



## Display

Якщо є потреба відобразити заголовок більшим і неординарним використовують клас `.display-`

```
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
<h1 class="display-5">Display 5</h1>
<h1 class="display-6">Display 6</h1>
```

Дисплей 1

Дисплей 2

Дисплей 3

Дисплей 4

Дисплей 5

Дисплей 6

## Lead

Якщо потрібно зробили більш виразний абзац використайте клас `.lead`.

```
<p class="lead">
  Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor.
  Duis mollis, est non commodo luctus.
</p>
```

Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Duis mollis, est nonmodo luctus.

## Візуальні елементи тексту

```
<p>You can use the mark tag to <mark>highlight</mark> text.</p>
<p><del>This line of text is meant to be treated as deleted text.</del></p>
<p><s>This line of text is meant to be treated as no longer accurate.</s></p>
<p><ins>This line of text is meant to be treated as an addition to the
document.</ins></p>
<p><u>This line of text will render as underlined</u></p>
<p><small>This line of text is meant to be treated as fine print.</small></p>
<p><strong>This line rendered as bold text.</strong></p>
<p><em>This line rendered as italicized text.</em></p>
```

Ви можете використовувати тег `mark` для **виділити** текст.

Цей рядок тексту має розглядатися як віддалений текст.

Цей рядок тексту вважається неточним.

Цей рядок тексту має розглядатися як доповнення до документа.

Цей рядок тексту буде підкреслено

Цей рядок тексту слід розглядати як дрібний шрифт.

**Цей рядок виділено жирним шрифтом.**

*Цей рядок виділено курсивом.*

## Скорочення

```
<p><abbr title="attribute">attr</abbr></p>
<p><abbr title="HyperText Markup Language"
class="initialism">HTML</abbr></p>
```

attr

HTML

## Цитати

Для цитування блоків контенту з іншого джерела в `<blockquote class="blockquote">`.

```
<blockquote class="blockquote">
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere
erat a ante.</p>
</blockquote>
```

Lorem ipsum dolor sit amet, conctetur adipiscing elit. Ціла кількість посуере ерат анте.

## Назва джерела

```
<figure>
  <blockquote class="blockquote">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere
erat a ante.</p>
  </blockquote>
  <figcaption class="blockquote-footer">
    Someone famous in <cite title="Source Title">Source Title</cite>
  </figcaption>
</figure>
```

Сам біль - це любов елое число посуере эрат анте.

— Хтось відомий у назві джерела

## Списки

### Без стилю

```
<ul class="list-unstyled">
  <li>Nulla volutpat aliquam velit
    <ul>
      <li>Phasellus iaculis neque</li>
      <li>Purus sodales ultricies</li>
    </ul>
  </li>
  <li>Faucibus porta lacus fringilla vel</li>
</ul>
```

Полегшення в аликвоте ціни гравців  
Нулла волутпат аліквам велить

- Phasellus iaculis neque
- Чисті члени класу
- Vestibulum laoreet porttitor sem
- І комфортних безкоштовних вихідних в Біля входу в оз

Енеан сиджу амет ерат нунк

### В рядок

```
<ul class="list-inline">
  <li class="list-inline-item">Lorem ipsum</li>
  <li class="list-inline-item">Phasellus iaculis</li>
  <li class="list-inline-item">Nulla volutpat</li>
</ul>
```

Lorem Ipsum Phasellus iaculis Нуль волутпат

## Зображення

### Створення гнучкого (responsive) зображення

Для того щоб зображення не залежало від використовуваного пристрою, потрібно додати клас `.img-fluid` до елемента зображення.

```

```

### Створення заокруглених країв у зображення

Bootstrap 5 надає можливість реалізувати зображення із заокругленими краями. Просто додати клас `rounded`

```

```

У цьому прикладі заокруглені краї будуть у всіх сторін зображення. Однак є можливість зробити заокруглений край тільки на якійсь одній стороні. Наприклад, застосування класу `rounded-top`, створить зображення тільки з заокругленим верхом. Клас `.rounded-bottom` зробить те ж саме з протилежною стороною.

```

```

### Створення повністю круглого зображення

Bootstrap надає можливість по створенню круглих зображень завдяки класу `rounded-circle`.

```

```



### Вирівнювання зображення по правому чи лівому краю

Вирівнювання - це важлива складова дизайну. Для вирівнювання зображень по правому чи лівому краю необхідно використати один з класів: `float-start` або `float-end` (рис.6.4).

```
  

```

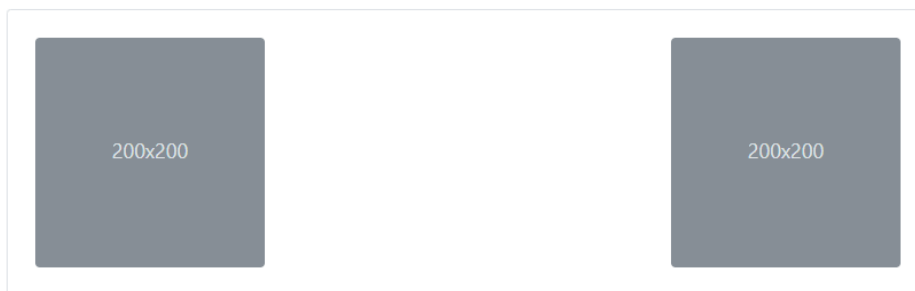


Рис.6.4. Вирівнювання зображення краях батьківського блоку

### Центрування зображення

Перед центруванням зображення варто переконатися, що зображення, не займає всю доступні ширину або 100%. Якщо є клас `img-fluid`, то його слід прибрати (рис.6.5).

Існує два способи центрування зображення в Bootstrap:

#### 1. Використання класу `.mx-auto`

```

```

2. Використання класу `.text-center`. Клас `text-center` додають в батьківський елемент `<div>`, а не в елемент зображення `<img>`.

```
<div class="text-center">  
    
</div>
```

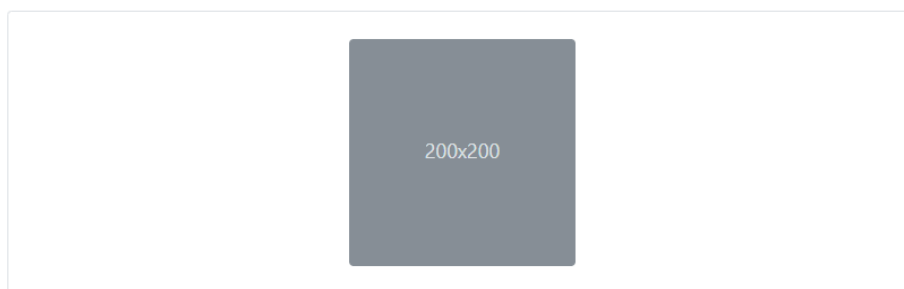


Рис.6.5. Центрування зображення у батьківському блоці

### Таблиці

Через широке використання елементів `<table>` таблиці Bootstrap мають базовий клас `.table` до будь-якого елемента `<table>` (рис.6.6). Стилізацію можна розширювати за допомогою додаткових класів (рис.6.7).

```
<table class="table"> ... </table>
```

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Рис.6.6. Таблиця з базовим класом .table

```

<table class="table-primary">...</table>
<table class="table-secondary">...</table>
<table class="table-success">...</table>
<table class="table-danger">...</table>
<table class="table-warning">...</table>
<table class="table-info">...</table>
<table class="table-light">...</table>
<table class="table-dark">...</table>

```

Class	Heading	Heading
<b>Default</b>	Cell	Cell
<b>Primary</b>	Cell	Cell
<b>Secondary</b>	Cell	Cell
<b>Success</b>	Cell	Cell
<b>Danger</b>	Cell	Cell
<b>Warning</b>	Cell	Cell
<b>Info</b>	Cell	Cell
<b>Light</b>	Cell	Cell
<b>Dark</b>	Cell	Cell

Рис.6.6. Стилїзована таблиця

## Порядок роботи

1. Ознайомитися з документацією Bootstrap, завантажити дистрибутиви.
2. Використовуючи відеоуроки створити тестову сторінку. Застосувати на сторінці типові елементи сторінки - верхня та нижня частини сторінки, навігація, форма та кнопки, ліва та центральна колонки, іконковий шрифт, випадний список тощо.
3. Протестувати тестову сторінку на адаптивність при різних розмірах екрану та пристроїв. Протестувати сторінку на різних браузерах та пристроях.

4. Ознайомитися з прикладами різних структур сторінок і з'ясувати особливості застосовних класів.
5. Розмістити тестову сторінку на хостингу. Результат виконання демонструвати в Інтернеті.
6. Під час виконання роботи робити скріншоти основних етапів роботи. Оформити звіт.

### *Зміст звіту*

1. Назва та мета виконання лабораторної роботи.
2. Вказати доменну адресу, назву і короткий опис проведених дій.
3. Скріншоти послідовності створення сторінки з коротким описом кожного з етапів.
4. У висновку оцінити зручність використання фреймворка Bootstrap.

### *Контрольні питання*

1. Які основні переваги надає Bootstrap для веб-розробників?
2. Які базові модулі складають фреймворк?
3. Які якості притаманні шаблонам сторінок на базі Bootstrap?
4. Яким чином Bootstrap пропонує зробити розподілення інформації по кількох колонках?
5. Які класи Bootstrap надають змогу форматувати фрагменти тексту?
6. Які візуальні акценти можна застосувати до фрагментів тексту?
7. Які файли з бібліотеки Bootstrap є важливими та першочерговими для під'єднання?
8. Назвіть динамічні елементи, які можна вирішувати скриптами бібліотеки.
9. Яку зовнішню JS-бібліотеку обов'язково потрібно підключити до сайту Bootstrap? Чому?
10. Чим відрізняються різні версії Bootstrap між собою?

### *Використані джерела*

1. Офіційний сайт Bootstrap 5 - <https://getbootstrap.com/>
2. Офіційний сайт Bootstrap 5 переклад російською <https://getbootstrap.su>
3. Спільнота Bootstrap 5 - <https://bootstrap5.ru/>
4. Довідник Bootstrap 5 - <https://www.w3schools.com/bootstrap5>
5. Знайомство з Bootstrap 5 - <https://timeweb.com/ru/community/articles/znakomstvo-s-bootstrap>
6. Уроки по Bootstrap 5 - [https://www.youtube.com/watch?v=7EKg\\_lgTztU](https://www.youtube.com/watch?v=7EKg_lgTztU)
7. Bootstrap Icons - <https://icons.getbootstrap.com>
8. Приклади Bootstrap-сторінок - <https://getbootstrap.com/docs/5.0/examples/>
9. Шаплони Bootstrap-сайтів - <https://bootstrapmade.com/bootstrap-5-templates/>
10. Шаплони Bootstrap-сайтів - <https://graygrids.com/bootstrap-5-templates/>

# Лабораторна робота №7. Системи керування контентом

**Мета роботи.** Ознайомитися з можливостями систем управління контентом CMS. Набути практичних навичок щодо адаптації базових шаблонів під визначену структуру сторінки. Додати до базових сторінок різні елементи: навігація, кнопки, таблиці, форми. Спромогтися змінити фрагменти коду або стилів, для налаштування під тему сторінки.

## Системи керування контентом

Content Management System (Система управління веб-вмістом) - програмний комплекс, що надає функції створення, редагування, контролю та організації веб-сторінок. CMS використовуються для створення веб-проектів різного призначення та складності і націлені на користувачів, мало знайомих з програмуванням.

### Основні функції CMS:

- Створення - користувач отримує універсальний інструмент зі створення контенту.
- Управління - можна обмежувати доступ до інформації, додавати, редагувати, видаляти і багато іншого.
- Публікація - стаття повністю адаптується до дизайну сайту, потрібно лише внести її у відповідне поле.
- Подання - можна змінювати кольорову палітру, розташування і багато іншого будь-якого елемента на сайті.

### Переваги використання CMS:

- Зручність. Дружній інтерфейс і зрозуміле управління є основними плюсами цього програмного забезпечення.
- Економія. Не знадобиться вдаватися до послуг фахівців і платити за це додаткові гроші.
- Безпека. Завжди можна зробити відкат і будь-які допущені помилки не будуть критичними. Також, в системи закладають протидії атакам хакерів, щоб інформація була в безпеці.
- Документація. Модулі мають help-файли, де розміщено довідкову інформацію щодо функцій даної CMS.
- Наявність шаблонів. Більшість CMS надають численні шаблони. Крім того, розробники постійно оновлюють систему і надають нові види.
- Функціональність. Кожен розділ або підрозділ має власні налаштування та конфігурації. Відповідно до вимог можна змінити шаблонний сайт до найдрібніших деталей.
- Комплексність. Можна створювати окремі вкладки і надавати контроль над ними іншим користувачам (стрічка новин, блоги та ін.).
- Незалежність. Користувач CMS не залежить від розробників, конфігурацію можна змінювати не потребуючи дозволу або згоди розробника.

- Можливість розширення. Можна змінити версію CMS і при цьому не втратити інформацію. В процесі перенесення сайт буде функціонувати як і раніше.
- Привабливий зовнішній вигляд. Різна колірна гамма і зручна панель дозволить отримувати естетичне задоволення від роботи.
- Керованість. CMS проста у використанні. За допомогою пару кліків можна змінити структуру, поміняти місцями пункти, розділи, кореневу папку, внести зміни у і багато іншого.

Кожна CMS незалежно від виробника створюється з врахуванням багатьох вимог користувача. Сучасний ринок представлено різними CMS. Деякі системи орієнтовані на вирішення конкретних завдань, а інші стали універсальними і практичними для будь-якого типу контенту. Деякі CMS мають багато функціональних розділів і підрозділів, інші, навпаки, складаються з єдиної системи.

### *Безкоштовні популярні системи управління*

CMS можуть бути як платними, так і безкоштовними. Виробники часто закладають модулі з відкритим і закритим кодом, тобто, можна вносити власні зміни в роботу CMS. Безкоштовні продукти не поступаються за якістю платних програмах. Навіть навпаки, у більшості випадків вільні CMS набагато більш ефективніше і продуктивніше, оскільки над ними працюють не тільки фахівці, а й армія їх прихильників (рис.7.1).

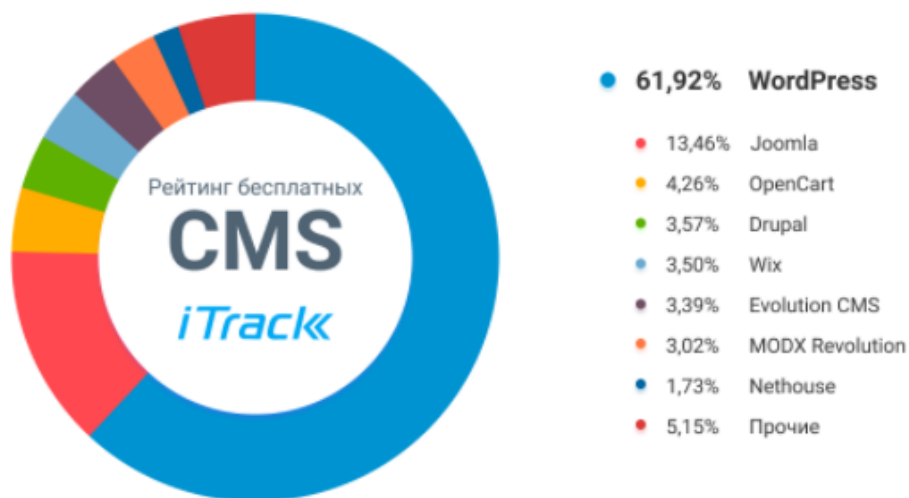


Рис.7.1. Рейтинг безкоштовних CMS у світі

### **WordPress (<https://wordpress.com/>)**

WordPress є популярною системою, що дозволяє створювати сайти різного типу, інформаційні, новинні тощо, але в першу чергу, це найкраща система для блогів. Якщо створювати власний блог, або нескладний сайт інформаційного типу, на який періодично додаються нові статті, які можна коментувати, то саме WordPress - непоганий вибір.





### *Переваги*

- Простота. Створення сайту на WordPress практично не вимагає ІТ-навичок. Багато хостинг-провайдерів дозволяють встановити CMS в один клік. Опублікувати новий контент не складніше, ніж друкувати щось в програмі Word. Про роботу з WP написано багато статей, знято детальні відео, опубліковано тисячі книг. На будь-яке питання щодо WordPress можна знайти відповідь.
- Маса безкоштовних можливостей. Гнучкі теми для створення унікального дизайну, плагіни для розширення функціоналу - все це можна знайти, не виходячи з адміністративної панелі сайту.
- Постійний розвиток. WordPress створили професійні розробники і спільно з тисячами волонтерів-ентузіастів постійно покращують платформу. Після кожного релізу CMS стає швидше, зручніше, безпечніше.

### *Недоліки*

- Порівняно обмежені можливості налаштувань. Це зроблено спеціально, щоб не заплутувати користувачів-початківців. Але просунуті користувачі не завжди можуть реалізувати якусь витончену задумку.
- Сайт на WordPress легко створити, але так само легко і зіпсувати. Значна частина відеоуроків і гайдів по WordPress записана людьми, які дуже далекі від веб-розробки. Наприклад, вони радять як вносити правки в код, але не попереджають, що всі правки зникнуть після поновлення ядра CMS або теми.
- «Движок» постійно намагаються зламати. У самій CMS WordPress практично ніколи не було серйозних дірок в безпеці, головна загроза для користувача - тисячі плагінів «движка». Саме через плагіни в більшості випадків зловмисники зламують сайти.

### **Joomla (<https://www.joomla.com/>)**

Joomla - друга за популярністю CMS в світі. Безкоштовна, проста, доступна і надійна, але з нею дещо складніше розібратися. Її не так просто встановити, налаштувати і наповнити контентом. Для створення простого сайту варто подивитися відеоуроки на YouTube.



### *Переваги*

- Для цієї платформи існують зручні багатофункціональні програми для створення інтернет-магазинів різної складності: VirtueMart, Joomshopping, Tienda тощо. Створити повнофункціональний інтернет-магазин на Joomla простіше і швидше, ніж наWordPress.
- Сайт на Joomla можна оптимізувати без додаткових плагінів. Користувач може прописати мета-теги і налаштувати URL (адреси сторінок, які відображаються в рядку браузера).
- У движку відразу доступно кешування, що пришвидшує завантаження сторінок, підвищує позиції сайту.
- Сайт можна редагувати не заходячи в адміністративну панель. Доступно так зване front-end редагування контенту.

### Недоліки

- Об'єктивні недоліки у Joomla такі ж, як і у WordPress - занадто багато навчальної інформації, серед якої даремні і навіть шкідливі поради та уроки.
- Дещо обмежені можливості.
- Велика кількість бажаючих знайти слабкі місця в коді і навчитися зламувати сайти на цій CMS.
- Якщо порівнювати з WordPress, то для Joomla існує значно менше додатків (плагінів і тем).

### Drupal (<https://drupal.com/>)

Drupal - це CMS абсолютно іншого рівня. Якщо Joomla та WordPress - прості, але обмежені рішення, то Drupal - платформа, яка дозволяє втілити практично будь-який задум, але вимагає певних навичок. Інтернет-магазини, соцмережі, великі корпоративні сайти - все це успішно створюється на Drupal.



### Переваги

- Високий рівень безпеки (без будь-яких складних налаштувань і установки додаткових модулів). Багато розробників вважають це рішення найзахищенішою CMS на php.
- Зручна оптимізація (не вимагає додаткового встановлення, SEO-налаштування доступні в стандартній комплектації CMS).
- Гнучкі налаштування прав користувачів. Можна вибирати, які можливості надати кожному користувачеві (в багатьох інших CMS цей набір можливостей стандартний).
- Велика кількість готових потужних безкоштовних рішень. Для електронної комерції, наприклад, є такі платформи, як Ubercart. Вони перетворюють Drupal в зручний багатофункціональний інтернет-магазин і не вимагають складних налаштувань і знань програмування.
- Адаптація для мобільних пристроїв. Починаючи з 8-ї версії, всі теми для Drupal добре відображаються на всіх популярних мобільних пристроях.
- Десятки і сотні видів контенту. Без додаткових правок коду можна створити такий тип контенту, як наприклад, опитування, статистичні дані, відео тощо.

### Недоліки

- Ціна і рівень складності. Непідготованому користувачу нелегко створити проект на цій CMS. Професійному розробнику, доведеться платити більше, ніж розробнику на WordPress або Joomla.
- Преміум-теми на Drupal коштують на порядок дорожче, ніж теми на WP.

### MODX (<https://modx.com/>)

MODX схожа на Drupal за можливостями для звичайного користувача без диплома програміста. MODX підійде для розробки будь-якого сайту: блогу, форуму, величезного порталу, сервісу з онлайн-платежами тощо. У CMS вбудовано багато речей, які спрощують роботу веб-розробника.



### *Переваги*

- Можна вибрати багато типів контенту.
- Безпека. Сайти на MODX більш захищені, ніж ресурси на більшості інших CMS.
- SEO без додаткових модулів і налаштувань (як і у Drupal).
- Створення складних багатофункціональних інтернет-магазинів за допомогою Foxu Cart.
- Підтримка багатомовних сайтів.

### *Недоліки*

- Фахівців з MODX (як і з Drupal) менше, ніж девелоперів, які працюють на більш простих CMS, тому їх послуги дорожче. І знайти їх складніше.

### **OpenCart (<https://www.opencart.com/>)**

OpenCart - зручна безкоштовна платформа для інтернет-магазинів, яку можна встановити і налаштувати буквально за 30 хвилин. Це ідеальне рішення для тих, кому потрібен інтернет-магазин швидше і дешевше.



### *Переваги*

- Функціональний і простий редактор.
- Можливість редагувати мета-теги після установки стандартної комплектації.
- Можливість відстежувати дані про продажі, продукти, переваги споживачів і ефективність реклами (без використання CRM).

### *Недоліки*

- Ця платформа створена виключно для інтернет-магазинів. На ній неможливо створити інший вид сайту.
- Недостатні стандартні можливості для SEO-оптимізації.

- **Magento (<https://magento.com/>)**

Спочатку розроблялася для надання послуг в сфері електронної комерції. Розробники позиціонують свій продукт як максимально простий в освоєнні, але початківцям, в силу відсутності вбудованого візуального конструктора, все ж доведеться вивчити його особливості та розібратися в функціоналі. Для більш просунутих користувачів робота з Magento не буде проблемою.



### *Переваги*

- Покращена система кешування Varnish Cash, що пришвидшує завантаження сторінок.
- Повна інтеграція з Google Analytics - повна звітність по магазину і інтеграція з такими сервісами, як Google AdWords.
- Розширене управління користувачами - кілька користувачів можуть мати один обліковий запис, а для деяких з них можна встановлювати обмеження.
- Висока швидкість роботи навіть при великій кількості відвідувачів.
- Адаптивність шаблонів під мобільні пристрої.

## Недоліки

- Складний процес встановлення.
- Необхідність в доробках - якщо робити інтернет-магазин індивідуальним і неповторним, тоді доведеться або вчити ZendFramework, на базі якого створено Magento, або наймати фахівців, вартість послуг яких також коштує грошей.
- Необхідність в якісному хостингу - для установки Magento не підійде простий і дешевий хостинг.
- Мала кількість додатків, плагінів і розширень.

## Інші безкоштовні CMS

- **TYPO3** (<https://typo3.org/>) - система з відкритим вихідним кодом і вільною ліцензією. Є гнучкою розширюваною системою з великою кількістю модулів і функцій. Написана на PHP, для зберігання даних використовує реляційну базу даних: MySQL, Oracle Database, PostgreSQL та інші. Працює на серверах Apache та IIS, на більшості операційних систем: Linux, Microsoft Windows, FreeBSD, Mac OS X і OS/2.
- **PrestaShop** (<https://www.prestashop.com/>) - система призначена для малого та середнього бізнесу і має більше 310 стандартних функцій для швидкого створення функціонального магазину.
- **Moodle** (<https://moodle.org/>) - система управління навчанням (LMS), система управління курсами (CMS), віртуальним навчальним середовищем (VLE) або платформа для навчання, яка надає розвинутий набір інструментів для комп'ютеризованого навчання, в тому числі дистанційного.

## Поради щодо вибору безкоштовної CMS

Вибір доцільної CMS залежить від різних причин, в першу чергу від тих цілей і завдань, які поставлено при створенні сайту, необхідної функціональності, і готовності долати труднощі в освоєнні конкретної системи.

Для стандартних сайтів підійдуть Joomla чи WordPress. WordPress буде доречним для блогу, нескладного інформаційного сайту, Joomla - для сайту-порталу з різними додатковими можливостями. Якщо вимоги більш специфічні і розробник має час на освоєння системи і розробку сайту, можна використати інші CMS, що відповідають цілям і завданням сайту.

Для складного завдання можна вибрати комерційну CMS, це буде більш зручно та надійно.

## Початок роботи з WordPress.com

### Основи

На кожному сайті WordPress.com є два основних компоненти.

- **Вміст.** Вміст сайту включає сторінки, записи та медіафайли. Можна додавати або редагувати сторінки та записи, перейшовши в меню *Мої сайти* → *Записи/Сторінки*. Для внесення змін необхідно використати редактор блоків.
- **Налаштування.** Для внесення змін по всьому сайту, наприклад, для редагування меню, налаштування кольорів, додавання віджетів та іншого, необхідно

використовувати Конфігуратор. Щоб відкрити конфігуратор, виберіть *Мої сайти* → *Зовнішній вигляд* → *Установки*.

## Додати вміст

Найпершим завданням є додавання вмісту. Для цього виберіть *Мої сайти* → *Сторінки* та *Мої сайти* → *Записи*. У цих розділах можна додати вміст на веб-сайт (рис.7.2).

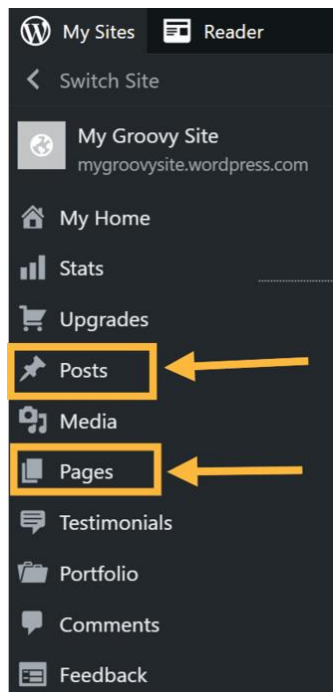


Рис.7.2. Вибір розділів для додавання контенту

*У чому різниця між сторінкою та записом?*

**Сторінка** найкраще підходить для вмісту, який змінюватиметься рідко. Хорошими прикладами сторінок є головна сторінка, сторінка відомостей, сторінка з контактами або сторінка, де міститься список продуктів або послуг.

**Записи** нагадують статті новин. Хорошими прикладами записів є оглядові статті про новинки галузі, особистий журнал, відгуки про кінофільми та інші подібні коментарі чи розповіді. На кожному сайті WordPress.com можна додати блог, але цю функцію можна використовувати тільки за необхідності.

## Додати першу сторінку

Відкрийте *Мої сайти* → *Сторінки* та натисніть *Додати нову сторінку*. Відкривається редактор блоків. Під час створення нової сторінки відображається опція, яка дозволяє вибрати готовий макет. Можна вибрати існуючий шаблон і змінити його або почати з порожньої сторінки (рис.7.3).

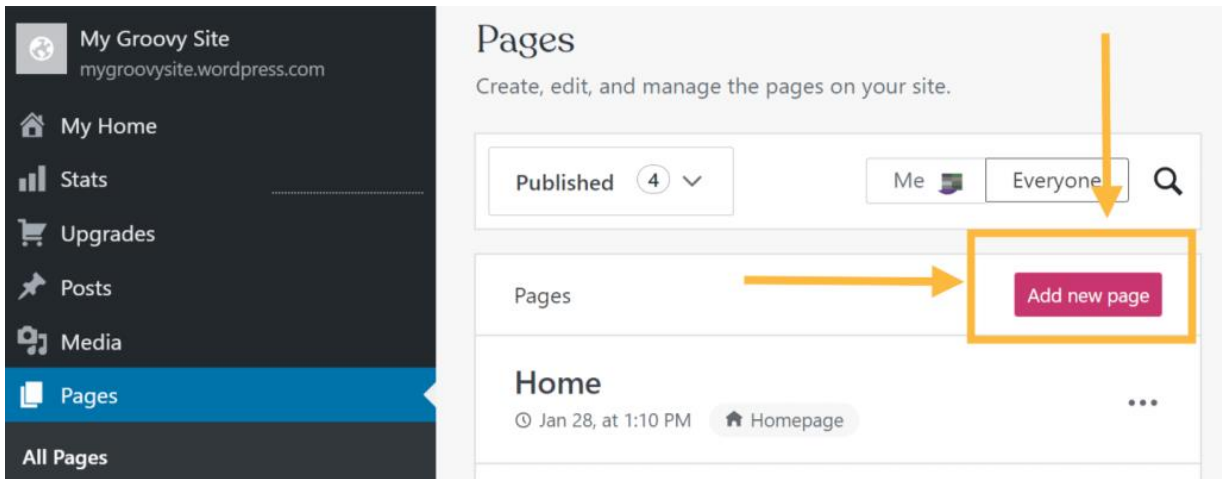


Рис.7.3. Створення нової сторінки

В правій стороні редактора відображаються параметри сторінки. Тут можна побачити статус сторінки, додати рубрики та теги у записі, вказати обране зображення та багато іншого (рис.7.4).

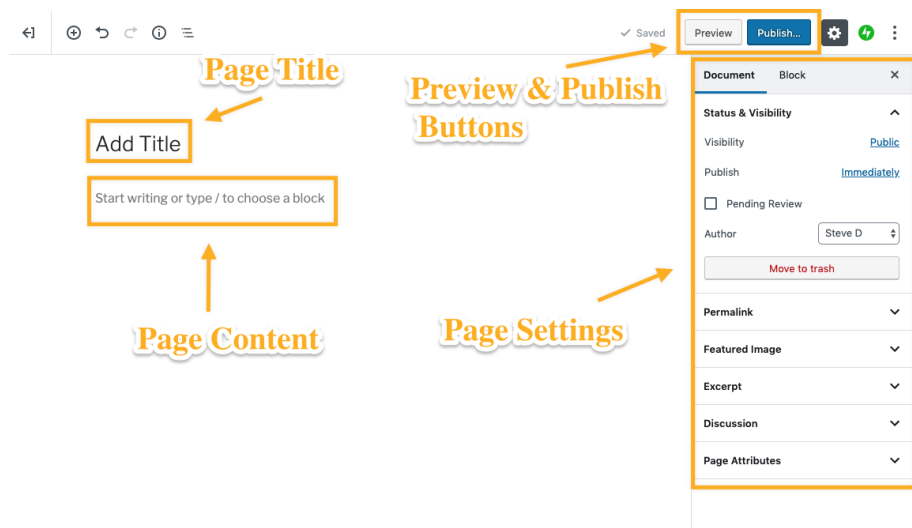


Рис.7.4. Опції для редагування вмісту сторінки

Редактор блоків містить такі компоненти. Якщо ви не бачите панель налаштувань праворуч, натисніть значок шестерні поруч із кнопкою *Опублікувати*.

### Редактор блоків

Для створення сторінок і записів використовується редактор блоків, який дозволяє працювати з готовими блоками для веб-сайту. Можна додавати вміст до сторінки, вставляючи різні блоки. Для цього можна натиснути значок інструмента вставки блоків +, який відображається у різних областях редактора (рис.7.5).

Для додавання блоку та вибору його типу можна використовувати різні способи:

- Натиснути + у лівій частині будь-якого порожнього блоку.
- Натиснути + у верхньому лівому куті редактора.

- Натиснути клавішу *Введення* (клавішу перекладу рядка), перебуваючи у вибраному блоці, щоб створити під ним новий блок.
- Натиснути + у верхній частині блоку, щоб додати блок над ним.
- Клацнути три точки над виділеним блоком та вибрати, куди додати новий блок: вище або нижче поточного.

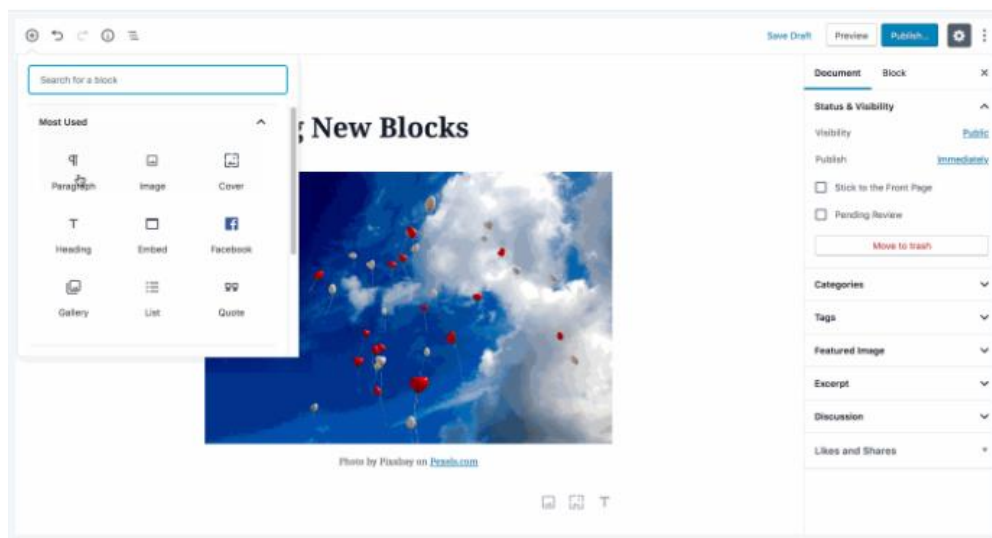


Рис.7.5. Редактор блоків

Можна додавати різні блоки: абзаци, зображення, галереї, стовпці та таблиці. Список варіантів дуже великий!

### Список блоків

Нижче наведено деякі типові блоки, які можна використовувати (рис.7.6):

- Блок «*Абзац*»: стандартний блок, який можна використовувати для додавання звичайного тексту до запису або на сторінку.
- Блок «*Зображення*»: щоб додати окремі зображення до запису або на сторінку.
- Блок «*Обкладинка*»: щоб додати текст зверху зображення.
- Блок «*Стовпці*»: дозволяє розміщувати блоки поруч.
- Блок «*Заголовок*»: додає текст заголовка до запису або на сторінку.

Якщо натиснути піктограму інструмента вставки блоків +, можна виконати пошук блоків різного типу або прокрутити вниз і розгорнути кожний список, щоб побачити наявні елементи. Додавши блок, ви можете змінювати його розташування за допомогою стрілок ліворуч або праворуч або перетягуванням.

При додаванні блоку та його виборі на правій стороні редактора з'являться додаткові параметри блоку. Кожен блок містить свої налаштування, які можна змінювати. Перейдіть на вкладку «*Документ*» нагорі, щоб повернутися до загальних параметрів сторінки.

Ви можете видалити блок, вибравши його, натиснувши меню кнопки з трьома крапками і вибравши команду *Видалити блок*.

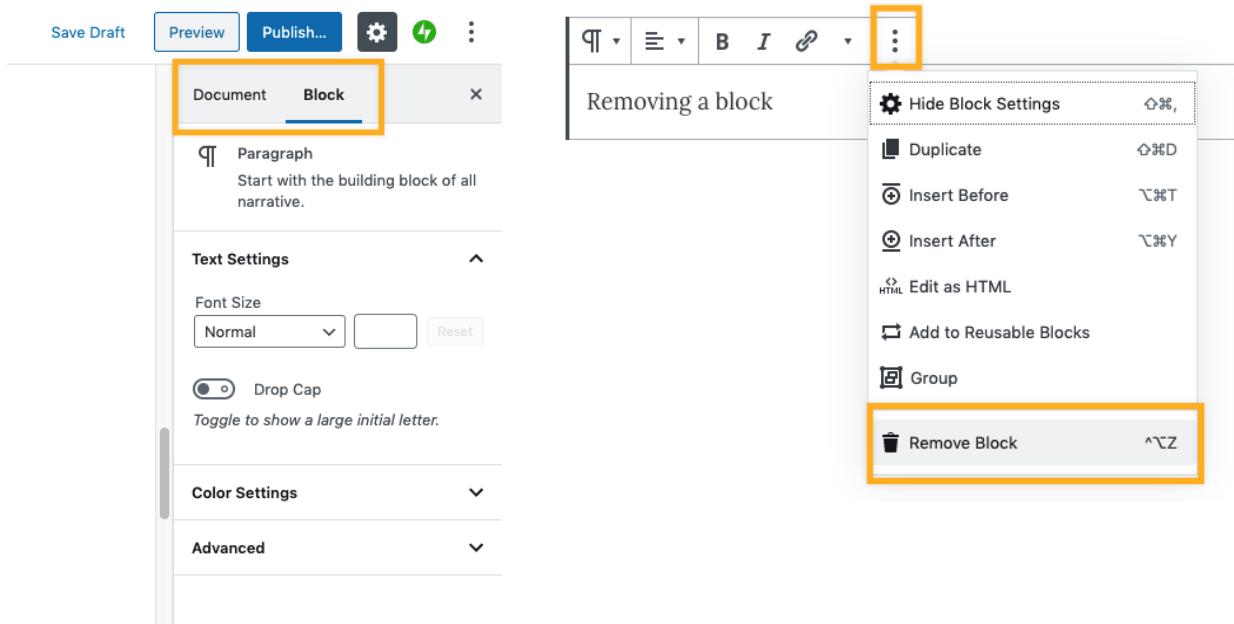


Рис. 7.6. Опції для роботи з блоками

### Порядок використання редактора блоків

Якщо вас влаштовує зовнішній вигляд сторінки, натисніть *Публікувати* у верхньому правому куті. Бажаєте додати інші сторінки? Тоді опублікуйте їх, а потім перейдіть до наступного кроку. Якщо ви хочете додати сторінку блогу, створіть та опублікуйте сторінку під назвою «Блог» (або будь-якою іншою назвою) та залиште її порожньою.

### Конфігуратор

Тепер є кілька сторінок вмісту, з якими можна працювати. Варто переконатися у тому, що вони помітні іншим користувачам. Щоб внести ці зміни, потрібно відкрити конфігуратор (рис.7.6.). Для цього виберіть *Мої сайти* → *Зовнішній вигляд* → *Налаштування*.

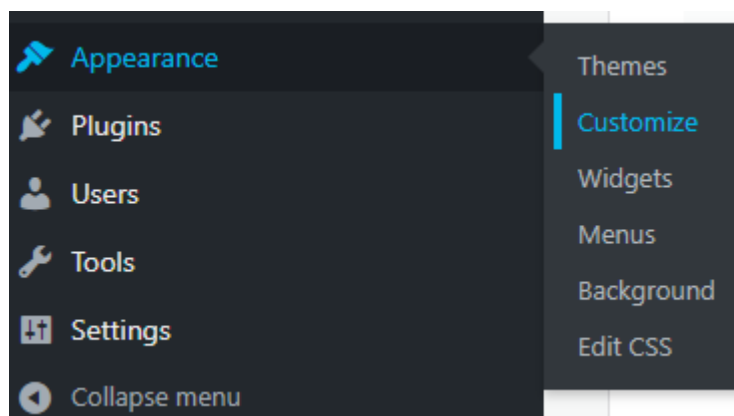


Рис.7.7. Конфігуратор для налаштування сторінок сайту

### Налаштування головної сторінки та сторінки записів

Насамперед, необхідно налаштувати головну сторінку. Для цього виберіть *Установки головної сторінки*. Тут можна вибрати режим відображення на головній сторінці списку останніх записів (*Ваші останні записи*) або *Статична сторінка* (типовий варіант для



корпоративних сайтів). При виборі статичної сторінки також можна вибрати сторінку, щоб відобразити список опублікованих записів (рис.7.8).

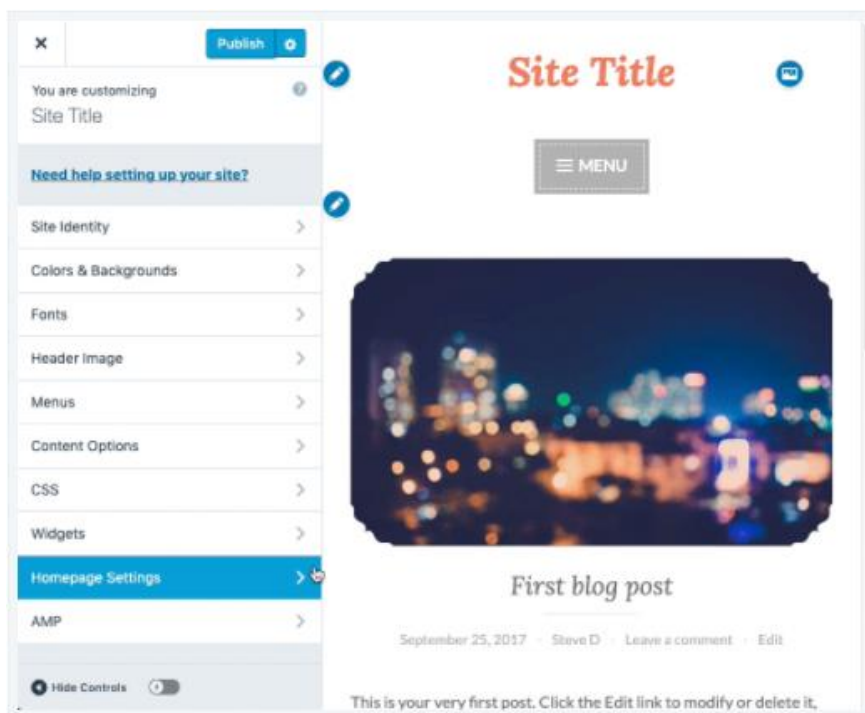


Рис.7.8. Налаштування головної сторінки та сторінки записів

## Додати меню

Тепер додамо в меню інші створені сторінки, щоб відвідувачі могли ними переміщатися. Виберіть *Меню* → *Головне* → *Додати елементи*. Відображається список опублікованих сторінок (рис.7.9). Виберіть сторінки, які потрібно додати до меню.

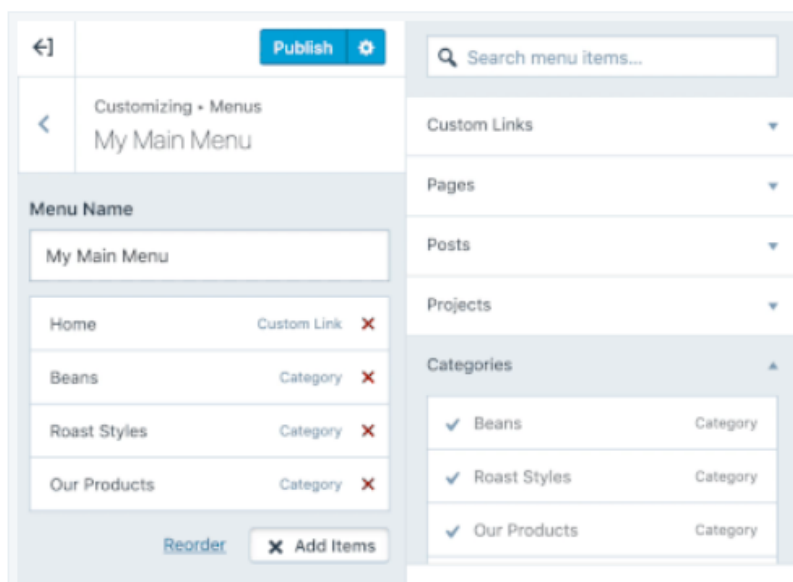


Рис.7.9. Упорядкування меню

Щоб змінити порядок меню, їх можна вибрати та перетягнути. Щоб створити підменю, виберіть його та перетягніть під інше меню. Це також можна зробити за допомогою опції *Змінити*

*порядок*. Зверніть увагу на те, як зміни відображаються в режимі попереднього перегляду сайту на правій стороні.

### Зберегти зміни у конфігурації

Після внесення будь-яких змін обов'язково натисніть «*Опублікувати*», щоб зберегти їх. Це буде схоже на дію кнопки "*Зберегти*". Якщо ви перевели сайт у режим «*Закрито*», відвідувачі не зможуть його побачити, доки ви не зміните цей режим.

### Теми

Щоб змінити теми, виберіть *Мої сайти* → *Зовнішній вигляд* → *Теми*. Теми можна міняти будь-яку кількість разів без будь-яких обмежень. При зміні теми змінюється вигляд сайту, але його вміст не видаляється. Деякі теми можуть потребувати додаткового налаштування.

Щоб переглянути доступні теми, перейдіть до розділу *Зовнішній вигляд* → *Теми* (рис.7.10).

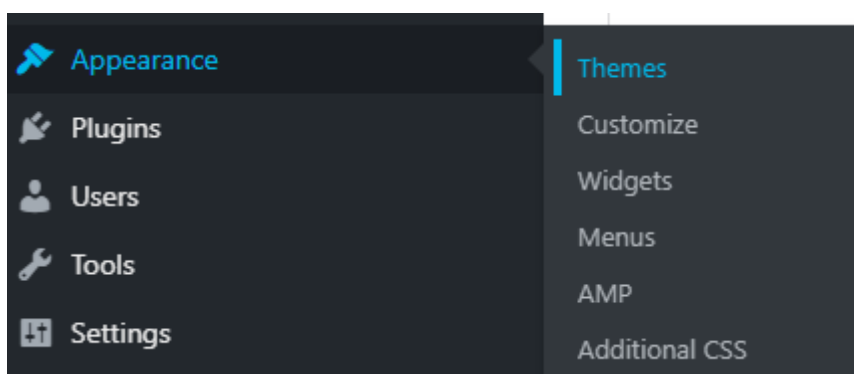


Рис.7.10. Вибір теми для зміни зовнішнього вигляду

Тут можна перейти на одну із вкладок:

- **Рекомендовані.** Тут зібрані красиві та максимально прості у використанні теми, що особливо важливо для новачків. Якщо не знаєте, яку тему вибрати, використовуйте одну з рекомендованих.
- **Популярні.** Це теми, популярні серед власників WordPress.com.
- **Мої теми** (тільки для сайтів із тарифними планами Business та eCommerce). Це теми, які були завантажені або використані на сайті.
- **Усі теми.** Всі теми доступні на WordPress.com.

### Розділ тем

Для пошуку тем за назвою використовуйте рядок пошуку у верхній частині екрана. За допомогою кнопки *Фільтри* можна звузити результати пошуку тем за функціями, кількістю колонок та предметами:

- **Функція:** такі відмінні риси тим, як фіксоване меню, заголовки, слайдер тощо.
- **Колонки:** кількість колонок у темі, розташування бічної панелі ліворуч або праворуч.
- **Предмет:** тематика сайту, наприклад, мистецтво, бізнес, новини, фотографія тощо.

## Налаштування теми

Кожна тема має такі опції.

- **Демонстрація в реальному часі:** справжній сайт, де демонструються всі можливості конкретної теми.
- **Активувати:** негайне застосування теми до вашого сайту.
- **Спробувати та налаштувати:** попередньо переглядати, як ваш вміст буде виглядати в цій темі, та внести зміни до налаштувань теми. Ваш сайт буде змінено лише при натисканні кнопки Активувати та зберегти на наступному екрані.
- **Інформація:** додаткові відомості про функції теми та інструкції з налаштування.
- **Підтримка:** звернутися за додатковою допомогою на тему.

Якщо натиснути три точки праворуч від назви теми, з'являться опції "Демонстрація в реальному часі", "Активувати", "Інформація" та "Підтримка" (рис.7.11).

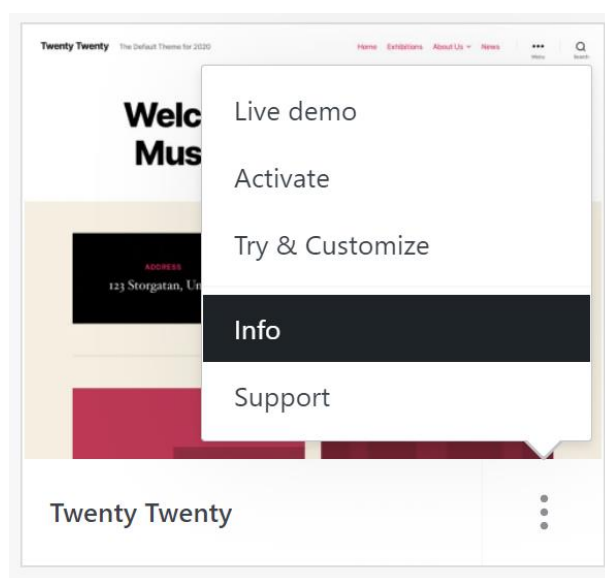


Рис.7.11. Меню додаткового налаштування теми

## Активація теми

Вибравши тему, виконайте наведені нижче дії, щоб застосувати її до сайту.

1. Виберіть тему в розділі *Зовнішній вигляд* → *Теми*.
2. Натисніть три точки поруч із темою, щоб відкрити список опцій, як показано на знімку екрана у попередньому розділі.
3. Натисніть *Активувати*, щоб встановити тему на веб-сайті.
4. Якщо така опція, вкажіть, чи хочете ви замінити поточну головну сторінку на головну сторінку нової теми. У багатьох темах є демо вміст, який може спростити налаштування теми.
5. Натисніть кнопку *Редагувати головну сторінку* або *Налаштувати сайт*, щоб розпочати роботу над новим дизайном сайту.

При активації теми весь вміст сайту зберігається. Ніякі дані не видаляються, тому перемикання абсолютно безпечне.

## Вибір теми

Коли вибір тем настільки багатий, вибрати дуже складно. Варто враховувати такі фактори.

- Чи подобається зовнішній вигляд області заголовка? Зазвичай тут відображаються назва сайту, меню та логотип.
- Чи подобається, як виглядає тема на екранах мобільних пристроїв та планшетів? Щоб це побачити, відкрийте демонстрацію в реальному часі та за допомогою селектора нагорі перемикайтеся між настільним комп'ютером, планшетом та мобільним пристроєм. Всі теми WordPress.com легко пристосовуються до різних розмірів екрана, включаючи мобільні телефони.
- Чи можна все налаштувати так, як подобається? Щоб дізнатися, як налаштовуються такі елементи, як кольори, меню та зображення заголовка, використовуйте опцію *Спробувати та налаштувати*.

## Налаштування теми

Для кожної теми на WordPress.com є посібник, в якому описуються її особливості та їх використання. Щоб знайти посібник користувача до своєї теми, перейдіть до розділу *Зовнішній вигляд* → *Теми*. Ваша тема буде відображатися у верхній частині списку. Натисніть Інформація, щоб відкрити довідкову інформацію щодо даної теми (рис.7.12.):

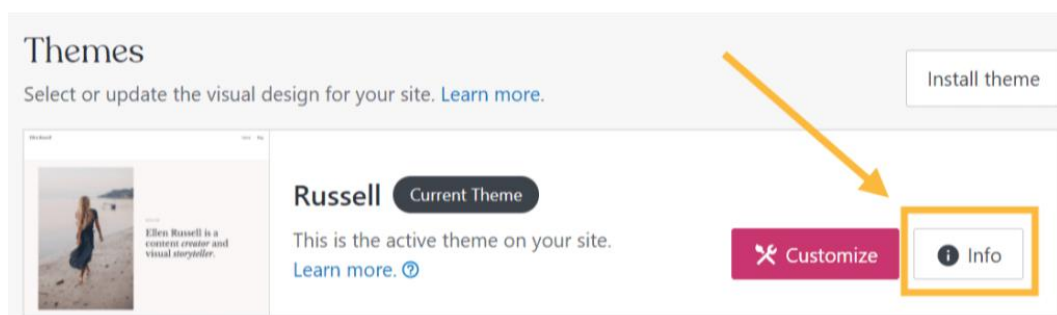


Рис.7.12. Отримання додаткової інформації про тему

## Змінити головну сторінку

Головну сторінку бачать першими ті, хто заходить на ваш сайт, тому їй варто зайнятися насамперед. Щоб редагувати головну сторінку, перейдіть до розділу *Сторінки* та виберіть головну сторінку.

Тут можна виконати такі дії:

- Замінити текст та зображення у редакторі WordPress.
- Видалити весь вміст, щоб розпочати оформлення спочатку.
- Додати патерни блоків – готові модулі вмісту, які можна легко вставляти та налаштовувати.

Якщо вам не подобається поточний дизайн головної сторінки, створіть новий або виберіть один із макетів сторінок. Щоб переглянути готові макети, перейдіть до розділу *Сторінки* та натисніть *Додати нову сторінку*. Потім у списку пакетів виберіть *Головна*. Ви побачите нові дизайни основних сторінок, які можна використовувати на сайті.

## Налаштування теми

Виберіть *Зовнішній вигляд* → *Налаштувати*, щоб настроїти оформлення та функції теми. Тут можна виконати такі дії:

- Змінити назву сайту, ключову фразу, логотип та значок сайту.
- Налаштувати колірну схему сайту.
- Змінити шрифти.
- Налаштувати меню навігації.

Які параметри можна змінити в розділі *Налаштувати*, залежить від теми. Вивчіть всі налаштування та змініть потрібні.

Після завершення натисніть кнопку *Зберегти зміни*.

## Плагіни

Всі сайти WordPress.com мають великий набір вбудованих можливостей, які повинні охоплювати всі потреби. Крім того, сайти з тарифними планами Business і eCommerce дозволяють встановити сторонні плагіни.

### Установка плагінів

Щоб встановити плагін, перейдіть до розділу *Мій сайт* → *Плагіни* → *Додати новий*. Ви побачите вітрину вибраних плагінів, а також вкладку з популярними, рекомендованими та улюбленими плагінами.

Після натискання назви плагіна ви побачите кнопку для його встановлення на сайт. Крім того, можна прочитати інформацію про плагін: які функції він пропонує, як працює, як налаштовувати його параметри (рис.7.13).

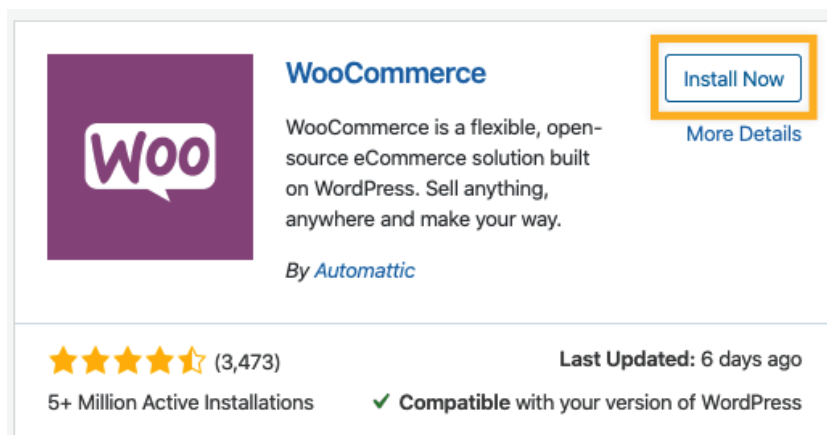


Рис.7.13. Інсталяція теми

У галереї плагінів вгорі праворуч від імені кожного плагіну розміщено кнопку "*Встановити зараз*".

Більшість плагінів створюються сторонніми розробниками, які не пов'язані з WordPress.com, для розширення можливостей WordPress.com. Перед встановленням плагіна виконайте наведені нижче перевірки.

- **Останнє оновлення:** показує, наскільки активно плагін підтримується та допрацьовується. За можливістю не користуйтеся плагінами, які довго не оновлювалися.
- **Активні установки:** показує популярність та частоту використання плагіну.
- **Рейтинг:** зворотний зв'язок від клієнтів, які встановили цей плагін.
- **Сумісність:** перевірка сумісності плагіна з версією WordPress, що використовується (сайти WordPress.com завжди автоматично використовують найновішу версію).

Під кожним плагіном у галереї плагінів показані рейтинги, кількість активних установок, дата останнього оновлення, сумісність із версією WP. Перед встановленням плагіна доцільно ознайомитися з інформацією про плагін, щоб точно знати, як ним користуватися та які завдання він виконуватиме.

### Робота із встановленими плагінами

Встановивши плагіни на сайті, перейдіть до розділу *Плагіни* → *Встановлені плагіни*, де представлений список усіх плагінів на вашому сайті.

Як правило, при додаванні плагінів до панелі адміністратора з'являються нові пункти меню. У прикладі праворуч пункти меню, додані плагінами, виділені підсвічуванням.

Деякі плагіни будуть додавати меню в існуючих опціях, таких як *Установки*, *Інструменти* або *Зовнішній вигляд*.

У списку встановлених плагінів знайдіть *Установки*, *Показати деталі*, *Документи* або *Підтримка* (кожен плагін дає різні посилання). Через ці пункти меню ви отримаєте доступ до інформації про налаштування плагіна (рис.7.14):

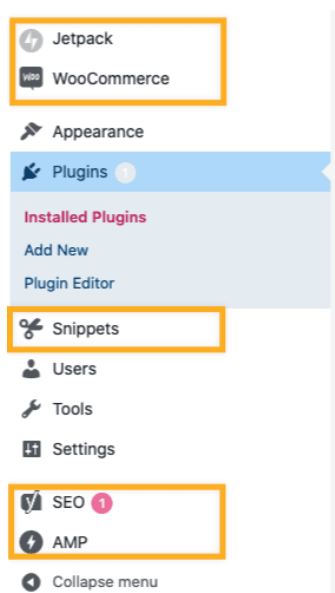


Рис.7.14. Меню вибору опцій для налаштування плагіну

### Відключення плагінів

Плагін на сайті можна деактивувати, якщо він надалі не потрібний або якщо його потрібно тимчасово відключити. Деактивація плагіна оборотна: плагін залишатиметься на вашому

сайті, і при необхідності його можна буде активувати повторно. При повторній активації плагіну всі його налаштування та вміст будуть збережені у попередньому вигляді.

- Перейдіть до розділу *Плагіни* → *Встановлені плагіни*.
- Знайдіть плагін у списку.
- Натисніть *Деактивувати*.

Також можна видалити плагін із сайту без можливості відновлення. Після деактивації плагіну стане доступна опція *Видалити*.

## *Порядок роботи*

1. Ознайомитися з наданими безкоштовними системами керування контентом. Обрати певну систему і відповідні відеоуроки для виконання роботи.
2. Встановити на хостингу обрану CMS і отримати відповідні ідентифікатори для роботи.
3. Обрати тематичний шаблон. Сформулювати інформацію щодо наповнення. Мова сайту - українська!
4. Ознайомитися з адміністративною панеллю і створити кілька сторінок.
5. Наповнити сторінки текстом, зображеннями, медіа-контентом. За потребою змінити шаблон, розміщення блоків. Додати популярні сервіси до сайту (слайдер, галерея, календар, карта тощо).
6. Оцінити легкість користування адміністративною панеллю та існуючими модулями.
7. Опублікувати сайт в Інтернеті.

## *Зміст звіту*

1. Назва та мета виконання лабораторної роботи.
2. Вказати доменну адресу, назву і короткий опис роботи.
3. Скріншоти послідовності виконання роботи з коротким описом кожного з етапів.
4. У висновку оцінити зручність використання CMS і адміністративної панелі.

## *Контрольні питання*

1. Що називається системою керування контентом?
2. Назвіть умови, за яких використовується CMS.
3. Які основні переваги при використанні CMS отримує користувач, на які застереження слід вважати?
4. Які безкоштовні CMS є популярними та їх спеціалізація?
5. Назвіть особливості CMS WordPress. Які переваги притаманні цій системі?
6. Назвіть типові проблеми, з якими стикається користувач WordPress при створенні сайту.
7. Які висуваються вимоги до WordPress хостингу?
8. Які переваги отримує користувач від застосування комерційних CMS?
9. Які критерії вибору комерційної CMS є вагомими?

10. В яких випадках користуються самописними CMS? На які моменти слід вважати користувачам?

### *Використані джерела*

1. Порівняння безкоштовних CMS: Wordpress, Joomla, Drupal - <https://uguide.ru/sravnenie-besplatnyh-cms-wordpress-joomla-drupal>
2. WordPress чи Joomla: яку CMS обрати? <https://www.hostinger.com.ua/rukovodstva/wordpress-ili-joomla-kakuy-cms-vibrat>
3. Кращі хостинги для WordPress - <https://uguide.ru/rejting-luchshij-hosting-dlja-wordpress>
4. Безкоштовний хостинг на домені WordPress - <https://wordpress.com/>
5. Створення сайту на WordPress: Покрокова інструкція - <https://wordpress.com/ru/support/getting-started-with-wordpress-com/>
6. Уроки Joomla для початківців - <https://www.hostinger.com.ua/rukovodstva/uroki-joomla-dlia-nachinauschih/>
7. Уроки Drupal - <https://www.hostinger.com.ua/rukovodstva/uroki-drupal/>



## №8 Самостійна робота

**Мета роботи.** Створити повноцінний веб-сайт. Сайт створюється згідно стандартів HTML5, CSS3 із залученням Java Script та API зовнішніх ресурсів. Розміщення сайту на безкоштовному хостингу з вибором доменного імені. Здійснити низку заходів з просуванням проекту в провідних пошукових системах

### *Етапи розробки сайту*

Розробку сайту можна порівняти з будівництвом будинку, де мають послідовно бути виконані визначені етапи: від проектної документації і закладки фундаменту до внутрішньої і зовнішньої обробки приміщення.

Етапи розробки проекту, як правило, виконуються послідовно, тому, вкрай важливо дотримуватися черговості етапів і розуміти, що будь-які несподівані і неузгоджені заздалегідь зміни чи правки можуть значно вплинути на ефективність роботи.

Роботу над кожним проектом слід проводити в строгій відповідності з приведеними нижче етапами робіт з розробки сайту.

#### **1. Передпроектна підготовка. Ідея**

Визначення тематики майбутнього сайту, з'ясування цілей і завдань, визначення і аналіз цільової аудиторії. Аналіз конкурентів. Розробка структури сайту – ескіз сайту на папері. Створення списку майбутніх тематичних розділів.

#### **2. Розробка дизайну сторінок**

##### *Дизайн-концепція сайту (креативний дизайн)*

Креативна ідея, розробка основної графічної концепції дизайну сайту на прикладі головної та другорядних сторінок. Вибір колірної гамми, художнього стилю. Підготовка макету дизайну.

##### *Технічний дизайн*

Розробка логічної і фізичної структури ресурсу. Компонування сторінки, верстальної структури. Елементи навігації

#### **3. Верстка**

- Створення шаблонів сторінок.
- Перевірка правильності написання коду.
- Верстка сторінок сайту на основі затвердженого дизайну типових сторінок.
- Збірка сторінок.

## **4. Програмна частина проекту**

### *Інтеграція сайту з системою управління*

Зараз вже жоден сучасний сайт не обходиться без системи управління, оскільки важливою є не лише красива зовнішня оболонка цього сайту, але і можливість зручної роботи з ним. Це особливо актуально для сайтів з розгалуженою структурою і великим об'ємом даних.

В цей етап входить: інтеграція з системою управління, програмування, налаштування сервера, забезпечення безпеки проекту. Контроль якості.

### *Програмування, запуск проекту*

На цьому етапі допрацьовується функціонал, що не міститься у стандартному складі системи управління.

## **5. Інформаційне наповнення сайту**

- Підготовка текстових матеріалів.
- Підготовка графічних матеріалів у растровому форматі, оптимізація картинок.
- Заповнення сторінок.

## **6. Розміщення сайту в Інтернеті**

- Організація робіт з розміщення проекту в мережі Інтернет.
- Вибір та реєстрація доменного імені.
- Вибір хостинг провайдера, розміщення сайту.

## **7. Тестування та доопрацювання**

- Тестування сайту на наявність помилок та коректність функціонування в різних браузерах (Internet Explorer, Netscape, Opera, Safari).
- Перевірка ідентичності відображення сторінок в різних екранних роздільних здатностях в різних браузерах.

## **8. Просування сайту**

В даний час вже мало просто розробити якісний сайт із зручною структурою і навігацією, важливим є забезпечення для сайту високої відвідуваності. Спромогтися цього можна не лише розміщенням реклами в засобах масової інформації, але і здійснюючи просування сайту в пошукових системах і каталогах, а також рекламою в Інтернеті.

## **9. Подальша підтримка сайту**

Перші сім етапів відносяться безпосередньо до створення сайту, решта потрібні для подальшого існування сайту.

## 8.1. Передпроектна підготовка

### Ідея

Перш, ніж розпочати роботу над сайтом, потрібно чітко уявляти, що в результаті має бути, а також перспективи розвитку сайту.

*Отже, потрібно вирішити:*

- Навіщо створювати сайт (чи потрібно це взагалі).
- Тематика сайту.
- Тип сайту (домашня сторінка, портал тощо).
- Відмінність від сайтів з такою ж тематикою (якщо це не домашня сторінка).
- Яка буде аудиторія сайту (стать, вік, інтереси).
- Якого роду сервіси будуть присутні на сайті (форум, каталог, пошта).
- Перспективи подальшого розвитку.

### Поради

- Перед тим, як створювати сайт спочатку виберіть тему для Вашого майбутнього сайту, подумайте про кількість людей, що цікавляться даною темою, про кількість сайтів-конкурентів. Чим більше сайтів з подібною темою висвічується в пошукових системах, тим менше відвідувачів буде на Вашому сайті, тим складніше буде залучати відвідувачів. Наберіть в будь-якій пошуковій системі назву майбутньої теми сайту з 2-3 головних слів і подивіться, скільки знайдено сайтів за цією темою. Кількість сайтів з певних тем сягає мільйонів і всі хочуть, щоб їх сайт був на перших позиціях. Подумайте, чи зможете ви забезпечити високу відвідуваність.
- Ніколи не робіть сайт про те, в чому не розбираєтеся. Найбільшою помилкою багатьох новачків спроби робити сайти про те, як заробляти в Інтернеті або щось подібне. Вони, не розуміючи в цьому абсолютно нічого, починають вчити інших, як треба заробляти в Інтернеті. Пошуковики не дуже люблять сайти про заробіток, і за цією темою завжди віддають пріоритет лише тим сайтам, які створені давно і мають певний авторитет.
- Добре подумайте про інформацію, яка буде розміщена на сайті. Зробіть те, в чому Ви добре розбираєтеся, що буде корисне і цікаво іншим, нехай навіть це буде якась зовсім вузька і малопопулярна тема. Чим менше за цією темою буде сайтів в пошуковиках, тим більше буде відвідувачів і тим більше сайт в майбутньому принесе дохід.
- Ніколи не копіюйте і не розміщуйте на своєму сайті інформацію з інших сайтів без згоди їх авторів, щоб в подальшому уникнути неприємностей. В Інтернеті це теж називається «крадіжкою». Власник сайту, з якого взяли інформацію без згоди, може написати в пошуковій системі, хостинг-провайдеру, де розташовано Ваш сайт та в інші інстанції. Такий сайт пошуковики можуть вилучити з свого пошуку назавжди. А якщо сайт буде видалено з пошукових систем, він буде нікому не потрібний, окрім Вас, його ніхто ніколи не знайде, він буде недоступним для інших користувачів. Хостинг-провайдер, на хостингу якого буде розташовано такий сайт, теж може його видалити.

- В подальшому, для сайту необхідна щоденна підтримка сайту в актуальному стані. Треба постійно стежити за своєчасним оновленням ресурсу і вилученням з сайту сторонньої, недостовірної або застарілої інформації.

Лише після визначення чіткого образу сайту, можна приступати до інших етапів. Певні етапи розробки сайту можуть проводитися паралельно, особливо, якщо над створенням сайту працює не одна людина, а команда в кілька чоловік.

### **Визначення типу сайту**

На сьогодні типи сайтів в основному збігаються до кількох типів: лендинг, візитка, каталог, корпоративний сайт, Інтернет-магазин.

**Лендинг** (landing page - цільова, посадкова сторінка) - це веб-сторінка, що скерована на спонукання користувача вчинити будь-яку дію: підписатися на розсилку, купити продукт, завантажити електронні матеріали. Лендинг є потужним інструментом просування продукту на ринку і підвищення ефективності реклами.

Перехід до лендинг-сторінки, зазвичай, здійснюється через email-розсилки, соціальні мережі, рекламні посилання в пошуковій видачі. Такі сторінки створюють для вузької аудиторії, їх головним завданням є утримання уваги користувача на конкретному повідомленні та його подальша конвертація в клієнта.

**Сайт-візитка** - це віртуальна візитна картка (зазвичай, до 4-5 сторінок) із загальною інформацією, яка надає ознайомчі відомості про компанію або приватну особу. Особливостями є лаконічність, інформативність, індивідуальність.

Головними цілями сайту-візитки є ознайомлення потенційних клієнтів з компанією, надання інформації про послуги і товари, а також контактні дані для зв'язку з компанією. На сьогодні візитки розробляються разом з бізнес-сторінкою в соціальних мережах і вони мають між собою перехресні посилання.

**Сайт-каталог** - це сайт, що містить повну інформацію про компанію та електронний каталог її послуг чи продукції. Відвідувачі сайту можуть ознайомитися з прайсами, описом товарів і послуг, переглянути фотографії та звернутися в компанію для здійснення покупки. Такий сайт може збільшувати кількість продажів за рахунок доступу до повної інформації про продукцію та послуги тих споживачів, які територіально віддалені від компанії-виробника.

**Корпоративний сайт** – це розширена версія сайту-візитки. Як правило, цей сайт може складатися з необмеженої кількості сторінок, і власник може завжди розширювати і доповнювати контент. Містить загальні інформаційні блоки: про компанію, її послуги, продукція, контактна інформація.

Корпоративний сайт є динамічним і потребує програмування або використання системи керування контентом. Для підтримки працездатності сайту потрібна спеціально навчена людина, що має навички у користуванні адміністративною частиною.

**Інтернет-магазин** - більше ніж просто сайт, це складна платформа, яка потребує професійного командного підходу. Вимоги до сучасного Інтернет-магазину:

- **Зручний для користувача.** Він має зрозумілу структуру і категоризацію, працює швидко і безпомилково.
- **Багатоформатовий.** Сайт однаково відображається та працює в різних браузерах, на різних операційних системах, на смартфонах і планшетах.
- **Наповнений якісним контентом.** Кожен товар повинен супроводжуватися докладним описом із зазначенням ціни, якісними і достовірними зображеннями. Умови оплати, доставки, повернення - вся комерційна інформація повинна бути доступна для користувача без ускладнень.

Технічне обслуговування Інтернет-магазину вимагає цілої команди фахівців різних спеціальностей: контент-менеджерів, програмістів, дизайнерів. Для успішного розвитку бізнесу важливим є застосування Інтернет реклама та маркетингу, відпрацьовані схеми оплати, доставки та гарантії у разі неякісного товару, call-центр з різних питань, юридична підтримка.

## 8.2. Розробка дизайну сторінок

### Графічний дизайн сайту

Продуманий дизайн сайту є важливим чинником для успіху діяльності компанії. Користувач, заходячи на сайт, в першу чергу звертає увагу на дизайн сайту. Грамотний, кваліфікований і зручний дизайн вирішує багато, він виконує роль візитної картки, що вигідно представляє компанію, професійно і повно відображає діяльність, пропоновані послуги або товари компанії і забезпечує зручність користування ресурсом.

На цьому етапі працює дизайнер (художній або технічний), який обізнаний в веб-технологіях, оскільки створений макет сайту має враховувати технічні можливості переведення картинки у html та програмний код. Необхідно подбати про нормальну швидкість завантаження, інакше відвідувач покине ресурс. Тому, не слід захоплюватися складними художніми складовими дизайну і прагнути використовувати оздоблювальні елементи, які можна реалізувати програмним шляхом.

### Генерація ідей дизайну

Як і мода, веб-дизайн слідує за модними трендами, тому, дизайнер має бути обізнаним щодо останніх тенденцій в веб-дизайні. Як і при знайомстві з новою людиною, тут важливо справити хороше перше враження.

**Простота і акуратність.** Реклама, банери, іконки, стрічки, різні знаки, спливаючі вікна, кнопки і багато іншого - часом цього стає занадто багато. Використовуючи елегантний дизайн і вільний простір, можна значно вплинути на враження відвідувачів.

**Дотримання нових трендів в веб-дизайні.** Графічні та колірні рішення, цікаві деталі, оригінальне меню – все це можна обіграти в конкретному дизайні і красиво поєднати.

**Легке сприйняття тексту.** Текстова інформація надає необхідну інформацію і відповідає на питання ще до того, як вони були задані. Тому, важливо, щоб користувачам було легко читати тексти на сайті. Існує кілька простих правил, яких варто слідувати:

- **Достатня контрастність.** Наприклад, шрифт світло-сірого кольору на білому тлі може позбавити бажання читати або дратувати. При застосуванні колірної схеми варто протестувати наскільки легко читається текст.
- **Достатній розмір шрифтів.** Дрібний шрифт є абсолютно не практичним. Користувачі не повинні напружувати свій зір, щоб прочитати текст.
- **Обмежена кількість шрифтів.** Використовувати максимум 2-3 шрифти. Художні чи рукописні шрифти використовувати лише для заголовків чи особливих фрагментів.

## Вигляд мобільної версії

Важливим є адаптація задумів до втілення на дизайні для мобільних пристроїв. Це стосується розміщення елементів, вигляд слайдерів, таблиць, великих зображень. Якісно виконаний дизайн для телефону заохочує клієнтів частіше користуватися сайтом у вільні хвилини під час робочого дня чи за потребою.

## Узгодження ідеї

На підставі наведених правил складають схематичний ескіз проекту або на папері або в графічному редакторі (рис.8.1). Це виконується для того, щоб узгодити розташування основних інформаційних блоків, графіки та інших елементів дизайну. Змовник узгоджує основний напрямок дизайну з певними зауваженнями по деталях.

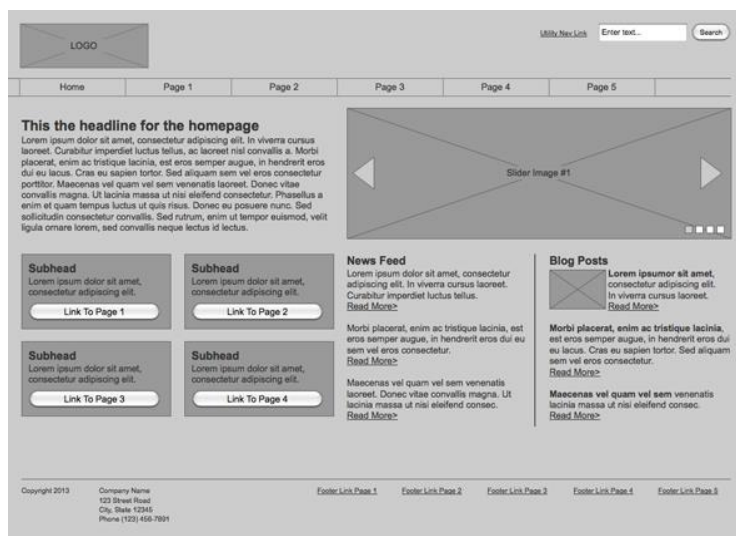


Рис.8.1. Схематичний ескіз сторінки

## Розроблення макету головної сторінки сайту

Для створення головної сторінки, що здатна привернути увагу користувачів з першого погляду варто дотримуватися певних правил.

**Структура.** Структура головної сторінки повинна бути простою і не містити зайвих елементів. Не слід зловживати занадто великою кількістю інформації і картинок. Контент сайту подається в акуратному і добре продуманому вигляді. В верхній частині сторінки розміщується сама важлива інформація: логотип, назва сайту та його напрямок або вид послуг, що надаються.

**Зображення.** Доречно застосовані зображення можуть краще представити проект ніж багато рядків тексту. Слід використовувати зображення високої якості, які викличуть у користувачів бажання продовжити вивчення сайту.

**Кольори і фон.** Для підкреслення бренду компанії або тематики сайту слід використовувати колірну схему і фон, які добре працюють разом. Варто уникати використання надмірної кількості різних кольорів і не використовувати складний фон, що відвертає увагу з основного тексту та зображень.

**Кнопки.** Кнопки із закликом до дії ведуть на інші сторінки, сайти, промо-акції, каталоги і т.д. Їх слід зробити привабливими, щоб переконати користувачів натиснути на кнопку і зробити перехід. Текст на кнопці повинен бути коротким і чітким – одне-два слова.

**Текст.** Складно створити ідеальну головну сторінку, не використовуючи текст. Текст головної сторінки має коротко пояснити суть сайту. Не варто надавати великі секції з текстом, на головній сторінці обмежуються 1-2 абзацами, більш розлога інформація подається на відповідних сторінках сайту.

## Основні елементи сторінки

Зазвичай, основними елементами сторінки є основний блок сторінки (wrapper, container), логотип, навігація, контент, хедер (верхній колонтитул), футер (нижній колонтитул), вільний простір.

- **Основний блок (контейнер).** Роль контейнера на сторінці може виконувати безпосередньо елемент body або div. Ширина основного блоку може бути еластичною (fluid), а може бути фіксованою (fixed).
- **Логотип.** Текстовий або графічний логотип найчастіше розташовується у верхньому лівому кутку сторінки або ж посередині (в залежності від ідеї, макета).
- **Навігація.** Основна навігаційна панель містить посилання на основні розділи сайту. Навігаційна панель часто розташовується у верхній частині сторінки (незалежно від того вертикально або горизонтально розташовуються елементи навігації).
- **Контент.** Основна складова веб-сторінки. Він займає чільну роль в дизайні сторінки, тому займає більший простір, підкріплений, окрім тексту, графікою.
- **Нижній колонтитул (footer).** Даний елемент розташовується внизу сторінки і, зазвичай, містить інформацію про власників, контактні дані, посилання на основні розділи сайту (може дублювати основну навігацію), посилання на соціальні мережі, форму зворотного зв'язку та інше.

## Дизайн під мобільні пристрої

Швидкий розвиток мобільних пристроїв потребує сучасного підходу до розробки веб-сайтів. Інтернет-користувачі переглядають сайти на різних пристроях з екранами різних розмірів. Розміри екранів постійно змінюються, тому, важливо, щоб сайт адаптувався до будь-якого з них.

Першим підходом до вирішення цієї проблеми було створення окремої мобільної версії сайту, яка розміщлася на окремому домені виключно для мобільних користувачів. На сьогодні розробники відмовляються від окремих мобільних версій веб-сайтів і створюють єдину версію, яка працює і адаптується безпосередньо під всі пристрої: стаціонарні комп'ютери, ноутбуки, планшети або смартфони.

Існує два основні підходи створення сайтів, що легко адаптуються для різних типів пристроїв:

- **Responsive Design** (Responsive Web Design, RWD) - чуйний дизайн - проектування сайту з певними значеннями властивостей, наприклад, гнучка сітка макета, які дозволяють одному макету працювати на різних пристроях;
- **Adaptive Design** (Adaptive Web Design, AWD) - адаптивний дизайн - проектування сайту, що базується на кількох макетах фіксованої ширини.

Чуйний і адаптивний веб-дизайн тісно пов'язані і вирішують одну задачу, але в різний спосіб. Основна відмінність між цими прийомами - чуйний дизайн - один макет для всіх пристроїв, адаптивний дизайн – різні макети для різних пристроїв. Їх поєднання забезпечує ідеальну формулу для функціональних сайтів.

#### *Чуйний дизайн*

Чуйний дизайн об'єднує три методики - гнучкий макет на основі сітки (fluid grid), гнучкі зображення (fluid images) і медіазапити. Пропорції та розміри елементів задаються у відсотках. При зменшенні ширини сторінки весь вміст плавно стискається, структурні елементи зменшуються відносно один одного (рис.8.2). Так, наприклад, якщо веб-сайт мав 3-х колонкову структуру, то на вузькому екрані він буде мати дві або одну колонки контенту.

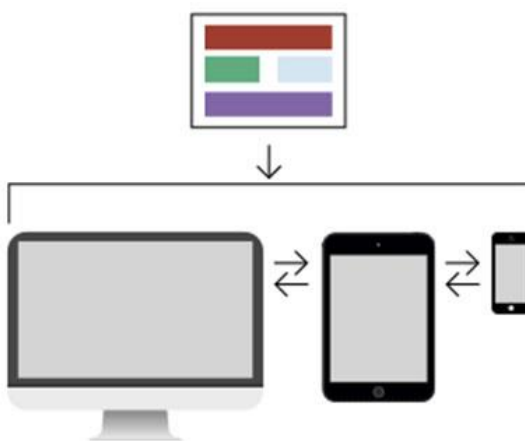


Рис. 8.2. Макет для чуйного дизайну

Для створення чуйної версії веб-сайту використовуються медіа запити (media queries) - блоки коду із вказуванням параметрів екрану. Медіа-запити дозволяють застосовувати певні правила (стилі) для виведення контент-блоків у різний порядок та пропорції залежно від ширини екрана та можливостей пристрою, на якому відбувається перегляд сайту.

#### *Адаптивний дизайн*

На відміну від чуйного дизайну, адаптивний дизайн орієнтується на розміри пристроїв. Він використовує кілька статичних макетів для різних типів пристроїв, базуючись на контрольних



(переломних) точках (рис.8.3). Макети завантажуються при певних розмірах вікна браузера пристрою, а переходи між макетами відбуваються стрибкоподібно, а не плавно.

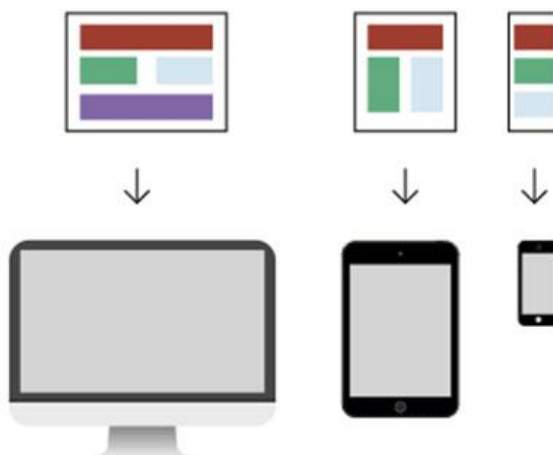


Рис. 8.3. Макети для адаптивного дизайну

В адаптивному дизайні серверні скрипти спочатку визначають тип пристрою, за допомогою якого користувач намагається отримати доступ до сайту (настільний ПК, телефон або планшет), потім завантажує саме ту версію сторінки, яка найбільш оптимізована для нього. Для елементів сітки задаються фіксовані розміри.

При розробці макету мобільної версії сайту намагаються на перший план винести основний контент, тому, часто навігаційне меню перетворюється в «гамбургер-меню», великі банери та декоративні елементи приховуються, блоки контенту, зазвичай, розташовують один під одним. На заздалегідь складеному макеті можна визначити, які елементи залишаються видимими на мобільному, а які будуть приховані.

### **Модульна сітка**

Модульна сітка представляє набір невидимих напрямних, вздовж яких розташовуються елементи веб-сторінки. Це полегшує розміщення даних в макеті, забезпечує візуальний зв'язок між окремими блоками і зберігає спадкоємність дизайну при переході від однієї сторінки до іншої (рис.8.4).

#### *Переваги застосування модульної сітки*

- Сітка задає стандарт розташування елементів: це полегшує вирівнювання елементів, додавання нових блоків і підтримку сторінки в подальшому. Напрявні інтуїтивно підказують, де краще розташувати елементи. Зменшується ймовірність помилок при перенесенні елементів з однієї сторінки на іншу. На основі сітки простіше робити адаптивний дизайн.
- Сітка дозволяє створювати макети та шаблони швидше, спрощує роботу для дизайнера і верстальника. Допомагає швидше розібратися в макеті новим учасникам, оскільки в сітці завжди є логіка.
- Сітка створює візуальний порядок і допомагає користувачеві швидше орієнтуватися та зчитувати інформацію. Допомагає сторінці виглядати більш естетично за рахунок того, що елементи є пропорційними і структурованими.

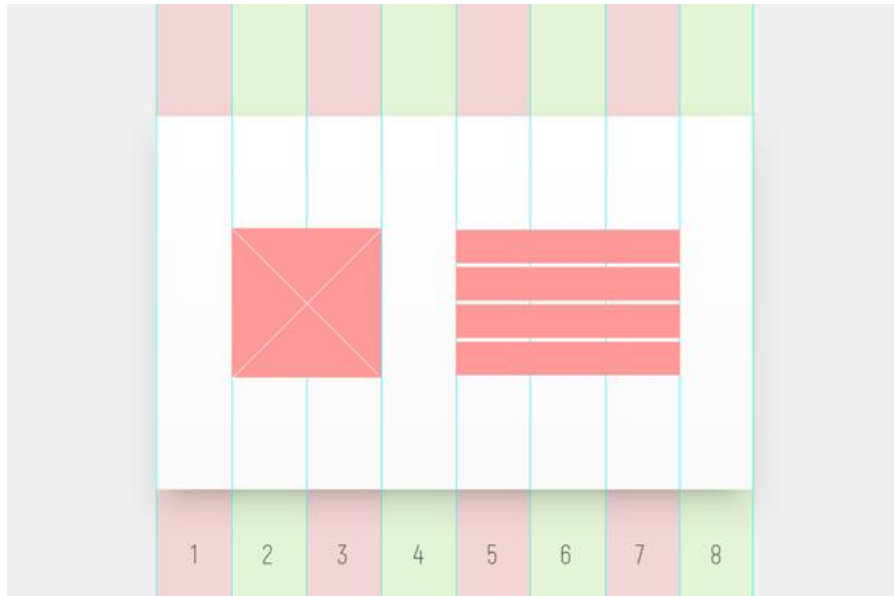


Рис. 8.4. Модульна сітка макету

Завдяки модульній сітці блоки контенту і елементи будуть розташовуватися на певній відстані один від одного, мати прийнятну ширину, що в подальшому візуально буде приємно користувачеві і не викликати в нього незручностей у сприйнятті сайту.

### **Навігаційне меню**

Навігація є важливим елементом дизайну, який допомагає користувачеві швидко отримати доступ до потрібних частин сайту. Планування навігації заслуговує на особливу увагу, оскільки сайт стає логічним і впорядкованим, підвищується його юзабіліті і відвідувачі швидко орієнтуються в географії сторінок.

Навігаційні меню є кількох типів, які можуть комбінуватися в залежності від розгалуженості структури сайту.

### **Горизонтальне меню**

Горизонтальне меню підходить для сайтів з невеликою кількістю розділів, а також для односторінкових сайтів. Зазвичай, розташовується під або над шапкою сторінки.

Опції меню виконують у вигляді тексту, іконок, вкладок. Назви опції мають бути в 1-2 слова. Хорошим тоном є виділення поточної опції кольором, підкресленням або в інший спосіб. Якщо сайт достатньо довгий, горизонтальне меню можна зафіксувати, щоб воно не прогорталося.

Слабке місце горизонтальних меню полягає в обмеженні кількості опцій. Тому, сайти зі складною структурою використовують випадні меню, які допомагають приховати підрозділи і не захаращувати простір сайту. Випадні меню підходять для сайтів з розгалуженою структурою: магазинів з широким асортиментом і великою кількістю груп товарів. Часто на потужних ресурсах одночасно використовують горизонтальне і вертикальне меню.

## Вертикальне меню

Вертикальне меню може містити значно більшу кількість опцій, що складаються з більшої кількості слів. Таке меню є зручним для магазинів і каталогів, де кількість товарів може збільшуватися. Широко вживаються групування опцій, розгортання підменю, іконки.

## Меню-гамбургер

Популярний тип меню, що на сьогодні використовують не лише на мобільних, а й на десктопних версіях сайтів. У ньому може ховатися кілька пунктів або повний набір розділів і підрозділів. Недолік такого типу меню - непомітність, тому варто подбати про те, щоб користувачеві було зрозуміло, що перед ним саме меню.

## Рекомендації при розробці навігаційного меню:

- Логотип компанії завжди повинен вести на головну сторінку сайту. Згідно досліджень 36% людей схильні використовувати логотип як засіб переходу до головної сторінки.
- Горизонтальне меню має містити якомога менше його опцій (від 3 до 7). Більша кількість заплутує відвідувачів. Назви опцій повинні бути лаконічними, але разом з тим і ємними.
- Меню повинні бути оптимальними за розміром для екранів різних пристроїв: текст буде простіше читати, а на кнопки - натискати.
- Ефективним є виділення опцій кольором або hover-ефектом, щоб мати можливість вказати відвідувачам, в якій області сайту вони знаходяться в даний момент.
- Фіксовані меню, що залишаються на екрані при скролінгу, значно допомагають відвідувачам. Це відноситься до будь-якого стилю навігації: мобільного, десктопного, горизонтального і вертикального.
- Для користувачів мобільних і десктопних пристроїв пропонуються різні варіанти меню, що пов'язано з розмірами екрану. Меню для мобільної версії сайту варто робити більш виразним і краще оточити рамкою, щоб надати вигляд кнопки.
- Застосовані іконки мають бути легко впізнаваними за змістом.
- Глибокі, багаторівневі меню часто приводять до розгублення відвідувачів. Як альтернативу можна використовувати мега-меню і навігаційні ланцюжки (breadcrumbs), щоб полегшити подальшу навігацію.
- Креативний варіант меню краще розміщати в очікуваному для відвідувачів місці. Багато дизайнерів експериментують з рорир-формами, оскільки несподіваний рух на екрані має добре захоплювати увагу відвідувачів. Але спливаючі вікна не повинні бути зведені до вікон, що оповіщають про маркетингові пропозиції та інше, вони повинні використовуватися суто для навігації.

## Типографіка

Значна частка контенту, який потрібен користувачу - це текст: опис товарів, способи замовлення послуги, порівняння марок тощо. Багато користувачів не просто шукають, а

порівнюють, читають аналітичні матеріали та статті на сайтах, тому, для них потрібно забезпечити максимальний комфорт і читабельність.

Нечитабельний текст на сайті розчаровує користувача, який скоріше проігнорує і текст і сайт. Дрібний шрифт, надвеликі текстові блоки та відсутність підзаголовків відлякують значний сегмент цільової аудиторії.

### **Основні параметри розміщення тексту та типові помилки:**

**Шрифт.** Існує два типи шрифтів: із засічками (serif) і без засічок (sans serif).

## SERIF                      SANS SERIF

Для перегляду на екрані використовується шрифт без засічок, шрифти класу serif підходять краще для друкованої продукції. Це пояснюється тим, що з екрану читати складніше, а невелика роздільна здатність дисплеїв не надає високої якості для дрібних засічок. Стандартними і популярними для основного тексту є шрифти Arial, Verdana, Helvetica. Розміщення на сайті тексту в стандартному шрифті Times New Roman виглядає убого і нагадує саморобні безкоштовні сайти.

**Колір шрифту.** Кольори фону і основного тексту повинні бути контрастними. Чим контрастніше текст, тим він краще сприймається. Для з'ясування достатньої контрастності кольорів, потрібно створити дублікат макету сторінки і перетворити зображення у відтінки сірого, якщо текст є читабельним, тоді контрастність обраних кольорів є достатньою.

На сайті не слід використовувати більше чотирьох різних кольорів тексту. В ідеалі на сайті повинен бути темний текст на світлому фоні, або навпаки. Не варто використовувати синій колір для шрифту, оскільки він традиційно використовується для позначення посилань. Найгіршими поєднання вважають червоний чи синій на чорному тлі та світло-сірий на білому.

**Розмір шрифту.** Розмір шрифту для основного тексту повинен бути в межах від 16 до 20 пікселів, великий шрифт дратує, дрібний є не читабельним. Для заголовків таких обмежень немає, і якщо дизайн допускає – шрифт може бути доволі великого розміру

**Інтерліньяж.** Це відстань між рядками. У тексті інтерліньяж повинен бути в півтори рази більше, ніж у друкованих виданнях, це полегшує перегляд і читання тексту.

**Абзаци.** Тут немає якихось жорстких обмежень за розміром і кількістю, але завжди хочеться зробити текст не просто інформативним, але і красивим.

Відносно довжини абзацу, яка буде оптимальною для швидкості читання, існує припущення, що людське око в певний момент часу може сфокусуватися приблизно на 5-7 рядках тексту довжиною приблизно в 10 слів. Вважається, за таких умов збільшується швидкість читання і сприйняття матеріалу. Занадто великі абзаци погано сприймаються, занадто короткі виглядають «брудно», тому варто самотні рядки варто об'єднувати до одного абзацу.

Вирівнювання тексту за лівим краєм збільшує швидкість читання, оскільки рівний лівий край допомагає знайти початок нового рядка. Вирівнювання тексту за шириною може надати вкрай небажані великі відстані між словами, тому використовується рідко.

У друкованому виданні для виділення наступного абзацу використовується відступ першого рядка. Це дозволяє читачеві легко відшукувати поглядом новий абзац і підвищує, таким чином, читабельність тексту. Для текстів на веб-сторінках цей прийом не використовується, а для розмежування абзаців застосовується збільшені відстані між абзацами.

**Підзаголовки.** Тексти довше за 1,5-2 тисяч знаків потрібно ділити на підрозділи з заголовками. Зазвичай, користувач спочатку переглядає текст у пошуках потрібної інформації. Підзаголовки економлять час і дозволяють швидше знайти те, що потрібно.

**Шрифтові виділення.** В арсеналі дизайнера є три види виділення основних думок тексту: грубіший шрифт, курсив і великі літери (капслок). Виділення грубішим шрифтом використовується для позначення головного слова абзацу, його тематики, курсив традиційно позначає головну думку автора, а великі літери використовуються для резюмування і привертання уваги. Не рекомендується використовувати великі літери занадто часто - це на мові шрифту позначає крик і дратує відвідувача.

**Виділення посилань.** Доцільним вибором будуть традиційні кольори: синій для посилання, фіолетовим для відвіданого пункту. Якщо існуючий дизайн сайту не передбачає цього, тоді слід пам'ятати правило: посилання повинні бути виділені в інший ніж основний колір, текст посилання слід підкреслювати, використані посилання повинні відрізнятися за кольором від активних.

Не варто підкреслювати текст, який не є посиланням - це вводить користувача в оману. Людина заходить на сайт, бачить щось схоже на посилання (те, що повинно бути посиланням, судячи з його досвіду), натискає і не отримує нічого. У мережі занадто велика конкуренція, щоб розчарувати і образити користувача такими дрібницями.

## Порядок у вихідному макеті дизайну

Всі роботи по створенню макету сайту проводяться в графічному редакторі. Беззаперечним лідером серед редакторів, що використовують веб-дизайнери є Adobe PhotoShop.

- Правильно складений макет повинен бути збережений у форматі \*.psd із чіткою ієрархією прошарків та груп прошарків.
- Вихідний макет очищають від елементів, які не використовуються, всі прошарки йменуються, для текстових прошарків роблять растрові копії. Прошарки, що відносяться до одного елементу чи блоку групуються по папках. Зайві прошарки макету видаляються, щоб не заплутувати верстальника та зменшити розмір файлу.
- Наявність напрямних по модульній сітці макету. Такий порядок дозволяє швидко орієнтуватися в макеті, легко виділити, замінити чи змінити потрібний елемент.
- Несистемні шрифти додаються до папки з макетами, оскільки PSD-файл не зберігає їх дані. Для верстання за допомогою @font-face даний шрифт втілюється у сторінки і верстальник не витрачає час для пошуку потрібного шрифту.
- Розроблений макет також зберігають в популярному растровому форматі (.jpg або .png) для перегляду та остаточного затвердження клієнтом.

- Вихідні файли з текстовими поясненнями передаються до верстальника, який буде робити шаблони сторінок. Через різне кодування в операційних системах важливо використовувати латиницю в назвах.
- Для дизайнера варто мати базові знання щодо HTML та CSS кодування, принципи та основи верстки і тих фреймворків, що використовуються. Це дозволить створювати елементи та ефекти, які можна легко реалізувати кодом, мінімізуючи використання графічних зображень.

### *8.3. Верстання сторінки (кодинг)*

По закінченні роботи зі створення графічного макету дизайну і схвалення його іншими учасниками проекту чи замовником, приступають до створення HTML-шаблону сторінки.

Верстка - це процес перетворення дизайну сайту (картинки) у веб-сторінку. Макет дизайну це зображення в форматі PSD (Photoshop), а шаблон веб-сторінки - це інтерактивне відображення сторінки, де можна виділити текст, перейти за посиланням, заповнити форму, додати товар в корзину і так далі.

Верстання сторінок – це процес написання HTML коду сторінки, за яким сторінка набуває вигляд, подібний до дизайну макету. Безумовно, існують різні технології розмітки тексту, які підтримуються браузерами, проте, найпоширенішим варіантом є верстання сторінок за допомогою мови HTML та стилів CSS.

Верстка сторінки робиться поетапно: спочатку створюється HTML-структура (HTML-код), додаються стилі (CSS-код), а далі за потребою пишуться скрипти (JS), додаються необхідні плагіни і бібліотеки.

Для того, щоб навчитися верстанню, слід спочатку зрозуміти значення операторів HTML та CSS і приступати до вивчення верстки. Для початківців починати слід з верстання простих структур, але з часом ускладнювати завдання. Питання як зверстати це, як зверстати то - відпадуть з накопиченням досвіду.

Наприклад, почати з верхньої частини сторінки: створити контейнер, додати до нього фон, встановити логотип картинкою або текстом. Далі - створення меню по частинах: вивести пункти, вирівняти їх, поставили поля і відступи, зробити hover-ефект. І так поступово, слід верстати всю сторінку.

*Поради для початківців:*

1. Вивчати HTML та CSS по свіжих матеріалах і від фахівців, що мають авторитет серед спільноти розробників.
2. Намагатися верстати, відтворювати сторінки відеоуроків, вивчати код готових шаблонів, яких є багато у відкритому доступі.
3. Накопичений досвід сприятиме доведення навичок верстальника до автоматизму, і в подальшому верстання сторінок буде відбуватися і швидше і якісніше.

## Редактори коду

З найбільш популярних редакторів коду на сьогодні можна виділити три, які схожі за принципом роботи, до них додатково можна під'єднати необхідні модулі та плагіни:

- [Sublime Text](http://www.sublimetext.com/3) (http://www.sublimetext.com/3)
- [Atom](https://atom.io/) (https://atom.io/)
- [Brackets](http://brackets.io/) (http://brackets.io/)

Існують більш просунуті комплекси IDE (Integrated Development Environment - Інтегроване середовище розробки) такі, як Adobe DreaWeaver, Web Storm, PHP Storm тощо, але для верстки нескладного проекту цілком підійде редактор коду, а не цілий комплекс.

## Блокова верстка

Сучасний підхід до HTML верстки передбачає розподілення елементів сегментів у різних місцях, і це називається блоковою версткою. Блокова верстка базується на принципах розташування і взаємодії за допомогою блокових елементів.

В HTML4 як базовий елемент розмітки використовувався тег `div`, що не має певного семантичного значення та особливих вимог на його вміст. Тег `<div></div>` є контейнером для тексту, зображень та інших елементів сторінки і вони позиціонуються в різних місцях. Теги `div` можна помістити в будь-яке місце веб-сторінки, позиціонувати абсолютно (вказати координати `x` і `y`) або відносно (вказати відстань до інших елементів сторінки). На рис.6.16 показано просту структуру сторінки в кодуванні HTML4.

В HTML5 з'явилися нові елементи, які служать заміною для звичайних блочних елементів з атрибутами `class` і `id`, наприклад `<div class="article">` або `<div id="page">`. На тепер можна використати семантично змістовні блочні теги, що за призначенням аналогічні до тегу `div`. Це теги `section`, `article`, `aside`, `header`, `footer` та інші.

Для спрощення роботи можна скористатися генераторами структури сторінки, які створюють код загальної структури сторінки з CSS-файлом, в якому прописані загальні стилі. Як приклад таких генераторів є шаблонізатор Dreamweaver, онлайн генератори [CSStemplater.com](http://CSStemplater.com) і [CSScreator.com](http://CSScreator.com).

*Особливості сторінки з блочною версткою:*

- Легкий і зрозумілий код сторінки. Код сторінки компактний, коментований, не захарашений зайвими стилями.
- Блокова верстка дуже «дружно» відноситься до позиціонування. Блоки можуть перекривати один одного, розробник вказує, який з блоків буде зверху.
- Блокова верстка дозволяє розмістити блок, що відображає будь-яку частину сторінки на початку html-коду. Це буде доречним при застосуванні певних технологій просування сайтів.
- Максимально ідентично відображає сторінку в різних браузерях при різних розмірах екрану.

## Перевірка HTML і CSS коду

Перевіряти код сторінки сайту необхідно, коли здійснюються кардинальні зміни в структурі сайту. Акуратний верстальник прагне це робити якнайчастіше. Перевірку HTML і CSS коду краще робити з використанням програм-валідаторів чи Інтернет-сервісів, які відображають знайдені помилки відповідно до різних стандартів. Кращим валідатором за допомогою якого можна перевірити любу сторінку, що розташована в Інтернеті або на локальному комп'ютері є валідатор міжнародного Консорціуму W3C (WorldWideWebConsortium) <http://validator.w3.org/>

Сучасні браузерери підтримують стандарти W3C значно краще, ніж їх попередні версії. Якщо правильний код відповідає певним формальним правилам, його легше інтерпретувати й обробляти, він швидше аналізується і відображається в браузері, з ним легше працювати пошуковим системам.

Відповідність коду сторінки до стандартів W3C не є гарантією якості та ефективності. Грамотний код - це важливий, але далеко не єдиний показник для оцінювання сторінки.

## Поширені вимоги до HTML та CSS-коду

### 1. Кросбраузерність

Сайт повинен нормально працювати як в останніх так і старших версіях популярних браузерів, зокрема Internet Explorer, Mozilla FireFox, Opera, Chrome, Edge.

### 2. Застосування коментарів

Основні HTML блоки коментарями коментуються в такий спосіб:

```
<!-- BEGIN FOOTER -->
<!-- END FOOTER -->
```

CSS блоки в такий:

```
/* FOOTER */
```

Якщо використовуються CSS префікси чи хаки, також потрібні коментарі, що це і для якого браузера. Коментарі допомагають орієнтуватися в коді не лише верстальнику, але і решта учасників проекту: програмісту, контент-менеджеру, оптимізатору.

### 3. Впорядкування в таблиці стилів

CSS файл повинен бути розділений за допомогою коментарів на блоки за функціональним або структурним призначенням, наприклад:

```
/* 1. Скидання CSS */
/* 2. Типові елементи */
/* 2.1. Заголовки */
/* 2.2. Посилання */
/* 2.3. Елементи форм */
/* 3. HEADER (Шапка сайту) */
/* 4. FOOTER (підвал) */
/* 5. SIDEBAR (Праворуч) */
```

### 4. Змістовні назви

Назви класів і id повинні за змістом відповідати застосуванню, наприклад, *header*, *menu*, *footer*, *news*.



## 5. Вживання Javascript

- Все що можна зробити без скрипту, робити без нього. Якщо Javascript коду багато - потрібно його виносити в окремий файл. Обробники подій теж краще відокремити і оголошувати в окремому файлі.
- Заздалегідь узгодити JavaScript-бібліотеки, наприклад, jQuery чи PrototypeJS.
- Якщо в макеті присутній JavaScript, що змінює DOM – потрібно прослідкувати його поведінку в різних браузерах.

## 6. Ширина сторінки

Для еластичних макетів обов'язково повинна бути задана мінімальна і максимальна ширина. Для фіксованих дизайнів ширина має бути 960-980 пікселів. В іншому випадку може з'явитися горизонтальний скролінг, що свідчить про низький професійний рівень розробника.

## 7. Особливості відображення

Обов'язково вказувати колір для фону сторінки (body), навіть якщо він білий. Певна частка відвідувачів встановлює за замовченням фоновий колір браузера, відмінний від білого, що призводить до відображення сторінки не в тому вигляді як задумав дизайнер.

## 8. Порядок в файлах

Файлова структура повинна бути стрункою і не містити файлів, які не використовуються в сайті (HTML-файлів, зображення, стилі, скрипти).

## 9. Розташування футера у нижньому боці браузера

В макетах, де висота сторінки залежить від контенту (а таких, як правило, більшість), футер має триматися низу браузера при відсутності або малій кількості контенту.

## 10. Кодування тексту

На сьогоднішній день для тексту стандартним є кодування UTF-8. Це кодування застосовують для HTML, CSS та JS коду, інакше ймовірність довгого виправлення багів критично зростає.

## 11. Використання нестандартних шрифтів

Якщо в макеті використовуються нестандартні шрифти, слід з'ясувати за якою технологією їх буде втілено: @ font-face, Google Font API чи інше.

## 12. Використання невалідних фрагментів

Якщо макет не проходить 100% HTML-валідацію, варто вживати лише виправданий невалідний код.

## 13. Звітність про виправлення

Якщо задача верстки проводиться більш ніж одним етапом (наприклад, верстальник відправляє сторінки по одній, або якщо йому повертаються на доопрацювання вже зверстані сторінки), а система контролю версій для верстки не використовується, тоді виконавець повинен в обов'язковому порядку прикріпити файл з описом змін в макеті приблизно такого змісту:

- Додано нові картинки в папку img,

- Картинки btnHome.jpg і btnFeedback.jpg вже не потрібні, можна видаляти.
- Змінено html-код в секції файлу anketa.html.
- Додано в кінець файлу main.css нові стилі (починаючи з .form\_row і нижче).

## 8.4. Програмна частина проекту

### Інтеграція сайту з системою управління

На ранніх етапах розвитку Інтернету, розробка сайту зводилася до створення файлової структури з HTML-сторінок та ручного розміщення в них тексту, зображень, елементів навігації та посилань. Тоді, це не було настільки трудомістким заняттям, щоб дбати про його автоматизацію, оскільки сайти здебільшого представляли невеликі проекти, які робилися для автора чи його колег.

З часом обсяги інформації почали експоненціальне зростати, збільшилося число відвідувачів сайтів, збільшилися трудовитрати на підтримку сайту в актуальному стані. Власник сайту був змушений більшу частину часу витратити не на безпосереднє розміщення статті або публікації, а на внесення супутньої інформації, на зразок посилань на цю статтю, зміни меню навігації або виправлення відомостей.

Виходом з даної ситуації стало створення спеціального класу програм – систем управління контентом CMS (Content Management System), які спроможні виконували рутинні операції, не пов'язані з безпосереднім створенням статей.

В принципі і зараз існують повністю статичні сайти. Вони цілком виправдовує себе, оскільки містять максимум десятків сторінок, причому заздалегідь відомо, що сторінки не будуть змінюватися і сайт виконує лише репрезентативні функції. Переводити такий сайт на движок не доцільно.

В загальному, CMS - це програмна оболонка, яка дозволяє легко вводити і редагувати дані: текст, зображення, додавати і видаляти сторінки, тобто, керувати сайтом в режимі онлайн, без ґрунтовних знань HTML, мов програмування та інших спеціальних навичок. CMS містить комплекс найбільш поширених модулів - наприклад, систему коментарів, голосування, фотогалерею тощо, які не доведеться шукати і додавати до сайту окремо.

Оскільки створити сайт за допомогою CMS можна швидко і без глибоких спеціальних навичок, цей інструмент стає все більш популярним.

## 8.5. Інформаційне наповнення сайту

**Контент (Content, вміст)** — це інформаційне наповнення сайту, інформація, яку розробник складає самостійно або копіює з дотриманням відповідних законодавств. Весь контент охороняється законом про авторське право, оскільки він є продуктом інтелектуальної праці і має своїх авторів і власників.

Якісний контент - це матеріал, що поєднує ряд властивостей: унікальність, корисність, актуальність, оздоблення, інформативність. Якісний контент дозволяє сайту краще ранжуватися в пошукових системах і привертає на сайт більше відвідувачів.

Користувач, заходячи в Інтернет, має на меті певні потреби і задовольнити їх є першочерговим завданням будь-якого контенту.

- **Потреба в інформації.** Користувачеві потрібна інформація, для нього Інтернет - це величезна бібліотека, де відповіді можна знайти на будь-яке питання.
- **Потреба в розвагах.** Людина хоче відпочити і подивитися відео-ролики, пограти в онлайн-ігри, розглядати фотографії.
- **Потреба в комунікаціях.** Користувачеві хочеться з кимось поговорити, зв'язатися.

Алгоритми роботи пошукових систем стрімко розвиваються і тепер потрібно наповнювати сайт хорошим контентом. Пошукові системи враховують унікальність контенту через середній час, які користувачі проводять на сайті, читаючи статті. Враховується дата додавання матеріалу та дата останнього оновлення. Чим вона свіже - тим вище актуальність вмісту, тим швидше сайт потрапить в топ видачі.

Хороші матеріали швидше зацікавлять користувачів. Якщо написано цікаву статтю, нею стануть ділитися, збільшиться потік відвідувачів, про сайт дізнаються, складеться хороша репутація. Все це допоможе досягти основної мети – сайт стане популярним.

## Типи контенту

**Текст. Це статті, новини, замітки.**

- **Копірайтинг.** Авторський унікальний матеріал, що ґрунтується на особистому досвіді автора, його знаннях.
- **Рерайтинг.** Дані беруться з різних джерел, обробляються, переписуються. Текст виходить унікальним, але дані, що використані для написання, запозичено ззовні.
- **Оптимізовані тексти (SEO-тексти).** Текст, що перероблений спеціально під пошукові системи і містить ключові слова. Від грамотності автора залежить, наскільки ключі будуть вдало вписані в текст, чи не будуть вони різати око.
- **Копіпаст, плагіат.** Скопійований текст з джерела без змін або з мінімальною корекцією.

## Відео-контент

Це зміст сайту, що представлений в відеоформаті. Ролики, мальоване відео (дудл-відео), огляди, інструкції - кожному виду знайдеться своє застосування. В кожного відео повинна бути мета: залучення користувачів, підвищення довіри, навчання, демонстрація продукту і так далі. І в залежності від мети вибирається вид контенту.

*За призначенням відеоконтент буває:*

- **Іміджевий.** Формує необхідне відношення споживача до продукції або компанії, породжує певні емоції. Це можуть бути кейси (успішні приклади послуг), відгуки споживачів.

- **Презентаційний.** Завданням роликів є візуальна демонстрація послуги, продукту, явища, користі або вигоди для споживача. Прикладом такого ролика є звичайна телереклама.
- **Вірусний.** Повинен привертати увагу, заражати ідеєю і спонукати ділитися цим роликом з іншими користувачами.
- **Навчальний.** Це відео вчить споживача конкретним діям, показує наслідки дій, розповідає, як поводити себе в різних ситуаціях, як товар або послуга допоможуть йому. Якщо товар складний в експлуатації, то відеоінструкції допоможуть з цим розібратися.
- **Соціальний.** В такому відео показується проблема та шляхи її вирішення. Подібна реклама повинна бути тонкою, ненав'язливою, максимально природною.

Самим популярним відеоконтентом є **відеоролики**. Вони поділяються на кілька видів, наприклад, можуть бути простими і постановочними, з закадровим озвученням чи музикою.

Різновид таких роликів - **дудл-відео**. Це анімований відеоролик, в якому певна особа малює в режимі реального часу. До нього можна додавати корпоративних персонажів, текст, озвучення. Динамічні ролики мимоволі утримують увагу глядача. А значить, він активно сприймає всю інформацію з дудл-відео.

На сайті можна розмістити **просту презентацію**, що створена зі слайдів. Якщо до неї додати оригінальні спецефекти, музику або голосове озвучення - за належної частки креативу вийде гідний ролик.

Ще один варіант відеоконтенту - **скрінкасти**. Це запис того, що відбувається на екрані, з додаванням пояснюючих коментарів. Так записуються навчальні відео.

Популярністю серед користувачів користуються і **GIF-анімації**. Вони короткі, але здатні передати більше інформації, ніж невеликий текст.

Залежно від цілей компанії та її аудиторії можна використовувати **мультиплікаційні ролики** або **ролики з 3D-анімацією**. Цей формат допомагає змодельовати найважливіші робочі процеси і наочно показати їх споживачеві.

**Відео в реальному часі** - це можливість для покупця задати питання та отримати відповіді, побачити в ролику знаменитість або подивитися демонстрацію продукту, наприклад на виставці. Деякі компанії навіть проводять навчальні вебінари.

## Інфографіка

Інфографіка - це графічне представлення складної інформації. Вона необхідна, коли складні дані потрібно доступно викласти для широкої аудиторії. В інфографіці використовуються різні засоби візуалізації: графіки, зображення, діаграми, таблиці, карти, схеми.

### Типи інфографіки

- **Аналітична** використовується на сайтах для бізнесу і економіки. В такий інфографіці багато статистики та цифр.

- **Новинна** використовується в газетах. Завдяки неї можна коротко описати події, порівняти «до» і «після», відобразити хронологію і провести аналітику, що сталося.
- **Реконструкція** відображає хронологію подій і відтворює певний процес.
- **Презентація** є ідеальною інфографікою для бізнесу. Відмінно підійде як для реклами, так і для звітів.

Інфографіка може бути представлена в двох формах.

- В єдиній є лише один блок, де всі елементи пов'язані між собою.
- В роздільній може бути кілька блоків, які не залежать один від одного.

Категоріально інфографіки бувають:

- **Числові**, які зазвичай відображають графіки, схеми, діаграми тощо.
- **Розширені** відносяться до візуальних і часто доповнені коротким описом.
- **Покрокові** - це інструкція графічного формату.
- **Інтерактивні** використовують тільки на сайтах, оскільки на папері реалізувати задумане не вийде. Така інфографіка може змінювати показники в залежності від обраних раніше значень.

### Графічні зображення

Правильно підібрані графічні файли здатні поживити текст, дозволяють проілюструвати слова, продемонструвати наочні приклади і привертають увагу аудиторії. Зображення мають бути належної якості і пасувати до змісту тексту.

Щоб збільшити інформативність тексту для пошукових систем варто слідувати нескладним правилам попередньої обробки.

- **Правильна назва графічного файлу.** Назва файлу має відповідати його вмісту. Так пошуковим роботам буде простіше зрозуміти, що зображено на картинці.
- **Розмір файлу.** Велика вага зображення значно уповільнює завантаження сторінки. Перед публікацією слід попередньо обробити зображення спеціальними інструментами для стиснення. Вони дозволяють зменшити розмір та зберегти при цьому якість і чіткість картинки.
- **Підписи до зображення.** Користувачі звертають увагу на текст, який супроводжує зображення, навіть коли побіжно переглядають статтю. Це така ж важлива частина структурування контенту, як заголовки, підзаголовки і списки. Звісно, що є зображення, які таких підписів не потребують
- **Атрибути Alt і Title.** Не слід нехтувати атрибутами Alt і Title, що описують зображення. Це важлива інформація для пошукових систем та пристроїв читання з екрану. Опис може бути розлогим і стосуватися саме цього графічного файлів.

- **Розміщення зображення.** Варто, щоб зображення оточував пов'язаний текстовий контент.

### **Принципи якісного контенту**

**Унікальність.** Хороший контент - унікальний контент. Пошукові машини цінують авторські тексти і зображення, які не збігаються з матеріалами інших сайтів. Сторінки з унікальними статтями частіше індексуються пошуковими системами, піднімаються в топі видачі, залучають більше відвідувачів. Кращий спосіб створити унікальний контент - писати його самостійно. Це складний і довгий шлях, але витрати себе виправдовують.

**Інформативність.** Матеріал повинен бути корисним для відвідувачів.

**Достовірність.** Для цікавого матеріалу потрібні хороші джерела інформації. Публікуються лише достовірні факти, це викличе довіру читачів.

**Актуальність.** Контент на сайті повинен бути актуальним. Актуальність - це не просто огляд подій, які відбулися недавно. Актуальність - це те, що саме зараз читач може взяти і втілити. Посилатися можна і на факти, які мали місце кілька років тому, але все ще не втратили своєї значущості.

**Грамотність.** Правильна мова, письмова або усна, завжди приємна. Зараз складно знайти статті в блогах, які були б написані з помилками. Тексти мають обов'язково перевірятися на наявність орфографічних і граматичних помилок.

**Структурування та ієрархія тексту.** Заголовки, підзаголовки, списки. З такими текстами зручніше працювати читачеві. І пошукові боти віддають перевагу саме тому контенту, в якому є чітка структура.

**Наявність доцільних посилань.** Допомогають читачам орієнтуватися в матеріалах, надають можливість швидкого доступу до важливих розділів.

### **8.6. Публікація сайту в Інтернеті**

Після створення сайту, його викладають в мережу Інтернет. Процедура розміщення сайту в Інтернет (публікація) не проста тема, головним чином, тому що є багато різних способів зробити це.

#### **Отримання хостингу та доменного імені**

Щоб мати більший контроль над контентом та зовнішнім виглядом веб-сайту, більшість людей вважають за краще купувати веб-хостинг та доменну адресу:

Хостинг - орендований файловий простір на веб-сервері хостингової компанії. Файли веб-сайту розміщуються в цьому просторі, а веб-сервер видає контент для користувачів, які запитують його.

Доменна адреса — унікальна адреса, за якою люди можуть знайти веб-сайт, наприклад, <https://lpnu.ua>. Купити доменну адресу можна у реєстратора доменів. Багато професійних веб-сайтів розташовується в Інтернеті саме таким чином.

Щоб знайти надійний хостинг та реєстратора доменних імен, варто в пошуковій системі зазначити «веб-хостинг» чи «доменні імена», також можна додати слово «відгуки». Часто ці компанії пропонують одночасно і хостинг та реєстрацію доменного імені.

Ваш Інтернет-провайдер може надавати послуги хостингу для невеликих веб-сайтів. Набір можливостей у такому разі може бути обмежений, але, тим не менш, він може добре підійти для перших експериментів, варто переконатися в наявності таких пропозицій.

Для публікації самостійної роботи добре підійдуть безкоштовні сервіси, такі як 000webhost.com, blogger.com, Wordpress.com. Безкоштовні послуги здебільшого не вимагають FTP-клієнтів – файли можна розміщати безпосередньо через веб-інтерфейсі.

Для розміщення файлів на віддаленому сервері потрібно мати програму, в якій реалізовано FTP-клієнт. Існує багато FTP-клієнтів, але, як правило, потрібно увійти на веб-сервер, використовуючи дані, надані хостинговою компанією (наприклад: ім'я користувача (логін), пароль, ім'я хоста). FTP-клієнт відобразить файли на локальному комп'ютері в одній половині вікна і файли на хостинговому сервері в іншій половині, так можна наочно копіювати файли з локального комп'ютера на сервер і назад (рис.8.5).

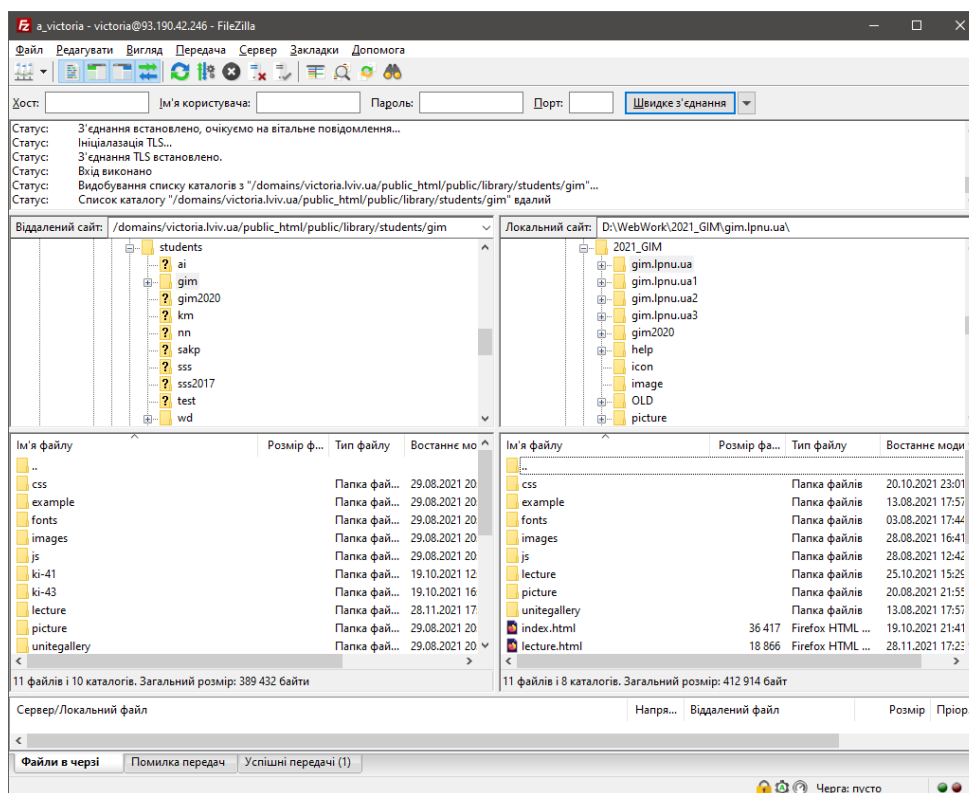


Рис.8.5. Інтерфейс FTP-клієнтів FileZilla

## 8.7. Тестування і доопрацювання

Коли сайт розміщено в Інтернеті, то перш, ніж його просувати і рекламувати, слід перевірити сайт на працездатність. Можливо, якісь скрипти виконуються з помилками або десь пропущено важливу і цінну інформацію або потрібна картинка.

Після того, як протестовано і вивірено сайт на працездатність, слід попросити людей, що не брали участь в розробці, ще раз оглянути і протестувати сайт. Можливо, що не помічено певних недоліків, які може побачити людина, що не брала участь в розробці. Також, можна

провести дослідження за допомогою опитування: що подобається або не подобається відвідувачам на сайті, чи є зручною навігація тощо.

Фінальне тестування є вкрай важливим для репутації сайту. Що зробить відвідувач, якщо не зможе додати запис у форум, або якщо стаття, що цікавить його, буде не доступною, оскільки до неї невірно прописано шлях? Правильно, він може піти і більше не повернутися.

## **8.8. Просування, реклама**

Цей етап доведеться виконувати з певною періодичністю, щоб на сайт приходили нові відвідувачі. Кількість і постійний потік відвідувачів на сайт не залежить цілком від реклами, основну роль все ж таки грає наявність цікавої для певної групи людей інформації, заради якої вони і відвідуватимуть ресурс.

Перш, ніж зайнятися рекламою сайту, варто подумати – а чи потрібно це? Якщо потрібно, то навіщо. Що зміниться від того, що на сайті стане десятком відвідувачів більше?

Якщо це все-таки потрібно, слід продумати, яка аудиторія зацікавлена у ресурсі (вік, стать, професійна зайнятість, наприклад, програмісти або вчителі, люди похилого віку або діти). Після визначення аудиторії, потрібно з'ясувати її уподобання: які сайти відвідує, які журнали читає тощо. Там слід розставити посилання: розмістити рекламу в журналі або газеті, на потрібних сайтах. Краще за все дослідження аудиторії і рекламу сайту довірити професіоналам, які ретельно вибудовують рекламну кампанію, і в них краще вийде виконати поставлене завдання в рамках бюджету. Адже реклама теж вимагає певних знань і грошових вкладень, це не такий простий етап, як може здаватися.

З безкоштовних шляхів рекламування ресурсу: каталоги, пошукові системи, банерообмінні мережі, а також схвальні відгуки про сайт на сторінках інших людей. На жаль, безкоштовно розкрутити свій ресурс дещо важко, але все-таки можна, головне запастися терпінням, адже на безкоштовне просування ресурсу знадобиться більше часу і сил.

Природно, є і інші безкоштовні технології просування сайту, як спам або накручування лічильників відвідувань – але хотілося б застерегти від використання даних методів. Ефект від них тимчасовий, дані способи є чорними, тобто такі способи не вітаються рейтинговими і пошуковими системами і можуть спричинити виключення сайту з рейтингів та блокування в пошукових системах.

### **Оптимізація сайту**

Оптимізація сайту - це комплекс методів, який зробить сайт цікавим, зручним і корисним для користувача, зрозумілим і близьким для пошукових систем.

Оптимізація сайту - це спосіб приведення сайту у відповідність до алгоритмів пошукових системах, що допоможе пошуковим системам оцінити якість виконання та наповнення. Метою пошукової оптимізації є виведення сайту в топ - в перші десять відповідей, які розташовуються на першій сторінці видачі результатів пошуку. Для того, щоб пошукова система прореагувала на сайт відповідно, потрібно вивчити її алгоритм і визначити моменти, які особливо важливі і потім, відповідно до виявлених моментів, провести заходи, які відповідають виконанню цих умов.



Пошукова оптимізація з'явилася разом з появою та розвитком пошукових систем в середині 1990 років. В той час пошукові системи надавали великого значення тексту на сторінці, ключовим словам в мета-тегах та іншим внутрішнім чинникам, якими власники сайтів могли легко маніпулювати. Це призвело до того, що у видачі багатьох пошукових систем перші кілька сторінок зайняли сайти, які були повністю присвячені рекламі, що різко зменшувало якість роботи пошуковиків. З появою технології PageRank стало додаватися більше ваги зовнішнім факторам, що допомогло Google вийти в лідери пошуку у світовому масштабі.

Однак оптимізація розвивається разом з пошуковими системами, і в сучасних результатах пошуку можна бачити багато комерційних сайтів з штучно роздутою популярністю, особливо за комерційно привабливими запитами (транзакційні запити).

Розроблено багато прийомів, здатних маніпулювати пошуковою системою, проте, вони різняться за своєю коректністю та легальністю. Будь-яка маніпуляція певними параметрами сайту може бути розцінена пошуковиком як небажаний вплив на його результати. Такі дії прямо заборонені в ліцензіях на використання пошукових систем, тому оптимізатори і маркетологи мають просувати сайт, не порушуючи правил пошукових систем.

### **Біла оптимізація**

Біла оптимізація - це легальні професійні методи, що доповнюють один одного і надають стабільний результат. Біла оптимізація спрямована на те, щоб і відвідувачі, і пошукові машини ставили сайту високу оцінку. Біла оптимізація заздалегідь передбачає, що оптимізатор має справу з корисним сайтом, наповненим унікальним контентом.

Біла оптимізація - оптимізаторська робота над ресурсом без застосування офіційно заборонених методів просування ресурсу - без впливу на пошукові алгоритми сайтів. Це передбачає роботу над самим сайтом, а саме над внутрішньою структурою, навігацією та вмістом, і роботу з зовнішнім середовищем сайту, тобто просуванням сайту шляхом статей, прес-релізів, оголошень, реклами, реєстрації в соціальних мережах, партнерських програмах із зазначенням посилань на сайт. Втім, якщо певний метод оптимізації не є офіційно забороненим, це не означає, що його можна застосовувати.

### **Сіра оптимізація**

До сірої оптимізації відносяться ті прийоми, які не заборонені, але потенційно є некоректними. Наприклад, до сірої пошукової оптимізації можна віднести додавання великої кількості ключових слів в текст сторінки, часто на шкоду читабельності для людини, наприклад: «Масло масляне, бо в ньому є маслоподібні масляні жири».

При легальному підході в підборі ключових запитів для конкретної веб-сторінки визначають необхідну частоту ключових слів в тексті, а потім формулюють речення і фрази, що містять певну кількість слів у різних відмінках, однині і множині, при різних формах дієслів. Ці параметри можуть потім коригуватися за результатами видачі пошукових систем. При цьому завдання SEO-розробника - написати оригінальний текст таким чином, щоб подібна оптимізація була непомітною для читача. Широко застосовується також додавання ключових слів в HTML-теги title, h1, alt, атрибут meta keywords.

Інший приклад сірої оптимізації - закупівля посилань, тобто розміщення посилань на сторонніх ресурсах з метою збільшення показників просувного сайту. Як правило, для цього використовуються спеціалізовані біржі. Пошукові системи намагаються боротися з подібним методом, але його використання не може привести до бану.

Сіра оптимізація відрізняється від чорної тим, що вона офіційно не заборонена, але її використання все одно може бути розцінено як неприродне завищення популярності сайту. Деякі пошукові системи, наприклад, Google, можуть тимчасово або постійно заблокувати такий сайт. Тобто, кінцеве рішення про те, чи є методи просування законними чи ні, приймає фахівець - асесор пошукової системи, а не програма.

### **Чорна оптимізація**

Чорна оптимізація містить набір прийомів, що відрізняються явною некоректністю, багато з них заборонені, деякі є звичайним обманом системи, тобто всі методи, які суперечать правилам пошукових систем. Серед них можна виділити наступні: використання дорвеїв (сторінок і ресурсів, створених спеціально для роботів пошукових систем, що суцільно заповнені великою кількістю ключових слів), прийом під назвою клоакінг (користувачеві віддається одна сторінка, а пошуковому роботу - інша, оптимізована під певні запити), використання прихованого тексту на сторінках сайту, використання «однопиксельних посилань».

У разі, якщо чорну або сіру оптимізацію буде виявлено пошуковою системою, сайт буде оштрафований або забанений.

### **Внутрішня оптимізація сайту**

До внутрішньої оптимізації відноситься аудит сайту і приведення структури та контенту сайту до вимог: наповнення мета-тегів (Description, Keywords), чітке структурування сайту, правильне оформлення головної сторінки сайту, карта сайту, навігаційне меню.

Особлива увага приділяється роботі з ключовими словами. Вона починається з визначення семантичного ядра, формування ключових слів, розподілу їх по сторінках сайту. Оформлення контенту має виконуватися з врахуванням певних вимог, які впливають на кінцеве ранжирування сайту. Ще один важливий момент - це внутрішнє перелінкування сторінок. Внутрішня оптимізація має свої особливості, які обов'язково повинен враховувати оптимізатор.

### **Зовнішня оптимізація сайту**

Зовнішня оптимізація насамперед полягає в популяризації сайту на інших ресурсах, розміщення посилань на інших сайтах, які ведуть на просувний сайт, тобто нарощуванні посилальної маси на сайт.

Способи розміщення посилань є різні і можуть відноситися до чорної, сірої, білої оптимізації. Пошукові системи з великою обережністю ставляться до раптового зростання посилань, до посилань, що розташовані у великій кількості на одному сайті, на сайті з низьким трастом, до посилань, які ймовірно є купленими. Зовнішня оптимізація, яка проведена з порушеннями, може стати підставою для отримання бану.

## Оптимізація сайту самостійно

Самостійна оптимізація сайту стала можливою завдяки появі спеціалізованих сервісів як, наприклад, WebEffector. Повністю автоматизований процес суттєво полегшує життя власнику веб-ресурсу. Успішна оптимізація сайтів за допомогою автоматичної програми, передбачає формування добірки оптимальних ключових слів і словосполучень для пошукового просування сайту. Також автоматичний сервіс якісно і недорого здійснює закупівлю посилань на інших веб-ресурсах.

Пошукова оптимізація сайту - не є сталою комбінація. Вона орієнтована на алгоритми пошукових систем, а ці алгоритми часто змінюються для того, щоб поставити різні заборони для чорної оптимізації. Разом з оновленням алгоритмів з'являються нові підходи пошукової оптимізації, вдосконалюються старі.

## Аудит сайту і внутрішня оптимізація

1. Правильний `<title> </ title>`.
2. Мета теги `description` і `keywords`.
3. Заголовки `<h1> .. <h6>`.
4. Виділення ключових слів.
5. Наявність ключових слів в контенті.
6. Структурований контент.
7. Обсяг сторінки від 500..10000 символів.
8. Внутрішнє перелікування сайту.
9. Неважлива інформація закрита для індексації.
10. Зрозумілі для людей `url`.
11. Максимальний рівень вкладеності сторінок – 3.
12. Тільки унікальний контент.
13. Прописування атрибутів `Alt` і `Title` до зображень.
14. Відсутність сторінок-дублікатів на сайті.
15. Доступність сайту з `WWW` і без.
16. Швидкість завантаження сайту.
17. Відсутність непрацюючих посилань.
18. Мікророзмітка сайту.
19. Зовнішні посилання.
20. Індексція сайту.

21. Адаптивна верстка.

22. Валідність коду.

23. Протокол https.

## **8.8. Подальша підтримка і оновлення**

Яким би хорошим не був сайт, його слід періодично поповнювати новою інформацією, щоб не втратити відвідувача. Підтримкою і оновленням комерційних сайтів, як правило, займається теж веб-фахівець. З одного боку це універсал, від якого потрібні знання роботи з графікою (поверхневе), хороші знання в області кодингу (HTML, CSS) і знання мов програмування (поверхневе). З іншого боку веб-фахівець не має глибоких знань ні в області програмування, ні в області дизайну, тобто, не може замінити ні дизайнера, ні програміста при розробці серйозного комерційного ресурсу. Підтримкою ж особистих проектів доведеться займатися самостійно, правда, з часом, якщо ресурс хороший, можуть з'явитися помічники-добровольці.

Але, відвідувача цікавить не лише нова інформація. Важливо підтримувати зв'язок з відвідувачами: спілкування на форумі або в гостьовій книзі, які розташовуються на сайті, за можливістю, відповідати на всі листи відвідувачів.

При розвитку ресурсу іноді необхідно орієнтуватися на думку відвідувача: запитувати у відвідувачів, в яких матеріалах вони зацікавлені, що їм хотілося б побачити на сайті найближчим часом. Опитування можуть допомогти в розвитку сайту.

Потрібно стежити за сайтами з подібною тематикою, і прагнути бути на належному рівні. Корисним є також відстеження за тенденціями в дизайні і новими технологіями: що було добре кілька років тому, сьогодні може виявитися застарілим і безглуздом.

### ***Вимоги до виконання самостійної роботи***

Самостійною роботою є створення статичного або динамічного адаптивного HTML-сайту на довільну тему об'ємом 4-5 сторінок. Мова сайту - українська (для підтвердження власної роботи). Наявність авторської сторінки за посиланням © Прізвище або нік (в нижній частині сторінок). На авторській сторінці поставити фотографію (аватар) і написати коротку інформацію про себе.

Створений сайт розміщується на хостингу і демонструється викладачеві онлайн.

### **Перелік тем для виконання самостійної роботи**

1. Портфоліо робіт певного фахівця (дизайнер, фотограф, відео-оператор, кулінар тощо).
2. Послуги умовної компанії (бізнес, медицина, перевезення, розваги, готелі, кафе тощо).
3. Візитка відомої особи: письменника, спортсмена, актора, співака, музичного гурту, політичного діяча.
4. Галерея творчих робіт: фотографії, картини, книги, пісні чи інше.
5. Новинки технічних досягнень: комп'ютери, телефони, автомобілі, побутова техніка, роботи.
6. Міста та країни світу. Улюблені краєвиди, історичні екскурси, цікаві відомості.
7. Хобі студента: спорт, творчість, навчання, домашні тварини та інше.

8. Паблік для популярного фільму, серіалу, мюзіклу чи творчої спільноти.
9. Електронна бібліотека із збіркою певних книжок, статей, коміксів тощо.
10. Домашня сторінка студента, де викладено його захоплення, мандрівки, фотографії друзів, домашніх тварин та інша відкрита інформація.

### **Вимоги до HTML-сайту**

1. Головна сторінка сайту має назву index.html. Назви файлів (сторінок, картинок, скриптів, стилів) та папок – латинкою, малий регістр, без пробілів (для візуального розмежування застосовувати символи «-» або «\_»). Кирилицю для назв не застосовувати!!!!
2. Застосувати таблицю стилів в окремому файлі. Опис стилів зі сторінки перенести в файл таблиці стилів \*.css. Назви стилів – латинкою, змістовні (.red .big тощо).
3. Графіка, що відноситься до дизайну сторінок міститься в окремій директорії (зазвичай, img або images). Решта зображень, що відносяться до наповнення сторінок – в інших директоріях зі змістовними назвами (наприклад, picture, photo тощо).
4. Верстання сайту блочне із застосуванням тегів HTML5. Кодування сторінки має підтримувати кирилицю (charset=utf-8).
5. Наявність основних тегів у коді (<html><head><body><div><p><h1>...<h6> тощо). Парні теги мають закриватися (<p></p>). Зберігати вкладеність тегів (<div><p>Текст</p></div>). Основні модулі коду сторінки мають бути закоментовані (<!--Comment is here-->).
6. Єдиний стиль для всіх сторінок сайту. Однакове розміщення основних блоків сайту (заголовку, рубрик, логотипу).
7. Наявність навігації на всіх сторінках сайту. Виключення лише для сторінок, що відкриваються у новому вікні і містять інформацію, яка може сприйматися і не в контексті сайту (збільшена картинка, інформація про розробника, стаття, тощо).
8. Для адаптивної структури сторінок застосувати сітку, користуючись Bootstrap, Grid, Flexbox або власними засобами.

### **Реалізовані сервіси на сторінках HTML-сайту**

1. Наявність логотипу, фавіконки та назви сторінки на вкладці.
2. Застосувати веб-шрифти для основного тексту та заголовків. Для іконок використати спеціалізовані іконкові шрифти або SVG-зображення з відповідних ресурсів.
3. На сторінках сайту розмістити зображення різних форматів (растрові та векторні). У відповідних місцях втілити SVG- CSS- або JS- анімовані зображення.
4. Втілити у сайт інтерактивну карту або панорамний перегляд певної місцевості. Це може бути ваш будинок, історичне місце, університет. Втілення з YouTube власного або доречного відео з обраної тематики. Відредагувати відео засобами YouTube-редактора: додати титри, ефекти, музику тощо.

5. Додавання віджета (наприклад, новини, афоризми, анекдоти), інформера (прогноз погоди, курс валют, календар подій) та кнопку певної соцмережі.
6. Наявність динамічного елемента (випадне меню, слайдер, збільшення фотографій, акордіон тощо) з бібліотеки jQuery або як результат під'єднання плагіну.
7. Розмістити сайт на платному або безкоштовному хостингу.

### **Вимоги до CMS-сайту**

1. Мова сайту - українська (для підтвердження власної роботи).
2. Створити власний або адаптувати готовий шаблон.
3. Сайт має містити 4-5 сторінок певної тематики. Єдиний стиль для всіх сторінок сайту. Однакове розміщення основних блоків сайту (заголовку, рубрик, логотипу).
4. Наявність зручної навігації, логотипу, фавіконки та назви сторінки на вкладці.
5. Наявність авторської сторінки за посиланням © Прізвище або нік (в нижній частині сторінок). На авторській сторінці поставити фотографію (аватар) і написати коротку інформацію про себе.
6. Для певних елементів сторінок застосувати власні стилі та скрипти.
7. Наповнити сторінки відповідними зображеннями.

### **Реалізовані елементи та сервіси на сторінках CMS-сайту**

1. Реалізація на сайті популярних сервісів, що надає дана CMS: е-мейл, архів статей, календар, пошук, коментарі тощо.
2. Застосувати веб-шрифти для основного тексту та заголовків. Для іконок використати спеціалізовані іконкові шрифти або SVG-зображення з відповідних ресурсів.
3. Втїлити у сайт інтерактивну карту засобами CMS або скористатися GoogleMaps API.
4. Встановити відео з YouTube, Vimeo або власного хостингу.
5. Додавання віджета (наприклад, новини, афоризми, анекдоти), інформера (прогноз погоди, курс валют, календар подій) та кнопку певної соцмережі.
6. Додати динамічні елементи (випадне меню, слайдер, збільшення фотографій, акордіон тощо).
7. Під'єднати доречний до теми плагін, який можна активувати на CMS.

### **Порядок роботи**

1. Обрати тему сайту.
2. Визначитися з колірною гамою, стилем, логічною та фізичною структурою сайту.
3. Зверстати шаблонну сторінку, перевірити цілісність структури, наявність всіх основних блоків сторінки (окремі модульні блоки закоментувати), правильність посилань.
4. За шаблонною сторінкою створити решта сторінок, об'єднати їх навігацією
5. Додати до сторінок сервіси згідно вимог.

6. Розмістити сайт в Інтернеті.

### *Зміст звіту*

1. Тема розробленого сайту.
2. Адреса розміщення.
3. Застосовані засоби, сервіси та технології (мови, плагіни, API тощо).
4. Скрін головної сторінки.
5. HTML код головної сторінки.
6. Фрагмент коду CSS файлу.
7. У висновку оцінити процес створення сайту, доступність та ефективність застосованих засобів і сервісів, якісні характеристики хостингу, зручність ftp-клієнта та інші моменти робочого процесу.

### *Використані джерела*

1. Колекція авторських відеоуроків  
<https://youtube.com/playlist?list=PLY7ISgSf9ivVX-rfX4OEaq84TeODgFjZK>
2. Створення статичного сайту <https://youtu.be/aJEgrVRFNd4>
3. Створення динамічного сайту <https://youtu.be/aJEgrVRFNd4>
4. Довідник з HTML, CSS, JavaScript, Bootstrap, SVG <https://www.w3schools.com/>
5. Довідник з HTML, CSS, JavaScript, Bootstrap <https://html5css.ru/>
6. Довідник з HTML, CSS, JavaScript, Bootstrap <https://html5book.ru>
7. Довідник з HTML та CSS <http://htmlbook.ru/>

## Додаток А. Основи HTML

HTML - не є мовою програмування, це мова розмітки, яка використовується для розміщення елементів (тексту, картинок, кнопок, таблиць тощо) на сторінці, тобто, для складання структури. Сторінка може мати складну або просту структуру, все залежить від досвіду веб-розробника.

HTML складається з ряду елементів, які використовують для того, щоб різні частини сторінки мали певний вид або спрацьовували певним способом. За призначення, вигляд та розташування елемента відповідають спеціальні оператори (теги), які представляють собою команду, що поміщена в кутові дужки. Теги, зазвичай, є скороченою назвою або аббревіатурою того завдання, що має виконати браузер наприклад:

```
<p> - тег, що позначає абзац (paragraph).  
<tr> - тег, що позначає рядок таблиці (table row).  
<h1> - тег, що позначає заголовок 1 рівня (header 1 level).
```

Для збільшення інформативності тегу використовуються атрибути, що вказують текстові чи числові значення.

```
<table width="50%" align="center">...</table>
```

В наведеному прикладі показано тег таблиці, яка має займати 50% від ширини вікна браузера (або блоку, в якому знаходиться) і розміщена по центру.

В мові HTML застосовують теги: одинарні, парні та коментарі.

### Одинарні теги HTML

Одинарні теги складаються з одного оператора. Наприклад, тег `<br>` - перенесення на новий рядок, `<hr>` - розділова лінія, `<img>` - уставка зображення. Властивості для одиночного тегу вказуються всередині через атрибути, наприклад:

```

```

В наведеному прикладі в одиночному тезі `<img>` прописано атрибути, які вказують на місце розташування і назву графічного файлу, ширину зображення в 200 пікселів і наявність рамки навколо зображення шириною в 1 піксел.

Певний час вимагалось наприкінці одиночного тегу вказувати слеш (`<br />`, `<img />`), натепер такий стиль написання вже не є обов'язковим.

### Парні теги HTML

Парних тегів є значно більше і вони вказують браузеру початок і закінчення певного структурного елемента: рядка, блоку, таблиці, заголовку

Наприклад, для того, щоб певний текст відобразився в браузері як рядок, слід застосувати теги, що позначають елемент-параграф (рис.1):



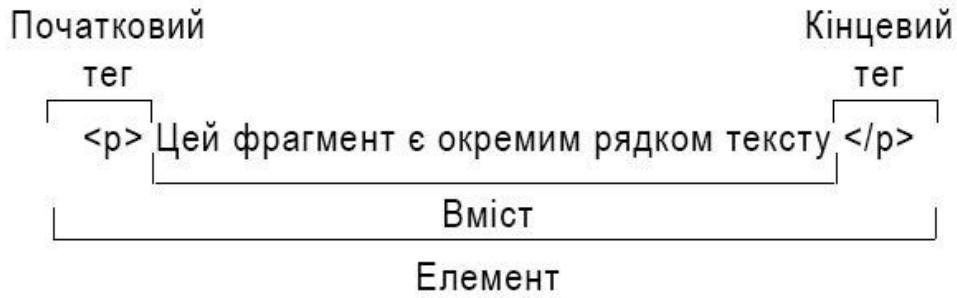


Рис. 1. Склад HTML-елементу

Основними частинами елемента є:

- **Початковий тег:** Він складається з назви елемента (<p>) і є ознакою початку елемента, з цього моменту тег починає впливати на той вміст, що слідує за ним.
- **Кінцевий тег:** виглядає як і початковий, але містить слеш перед назвою тега (</p>). Він вказує на закінчення елемента. Закінчення в парних тегах є обов'язковими, інакше браузер може невірно відобразити вміст.
- **Вміст:** в цьому випадку вмістом є простий текст.
- **Елемент:** початковий тег + вміст + кінцевий тег = елемент.

Всередині елементів можна вкладати інші елементи - це називається вкладеністю. Наприклад, якщо потрібно зробити акцент на певному слові і виділити його грубішим шрифтом.

```
<p>Цей фрагмент є <strong>окремим рядком</strong> тексту</p>
```

Якщо застосовується розміщення елементів всередині інших, слід обов'язково зберігати порядок вкладеності. В наступному прикладі наведено невірний запис:

```
<p>Цей фрагмент є <strong>окремим рядком тексту</p></strong>
```

Елементи повинні відкриватися і закриватися правильно таким чином, щоб явно перебувати всередині або зовні один одного. Якщо вони перекриваються так, як в прикладі вище, то браузер сам приймає рішення як відобразити, що може привести до непередбачуваного результату.

### Теги для коментарів

Коментування в HTML необхідно для покращення читабельності коду. В коментарях, зазвичай, вказується пояснення ділянки коду, що спрощує процес редагування HTML сторінки в подальшому. Розробнику легше орієнтуватися і вносити зміни, оскільки коли коду стає дуже багато, то в ньому легко заплутатися, можна поставити зайвий тег або, навпаки, не закрити його.

Тег для коментаря починається з кутової дужки, знак оклику, два дефіси, текст коментаря, два дефіси, закінчується кутовою дужкою.

```
<!-- текст коментаря -->
```

Коментарі в HTML не відображаються на сторінці в браузері користувачеві, їх можна побачити лише в коді веб-сторінки. Для зручності слід коментувати не лише початок певного блоку коду, а й його завершення, наприклад:

```
<!-- NAVIGATION -->
    <nav> ... </nav>
<!-- END NAVIGATION -->
```

## Структура HTML-документа

Для сторінок будь якої складності існує стандартна структура, яка має обов'язкові елементи.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8">
    <title>Тестова сторінка</title>
  </head>
  <body>
    <p>Моя перша сторінка</p>
  </body>
</html>
```

- **<!DOCTYPE html>**: посилання на набір правил doctypes, яким має слідувати HTML-сторінка, також вказівка на автоматичну перевірку помилок і інші корисні речі.
- **<html>** є кореневим елементом документа. Всі інші елементи містяться всередині тегів <html> ... </html>. Все, що знаходиться за межами цих тегів, не сприймається браузером як код HTML і не обробляється..
- **<head> елемент** містить службову, довідкову та додаткову інформацію, яка призначена для браузера, пошукових систем, сервера і не відображається для відвідувача сторінки. В цьому елементі міститься технічна інформація про сторінку: ключові слова, короткий опис сторінки, під'єднані шрифти, стилі, скрипти, зазначається тип кодування шрифтів і багато іншого.
- **<body> елемент** містить весь контент, який бачать відвідувачі - текст, зображення, відео, анімація тощо.

Елементи, що знаходяться всередині елементу <html>, утворюють дерево документа, так звану об'єктну модель документа, **DOM (Document Object Model)**. Елемент <html> є кореневим елементом, body і head – структурними елементами (рис.2).

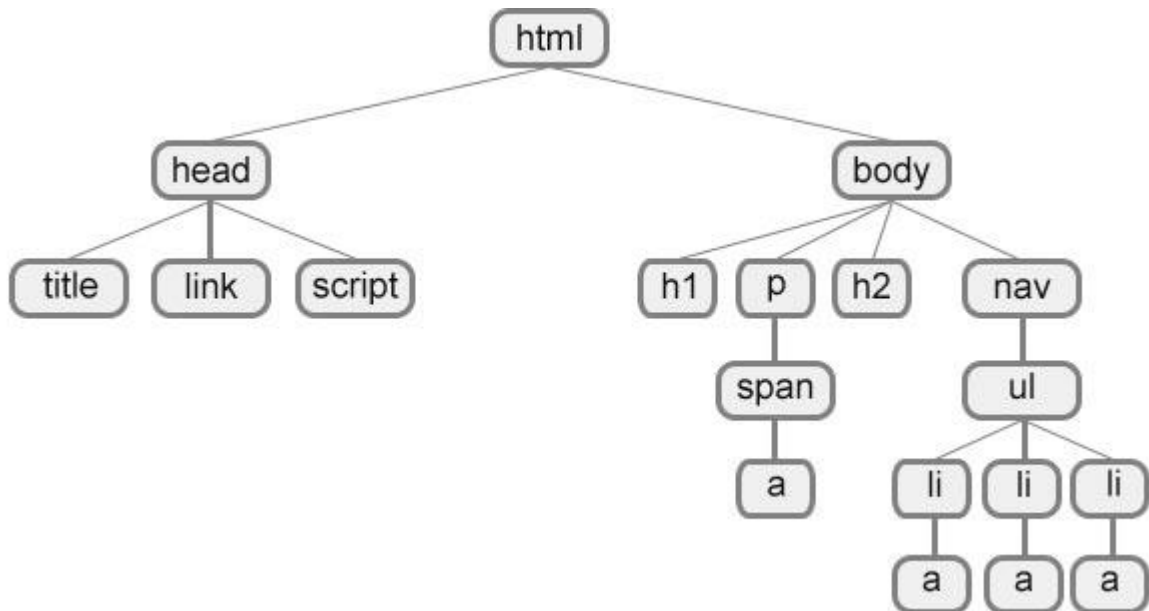


Рис. 2. Найпростіша структура веб-сторінки

У взаємодії елементів веб-сторінки задіяні "родинні стосунки" між елементами. Відносини між множинними вкладеними елементами підрозділяються на батьківські, дочірні та сестринські.

- **Батьківський елемент** - елемент, що пов'язаний з іншими елементами нижчого рівня, і що знаходиться на дереві вище за них. На рис.2 <html> є батьківським лише для <head> і <body>. Елемент <body> є батьківським для всіх елементів, що містяться в ньому: <h1>, <p>, <span>, <nav> і т.д. Тег <p> є батьківським тільки для <span>.
- **Дочірній елемент** - елемент, що розташований всередині іншого елемента. На рис.2 елементи <h1>, <h2>, <p> і <nav> є дочірніми по відношенню до <body>.
- **Сестринський елемент** - елемент, що має загальний батьківський елемент з даним, так звані елементи одного рівня. На рис.2 <head> і <body> - елементи одного рівня, так само як і елементи <h1>, <h2> і <p> є між собою сестринськими.

## Елемент <head>

Інформація всередині тегу не відображається у вікні браузера, однак, ці дані вказують браузеру, серверу чи пошуковій системі, як слід обробляти сторінку.

- **<title> елемент** є обов'язковим для розділу <head>, оскільки текст, що розміщений всередині нього, відображається у заголовку браузера. Текст заголовку повинен містити максимально повний опис вмісту сторінки.
- **<meta> елемент.** Елемент <head> може містити кілька елементів <meta>, які відповідно до використаних атрибутів несуть різну інформацію. За допомогою <meta> можна зазначити короткий опис сторінки і ключові слова для пошукових машин, автора html-документу та інші відомості.
- **<meta charset="тип кодування">** повідомляє браузеру кодування сторінки. Кодування HTML-сторінки потрібно вказувати для того, щоб веб-браузер міг правильно відображати текст на сторінці. Якщо браузер неправильно «вгадає» кодування, то

замість тексту будуть відображатися ієрогліфи. Стандартом на сьогодні є кодування - utf-8, що містить символи зі всіх відомих природних мов, тому цей тег має вигляд `<meta charset=" utf-8">`.

- **Елемент `<style>`**. Всередині цього елемента задаються стилі, які використовуються на сторінці.
- **Елемент `<link>`**. Даний елемент визначає відношення між поточною сторінкою та іншими документами. Таких елементів на сторінці може бути кілька. Наприклад, вказати зв'язок сторінки з файлом зі стилями (окремий файл з розширенням `.css`), тоді посилання на файл зі стилями буде виглядати так `<link href="style.css">`. Обов'язковим є вказування атрибуту `rel`, де вказується як цей документ буде розпізнано браузером (`<link rel="stylesheet" href="style.css">` - документ є таблицею стилів, `<link rel="icon" href="favicon.png">` - документ є фавіконкою). Додатково вказують тип документу атрибутом `type` (`<link rel="stylesheet" type="text/css" href="style.css">`, `<link rel=" icon" type="image/x-icon" href="favicon.png">`)
- **Елемент `<script>`**. Дозволяє приєднувати до документа різні сценарії. Закриваючий тег є обов'язковим, при цьому текст сценарію може розташовуватися або всередині цього елемента, або в зовнішньому файлі. Якщо текст сценарію розташований в зовнішньому файлі, то він під'єднується за допомогою атрибутів елемента, наприклад: `<script src="scripts.js"></script>`

## Елемент `<body>`

Всередині елемента `<body>` міститься інформація, яка відображається в браузері. В додатку В наведено повний перелік елементів, що підтримуються HTML5.

За призначенням теги можна умовно розподілити:

- **Контейнерні:** `div`, `article`, `aside`, `header`, `footer`, `span`, `p`, `h1-h6`, `ul`, `ol`
- **Функціональні:** `a`, `img`, `table`, `form`, `input`, `audio`, `video`, `canvas`
- **Фрейми:** `iframe`

У тегів можуть бути атрибути, які повідомляють браузеру, яким чином має відображатися той чи інший елемент сторінки та особливості відображення даного елемента. Значення атрибута завжди заключають в лапки " ". Назви та значення атрибутів не є чутливими до регістру, але, рекомендується набирати їх у нижньому регістрі.

```
<тег атрибут1="значення1" атрибут2="значення1 значення2">контент</тег>
```

Атрибути можна поділити на **універсальні (глобальні)** які можна використані для будь-якого HTML-елемента, та **власні**, які притаманні лише цьому тегу.

*Універсальними атрибутами є*

- **title** Вказує додаткову текстову інформацію про елемент, що відображається у спливаючій підказці над елементом, наприклад: `<div title="Червоний блок">`.

- **style** Вказує код CSS, що застосовується для оформлення даного елемента, наприклад: `<div style="width: 100%; background:red; color:#ffffff;">`.
- **class** Вказує назву класу для елемента, сам клас описано в таблиці стилів, наприклад: `<div class="red_box">`. Атрибут *class* застосовний до всіх HTML-елементів. Кожному конкретному елементу можна привласнити **кілька значень class**. Множинні значення *class* записуються через пробіл, наприклад, `<div class = "nav top">`. Назви класу повинні складатися з букв, цифр, дефісів і нижніх підкреслень і починатися тільки з літери.
- **id** Вказує унікальний ідентифікатор елемента, наприклад: `<div id="first_item">`. Атрибут *id* застосовний до всіх HTML-елементів. Кожному конкретному елементу можна привласнити **лише одне значення id**. Назви *id* повинні складатися тільки з букв, цифр, дефісів і нижніх підкреслень і повинні починатися тільки з літери.

*Умовно елементи можна поділити на групи*

- **Блокові елементи.** Характеризуються тим, що починаються з нового рядка, займають всю доступну ширину, висота елемента визначається його вмістом.
- **Рядкові елементи.** Є безпосередньою частиною іншого елемента, наприклад, текстового абзацу. В основному використовуються для зміни вигляду тексту, його логічного виділення чи акцентування.

## **Блокові елементи**

*Загальні характеристики блокових елементів*

- На початку і в кінці блокових елементів, автоматично ставиться перенесення рядка.
- За замовченням блокові елементи займають всю ширину батьківського елемента.
- В блокових елементах можна керувати шириною і висотою: `width`, `height`.
- В блокових елементах можна керувати зовнішніми і внутрішніми відступами.
- Об'єктами, що знаходяться всередині блокового елемента, можна керувати за допомогою горизонтального та вертикального вирівнювання.
- Всередині блокових елементів можна розміщувати інші блокові елементи, а також рядкові елементи.

*Типові блокові елементи*

- `<div>` Універсальний блоковий елемент.
- `<article>`, `<aside>`, `<header>`, `<footer>` Семантичні блоки для позначення структурних ділянок сторінки.
- `<p>`, `<h1>`-`<h6>`, `<blockquote>` Елементи для HTML тексту.

- **<hr>** Горизонтальна лінія.
- **<ol>**, **<ul>** Встановлює нумерований (маркірований) список.
- **<table>** Створює таблицю.
- **<form>** Встановлює форму на веб-сторінці.

## Рядкові елементи

### *Загальні характеристики рядкових елементів*

- На відміну від блокових, рядкові елементи слідують один за одним і НЕ переносяться на новий рядок.
- В рядкових елементах не можна керувати шириною і висотою: width, height.
- В рядкових елементах можна встановлювати лише внутрішні відступи (padding) та зовнішні бічні відступи (margin).
- Рядкові елементи НЕ займають всю ширину батьківського елемента.
- Оскільки рядкові елементи НЕ займають всю ширину батьківського елемента, то елементами, що знаходяться всередині рядкових елементів, не можна керувати за допомогою горизонтального чи вертикального вирівнювання.
- Всередині рядкових елементів можна розміщувати інші рядкові елементи, якщо розмістити там блоковий елемент, то він розірве логічний потік і розміститься з нового рядка.

### *Типові рядкові елементи*

**<a>** Є одним з важливих елементів HTML і призначений для створення посилань.

**<span>** Універсальний рядковий елемент.

**<br>** Розірвання рядка.

**<img>** Уставлення зображення.

## *HTML текст*

Елементи для форматування тексту несуть змістовне навантаження і, зазвичай, задають для тексту структурне та стильове оформлення, наприклад, виділяють текст грубішим шрифтом або відображають його іншим шрифтом.

Грамотно відформатований текст надає для пошукових систем відомості, які фрагменти тексту мають важливий зміст, за якими з них слід ранжувати веб-сторінку в пошуковій видачі.

### *Елемент абзацу <p>*

Розділяє текст на окремі абзаци, відокремлюючи їх збільшеною відстанню. Браузер автоматично додає верхній і нижній відступ.

## Елементи заголовків <h1>-<h6>

Заголовки є важливим елементом веб-сторінки, вони допомагають впорядкувати текст, сформуванню його структури. У специфікації HTML існує шість рівнів заголовків, завдяки яким можна легко виділяти теми і підтеми.

Заголовки є блоковими елементами, завжди починаються з нового рядка. Браузер автоматично додає перед заголовком і після нього збільшені відступи.

Текст в заголовках впливає на індексацію сайту пошуковими системами, оскільки багато роботів звертають увагу саме на вміст заголовків сайту, тому, краще завжди використовувати ці елементи, міняючи зовнішній вигляд і розмір за допомогою стилів.

При використанні заголовків необхідно враховувати їх ієрархію, тобто за <h1> повинен слідувати <h2> і т.д. Також, не рекомендується вкладення інших елементів в <h1> ... <h6>.

## Елементи для форматування тексту

- **<strong>** задає грубіше накреслення шрифту, вказуючи браузеру на важливість тексту. Є сучасною альтернативою для попереднього елемента <b>.
- **<em>** відображає текст нахиленим шрифтом (курсивом). Є сучасною альтернативою для елемента <i>, який на сьогоднішній день охоче застосовують для розміщення іконок. Для нахилу тексту можна використати елемент <cite>.
- **<q>** додавання зовнішніх лапок до фрагменту.
- **<code>** зміна шрифту на більш технічний шрифт.
- **<del>** перекреслений текст.
- **<ins>** підкреслений текст.
- **<sub>** використовується для створення нижніх індексів. Зсуває текст нижче рівня рядка, зменшуючи його розмір.
- **<sup>** використовується для створення ступенів. Зсуває текст вище рівня рядка, зменшуючи його розмір.
- **<pre>** дозволяє вивести текст на екран, зберігши початкове форматування. Пробіли і переноси рядків при цьому не видаляються.
- **<blockquote>** Використовується для оформлення фрагменту особливим блоком, наприклад, цитати, виділяючи його відступами і переносами рядків.
- **<br>** Переносить текст на наступний рядок, створюючи розрив рядка.
- **<hr>** Використовується для поділу контенту на веб-сторінці. Відображається у вигляді горизонтальної лінії.

## HTML посилання

HTML посилання складають основу Веб, вони пов'язують між собою різні веб-сторінки.

Посилання, або гіперпосилання, створюються за допомогою тегу <a>. Посилання складається з двох частин: покажчика посилання та адресної частини посилання, наприклад:

```
<p>Замовити книжки у <a href="http://shop.ua">магазині</a></p>
```

Покажчик посилання представляє фрагмент тексту або зображення, який візуально виділяється в документі (за замовчуванням, синім кольором і підкресленням).

Адресну частину посилання для користувача не видно, вона представляє URL-адресу об'єкта, до якого необхідно перейти.

## Уставляння гіпертекстових посилань в HTML-документ

### *Uniform Resource Locator, URL*

Єдиний покажчик ресурсів (URL) визначає місцезнаходження ресурсу. В загальному вигляді має наступний формат:

```
метод://IP адреса сервера|доменна адреса сайту/шлях#якір.
```

Адреси описують, де знаходиться ресурс, наприклад, [www.site.ua](http://www.site.ua). Якщо адреси не вказано, то посилання вважається локальним, тобто, воно відноситься до тієї ж машини, на якій знаходиться html-документ, що містить посилання.

Далі вказується шлях (частковий або повний), за яким здійснюється перехід.

Під якорем розуміють посилання на місце всередині поточного html-документа.

Сам текст URL не відображається браузером, він використовується для виконання запропонованих дій при активації посилання (зазвичай, клацанням миші).

Найбільш поширені методи доступу: [http](http://) ([https](https://)), [mailto](mailto:), [tel](tel:), [viber](viber://).

- **Метод [http](http://) ([https](https://))** надає доступ до веб-сторінки за протоколом HTTP, наприклад:  
<a href="https://lpnu.ua/">ЛУ "Львівська політехніка"</a>
- **Метод [mailto](mailto:)** запускає сеанс поштового зв'язку із зазначеним адресатом і хостом, наприклад:  
<a href="mailto:Iryna.Y.Yurchak@lpnu.ua">Iryna.Y.Yurchak@lpnu.ua</a>
- **Метод [tel](tel:)** забезпечує з'єднання з програмою для здійснення телефонних дзвінків. Такий метод спрацьовує в мобільних пристроях і користувач може відразу позвонити за вказаним номером, наприклад: <a href="tel:0679476121">(067) 947-61-21</a>
- **Метод [viber](viber://)** здійснює з'єднання з месенжером Viber, наприклад:  
<a href="viber://chat?number=+380679476121">

## Абсолютний URL

Абсолютний шлях містить всю інформацію, що необхідна браузеру для знаходження файлу: назву протоколу, доменну або IP-адресу комп'ютера (хосту), папку (шлях до файлу) та ім'я файлу. Наприклад,

```
https://www.site.ua/folder/file.html
```

Якщо файл знаходиться в кореневій папці, то він вказується відразу:



`https://www.site.ru/file.html`

У разі відсутності назви файлу буде завантажуватися веб-сторінка, яка задана за замовченням у налаштуваннях веб-сервера (зазвичай, це `index.*`):

`http://www.site.ua/`  
`http://www.site.ua/folder/`

Наявність закриваючого слешу / означає, що звернення йде до папки, а якщо його немає - то безпосередньо до файлу.

## Відносний URL

Відносна URL-адреса описує шлях до зазначеного документа відносно поточного, вона визначається з врахуванням місця розташування веб-сторінки, на якій знаходиться посилання. Відносні посилання використовуються при створенні посилань на інші документи на одному сайті. У такому випадку не вказується протокол або домен, а тільки сам шлях до файлу.

1. Наприклад, у разі переходу зі сторінки <http://site.ua/folder1/folder2/file1.html> на сторінку <http://www.site.ua/folder1/folder2/file2.html>, тобто потрібна веб-сторінка знаходиться в тій же папці, що і веб-сторінка, що містить посилання, то посилання буде мати наступний запис:

```
<a href="file2.html">Текст посилання</a>
```

2. Якщо зі сторінки <http://www.site.ua/folder1/folder2/file1.html> потрібно перейти на сторінку <http://www.site.ru/folder1/folder2/folder3/file2.html>, то посилання буде таким:

```
<a href="folder3/file2.html">Текст посилання</a>
```

3. При переході зі сторінки <http://www.site.ua/folder1/folder2/file1.html> на сторінку <http://www.site.ua/folder1/file2.html> посилання буде мати наступний вигляд:

```
<a href="../../file2.html">Текст посилання</a>
```

3. У разі переходу з <http://www.site.ua/folder1/folder2/folder3/file1.html> на сторінку <http://www.site.ua/folder1/file2.html> посилання буде таким:

```
<a href="../../../file2.html">Текст посилання</a>
```

## Зовнішні посилання

Посилання на веб-сторінки інших сайтів є зовнішніми, для них завжди вказують Абсолютний URL, наприклад:

```
<a href="https://english-films.co/">Фільми англійською мовою</a>
```

## Посилання на розділи поточної сторінки

Внутрішні посилання посилаються на різні частини поточної сторінки, що сприяє швидкому переміщенню, коли на сторінці багато тексту.

Внутрішні посилання також вставляються за допомогою тегу `<a>` з різницею в тому, що атрибут `href` містить назву покажчика (ідентифікатор), а не URL-адресу. Перед назвою покажчика ставиться знак `#`.

Наприклад, наступний запис описує код зі швидкими переходами на відповідні розділи.

```
<h1>Пори року</h1>
```

```

<h2>Зміст</h2>
  <a href="#p1">Літо</a> <!--Задаємо якір через id елемента-->
  <a href="#p2">Осінь</a>
  <a href="#p3">Зима</a>
  <a href="#p4">Весна</a>
<p id="p1"> ... </p> <!--Додаємо відповідний id елементу-->
<p id="p2"> ... </p>
<p id="p3"> ... </p>
<p id="p4"> ... </p>

```

Якщо потрібно зробити посилання з однієї сторінки сайту на певний розділ іншої сторінки сайту, то необхідно задати id для цього розділу сторінки, а потім додати його до абсолютної або відносної адреси:

```


<a href="http://html5.ua/page.html#picture1">Вигляд картинки</a>

```

Або

```

<a href="page.html#picture1" target="_blank"> Вигляд картинки</a>

```

### Посилання на сторінки одного сайту

При створенні посилань на сторінки власного сайту задаються відносні посилання, які звертаються до певної сторінці на одному сервері.

Коли браузер не знаходить у посиланні протокол https://, він виконує пошук зазначеного документа на тому ж сервері. Відносні посилання описують шлях до потрібного файлу відносно розташування поточного документа.

У випадку, якщо сторінки знаходяться в одному каталозі, то замість запису

```

<a href="http://html5book.ua/styling.html">Оформлення гіпертекстових
посилань</a>

```

досить буде вказати ім'я файлу з розширенням (якщо воно задане в адресі сторінки)

```

<a href="styling.html">Оформлення гіпертекстових посилань</a>.

```

Якщо документи знаходяться в різних каталогах, і кінцевий файл знаходиться у вкладеному каталозі відносно поточного, необхідно вказати назву каталогу і назву сторінки через слеш, наприклад,

```

<a href="lessons/lesson1.html">Урок 1</a>

```

При створенні посилання на файли з батьківського каталогу, шлях до файлу повинен починатися з ../, що дослівно означає: повернутися на рівень вище і слідувати за зазначеним шляхом, аналогічно з кнопкою "Назад".

Можна задавати шлях до документів відносно кореневого каталогу. Звернення до кореневого каталогу здійснюється через символ /. Далі вказуються каталоги, в які браузер повинен перейти, щоб дістатися до потрібного документа.

```

<a href="/lessons/lesson1.html">Урок 1</a>

```

### Елементи, до яких застосовують посилання

Засобами HTML можна створювати посилання за допомогою тексту, зображень та блокових елементів, тобто, можна об'єднувати в одному посиланні текст, зображення та блоки:

```
<a href="http://summer.ua">
  <h1>ЛІТО</h1>
  
  <div>
    <ul>
      <li>Пропозиції по відпочинку</li>
      <li>Акції та знижки</li>
    </ul>
  </div>
</a>
```

## Атрибути посилань

Окрім глобальних атрибутів, елемент `<a>` підтримує власні атрибути.

- **href** - значенням атрибута є URL-адреса документа, на яку вказує посилання.
- **nofollow** - забороняє роботам пошукової системи переходити за посиланнями на даній сторінці або за конкретними посиланнями.
- **target** - вказує на те, в якому вікні повинна відкриватися сторінка або документ, до якого веде посилання. Приймає наступні значення:

**\_self** - сторінка завантажується до поточного вікна (вкладки). Значення за замовченням;

**\_blank** - сторінка відкривається в новому вікні (вкладці) браузера;

**\_parent** - сторінка завантажується у фрейм-батько;

**\_top** - сторінка завантажується в повне вікно браузера.

## HTML зображення

Використання графіки робить веб-сторінки візуально привабливими, зображення допомагають краще передати суть і зміст веб-документа. Також, за допомогою HTML-тегів можна робити карти-зображення з активними областями.

### Уставляння зображень в HTML-документ

#### Елемент `<img>`

Зображення додаються на сторінку за допомогою одинарного тега `<img>`. Оскільки елемент `<img>` є рядковим, то рекомендується розташовувати його всередині блокового елемента, наприклад, `<p>` або `<div>`.

Елемент `<img>` має обов'язковий атрибут **src**, значенням якого є адреса втіленого зображення, і рекомендований атрибут **alt**, наприклад:

```

```

Для елемента `<img>` доступні наступні атрибути:

- **src** - вказує URL-адресу зображення (де знаходиться зображення).

- **alt** - зазначає альтернативний текст для зображення. Виводиться на місці появи зображення до його завантаження або при відімкнутій графіці, в деяких браузерах виводиться спливаючою підказкою при наведенні вказівника на зображення. Цей атрибут є важливим для пошукових систем.
- **width** - задає ширину зображення. Прийняті значення: пікселі або відсотки.
- **height** - задає висоту зображення. Прийняті значення: пікселі або відсотки.

### *Адреса зображення*

Адреса зображення може бути вказана повністю (абсолютний URL), наприклад:

```

```

Або через відносний шлях від документа або кореневого каталогу сайту:

```
 - відносний шлях від документа,  
 - відносний шлях від кореневого каталогу.
```

### *Розміри зображення*

Без завдання розмірів зображення відображається на сторінці в реальному розмірі. Відредагувати розміри зображення можна за допомогою атрибутів `width` і `height`. Якщо буде задано лише один з атрибутів, то інший буде обчислюватися автоматично для збереження пропорцій малюнка.

### *Формати графічних файлів*

- Растровий формат **JPEG (.JPG)**. Зображення JPEG є ідеальними для фотографій, вони містять мільйони різних кольорів.
- Растровий формат **GIF**. GIF-файли дозволяють встановити один з кольорів 100% прозорим, завдяки чому фон веб-сторінки буде видно через частину зображення (напівпрозорі ділянки зафарбовуються в суцільний колір). GIF-файли можуть містити просту покадрову анімацію. Основний недолік у використанні індексованої палітри кольорів, GIF-зображення містять всього лише 256 кольорів, через що зображення виглядають плямистими і нереалістичного кольору.
- Растровий формат **PNG**. Містить кращі властивості GIF- і JPEG-форматів. Містить мільйони кольорів і надає можливість зберігати напівпрозорі та прозорі ділянки. Недоліком є значно більший розмір файлу, ніж при JPEG-форматі.
- Векторний формат **SVG**. SVG-зображення, що складається з набору геометричних фігур, які описані у форматі XML: лінія, еліпс, багатокутник тощо. Підтримується як статична, так і анімована графіка. Набір функцій містить різні перетворення, альфа-маски, ефекти фільтрів, можливість використання шаблонів. Зображення у форматі SVG можуть змінюватися в розмірі без зниження якості.
- Растровий формат **WebP**. Веб-формат від Google, що створено в 2010 році. Використовує потужні способи оптимізації зображень для зменшення ваги файлу, підтримує прозорість і анімацію. На сьогоднішній день формат підтримують популярні

браузери (Chrome, Firefox, Edge), у застарілих браузерів зображення в цьому форматі не відобразяться.

- Растровий формат **APNG**. Формат зображення, що засновано на форматі PNG від Mozilla в 2008 році. Дозволяє зберігати анімацію, а також підтримує прозорість. Дозволу з боку Portable Network Graphics формат APNG не отримав, тому його неможна віднести ані до веб-стандарт, ані до стандарту PNG та libpng. Тому, робота з даним форматом може бути зі збоями.
- Растровий формат **ICO**. Формат зберігання фавіконок, що відображаються на закладці в браузері або поруч з адресою сайту у сторінці видачі результатів пошукової системи. На сьогоднішній день формат вважається застарілим, а фавіконки застосовують у форматах png та svg. Найбільш вживані розміри фавіконок: 16, 32, 48, 57, 72, 114, 256 пікселів.

## HTML списки

HTML списки представляють набір згрупованих абзаців тексту, помічених значками (маркірований список) або цифрами (нумерований список). Елементи списку додаються за допомогою елемента `<li>` (List Item - елемент списку). Для елемента `<li>` доступні різні атрибути, що дозволяють змінити нумерацію або маркування обраного елемента списку.

### Маркірований список

Маркірований список представляє невпорядкований перелік елементів (Unordered List). Створюється за допомогою парного елемента `<ul>`. На початку кожного елемента встановлюється маркер, наприклад, зафарбований кружечок.

```
<ul>
  <li>Microsoft</li>
  <li>Google</li>
  <li>Apple</li>
  <li>IBM</li>
</ul>
```

- Microsoft
- Google
- Apple
- IBM

### Нумерований список

Нумерований список поміщають всередину пари тегів `<ol>`. Кожен пункт списку потрібно помістити в елемент `<li>`. Браузер автоматично проставляє номери елементів по порядку і якщо видалити один або кілька елементів такого списку, то інші номери будуть автоматично змінені.

Для тегу `<ol>` доступні наступні атрибути:

- **start** - задає початкове значення, від якого починається відлік нумерації, наприклад, конструкція `<ol start = "10">` першому пункту привласнить порядковий номер "10". Також можна одночасно задавати тип нумерації, наприклад, `<ol type = "I" start = "10">`.
- **type** - задає тип нумерації для використання в списку (у вигляді букв або цифр). Прийнятні значення:

1 - значення за замовчуванням, десяткова нумерація.

A - нумерація списку в алфавітному порядку, заголовні букви (A, B, C, D).

a - нумерація списку в алфавітному порядку, малі літери (a, b, c, d).

I - нумерація римськими великими цифрами (I, II, III, IV).

i - нумерація римськими малими цифрами (i, ii, iii, iv).

```
<ol start="5">
  <li>Microsoft</li>
  <li>Google</li>
  <li>Apple</li>
  <li>IBM</li>
</ol>
```

5. Microsoft  
6. Google  
7. Apple  
8. IBM

### Список визначень

Списки визначень описуються за допомогою елемента <dl>, який містить пари термін-визначення. Для додавання терміну застосовується блоковий елемент <dt>, а для уставляння визначення - <dd>.

Для елементів <dl>, <dt> і <dd> доступні глобальні атрибути.

```
<dl>
  <dt>Національний
університет «Львівська
політехніка»:</dt>
  <dd>ІКНІ</dd>
  <dt>Кафедри</dt>
  <dd>САПР</dd>
  <dd>ІСМ</dd>
  <dd>АСУ</dd>
</dl>
```

**Національний університет  
«Львівська політехніка»:**  
ІКНІ  
**Кафедри**  
САПР  
ІСМ  
АСУ

### Вкладений список

Іноді можливостей простих списків бракує, наприклад, при створенні змістів потрібно застосувати вкладені пункти. Код для створення вкладеного списку буде мати наступний вигляд:

```
<ul>
  <li>Пункт 1.</li>
  <li>Пункт 2.
  <ul>
    <li>Підпункт 2.1.</li>
    <li>Підпункт 2.2.
    <ul>
      <li>Підпункт
2.2.1.</li>
      <li>Підпункт
2.2.2.</li>
    </ul>
    <li>Підпункт
2.3.</li>
  </ul>
</li>
<li>Підпункт 2.3.</li>
</ul>
</li>
```

- Пункт 1.
- Пункт 2.
  - Підпункт 2.1.
  - Підпункт 2.2.
    - Підпункт 2.2.1.
    - Підпункт 2.2.2.
  - Підпункт 2.3.
- Пункт 3.

```
<li>Пункт 3.</li>  
</ul>
```

Більш докладну інформацію про HTML елементи можна дізнатися у специфікації мови та довідниках для веб-розробників.

1. Довідник з HTML, CSS, JavaScript, Bootstrap, SVG <https://www.w3schools.com/>
2. Довідник з HTML, CSS, JavaScript, Bootstrap <https://html5css.ru/>
3. Довідник з HTML, CSS, JavaScript, Bootstrap <https://html5book.ru>
4. Довідник з HTML та CSS <http://htmlbook.ru/>

## Додаток Б. Основи CSS

**CSS (Cascading Style Sheets)**, або каскадні таблиці стилів, описують правила відображення та розміщення окремого елемента веб-сторінки. Правила створення стилю складається з двох основних частин: **селектора** і **блоку оголошення** (рис.3).

Селектор повідомляє браузеру, який саме елемент форматувати, в блоці оголошення перелічено властивості форматування та їх значення.



Рис.3. Структура оголошення стилю елемента в CSS

### Додавання CSS до веб-сторінки

#### Вбудовані таблиці стилів

Вбудовані стилі знаходяться між тегами `<style> ... </style>`, що вставляються всередину елемента `<head>`. Вбудовані стилі діють тільки на сторінці, на якій вони містяться. На одній сторінці можна розміщувати довільну кількість вбудованих стилів:

```
<head>
  <style type = "text/css">
    h1, h2 {color: red; font-family: "Times New Roman", Georgia,
    Serif; line-height: 1.3em;}
  </style>
</head>
```

#### Внутрішні стилі елементів

Внутрішні стилі елементів не використовують селектори, опис стилю відбувається безпосередньо через атрибут `style` в початковому тезі елементу:

```
<p style="font-family:"Times New Roman", Georgia, Serif;
color:#70d7700;">Зверніть увагу на цей текст.</p>
```

#### Зовнішні таблиці стилів

Зовнішня таблиця стилів представляє текстовий файл з розширенням `.css`, в якому знаходиться весь набір стилів CSS. Задані в файлі стилі будуть працювати для всіх сторінок веб-сайту. Під'єднати зовнішній файл зі стилями можна в два способи:

*Прикріплення до веб-сторінки за допомогою тега `<link>`, вкладеного в тег `<head>`:*

```
<head>
  <link rel="stylesheet" type="text/css" href="style1.css">
  <link rel="stylesheet" type="text/css" href="style2.css">
```



```
</head>
```

де `rel="stylesheet"` вказує тип посилання (посилання на таблицю стилів), а `type = "text/css"` повідомляє браузеру тип даних, в даному випадку це текстовий файл, що містить CSS-код.

### Прикріплення до веб-сторінки за допомогою правила `@import`

Правило `@import` дозволяє завантажити зовнішню таблицю стилів. Щоб директива `@import` працювала, вона повинна розташовуватися всередині тегу `<style>` перед іншими правилами:

```
<style type="text/css">
  @import url(mobile.css);
  p {font-size: 0.9em; color: grey;}
</style>
```

## Селектори

За допомогою селекторів створюються CSS-правила для форматування елементів сторінки. Як селектори можна використовувати елементи, класи, ідентифікатори, а також псевдокласи і псевдоелементи.

**Універсальний селектор.** Універсальний селектор позначає правила, що стосуються всіх елементів, наприклад, наступний запис обнулить відступи для всіх елементів веб-сайту:

```
* {Margin: 0;}
```

**Селектор Елементу.** Селектори елементів використовуються для визначення стилів для всіх даних елементів сайту, наприклад, стиль заголовків `h1` або загальний стиль абзаців:

```
h1 {font-family: Lobster, cursive;}
p {letter-spacing: 0.1em;}
```

**Селектор Класу.** Селектори класу використовуються для визначення стилів, які можна застосувати для різних елементів, розміщених в різних частинах або на різних сторінках сайту.

Код HTML	Код CSS
<pre>&lt;h1 class="headline"&gt;Інструкція користування персональним комп'ютером&lt;/h1&gt; &lt;p class="headline"&gt;Примітка. Це важливо для роботи&lt;/p&gt;</pre>	<pre>headline { text-transform: uppercase; color: lightblue; }</pre>

**Селектор Ідентифікатора.** Селектори ідентифікатора використовуються для привласнення стилю одному конкретному елементу. Ідентифікатор `id` належить унікальному елементу, тому його можна використовувати в документі лише один раз.

```
#sidebar {text-transform: uppercase; color: lightblue;}
```

**Селектор Нашадку.** До нащадків елемента відносяться його дочірні елементи. Селектори нащадків дозволяють стилізувати всі вкладені елементи, наприклад, можна відформатувати зовнішній вигляд всіх елементів маркованого списку:

```
ul li {text-transform: uppercase;}
```

Якщо потрібно відформатувати нащадки певного елемента, то можна поставити йому стильовий клас:

**p.first a {color: green;}** - означає, що потрібно застосувати даний стиль до всіх посилань, нащадків абзацу, що відносяться до класу з назвою first;

**p .first a {color: green;}** - якщо додати пробіл, то будуть обрані посилання, розташовані всередині будь-якого тегу класу .first, який є нащадком елемента <p>;

**.first a {color: green;}** - даний стиль застосовується до любого посилання, що розташоване всередині інших тегів, позначених класом .first.

**Дочірній селектор.** Дочірній тег є прямим нащадком тегу, що його містить. Тобто, відносини "батьки-діти" існують між елементами і тими елементами, які містяться безпосередньо всередині них. В одного елемента може бути кілька дочірніх елементів, а батьківський елемент може бути в кожного елемента тільки один.

**p> strong** - вибирає всі елементи strong, які є дочірніми по відношенню до елементу p.

**Сестринський селектор.** Сестринські відносини виникають між елементами, що мають загального батька. Селектори сестринських елементів дозволяють вибрати теги з групи елементів одного рівня.

**h1 + p** - вибере всі перші абзаци, що йдуть безпосередньо за будь-яким тегом <h1>, не зачіпаючи решта абзаців.

**h1 ~ p** - вибере всі абзаци, які є сестринськими по відношенню до будь-якого заголовку h1 і йдуть після нього.

**Селектор Атрибуту.** Селектори атрибутів дозволяють формувати елементи на основі вибірки будь-яких атрибутів, що містяться в них або значень атрибутів, наприклад:

**[атрибут]** - вибирає всі елементи, для яких задано вказаний атрибут.

**img [alt]** - вибирає всі картинки, що містять атрибут alt.

**img [title = "flower"]** - вибирає всі картинки, назва яких містить слово flower.

## Псевдокласи

Псевдокласи дозволяють додавати особливі класи до елементів, вибираючи об'єкти, яких немає в структурі веб-сторінки, або які не можна вибрати за допомогою звичайних селекторів, наприклад, перша літера або перший рядок одного абзацу. Псевдокласи добре проілюстровані різними станами посилання, наприклад:

**a: link** – описує стиль невідвіданого посилання.

**a: visited** - описує стиль вже відвіданого посилання.

**a: hover** - описує вигляд елемента, над яким проходить вказівник мишки.

**a: focus** - описує вигляд елемента, над яким знаходиться (сфокусований) вказівник.

**a: active** - описує вигляд елемента, який активовано користувачем.

**Структурні псевдокласи.** Структурні псевдокласи формують дочірні елементи відповідно до зазначеного параметра в дужках, наприклад:

**:nth-child (odd)** - вибирає непарні дочірні елементи.

**:nth-child (even)** - вибирає парні дочірні елементи.

**:nth-child (3n)** - вибирає кожен третій елемент серед дочірніх.

**Структурні псевдокласи типу.** Вказують на конкретний тип дочірнього тегу:

**:nth-of-type ()** - вибирає елементи за аналогією з: nth-child (), при цьому бере до уваги тільки тип елемента.

**:first-of-type** - дозволяє вибрати перший дочірній елемент.

**:last-of-type** - вибирає останній дочірній елемент конкретного типу.

**:nth-last-of-type ()** - вибирає елемент заданого типу в списку елементів відповідно до зазначеного місцеположенням, починаючи з кінця.

**:only-of-type** - вибирає єдиний елемент зазначеного типу серед дочірніх елементів батьківського елемента.

**Псевдоелементи.** Псевдоелементи не є елементами сторінки, їх використовують для додавання вмісту, який генерується за допомогою властивості content, і для зміни зовнішнього вигляду частини елемента:

**:before** – додає певний вміст перед елементом.

**:after** - додає певний вміст після елемента.

### Комбінації селекторів

Щоб домогтися більш чіткого вибору елементів для форматування, можна не обмежуватися завданням одного типу селектора, а використовувати комбінації селекторів, наприклад:

**a [href] [title]** - вибере всі посилання, для яких задані атрибути href і title;

**img [alt \* = css]: nth-of-type (even)** - вибере всі парні картинки, альтернативний текст яких містить слово css.

### Угруповання селекторів

Можна застосувати один стиль до кількох елементів, причому обмежень за кількістю елементів немає. Для цього необхідно в лівій частині оголошення помістити через кому потрібні селектори, наприклад:

```
h1, h2, h3, h4 {color: tomato; background: white;}
```

### *Принцип каскадування і специфічність правила*

**Каскадування** представляє процес застосування різних правил до одного і того ж елемента. Більш конкретні правила мають пріоритет над більш загальними. Якщо у відношення одного і того ж елемента визначено кілька стилів, то в результаті до нього буде застосований останній з них.

Для кожного правила браузер обчислює специфічність селектора, і якщо у елемента є конфліктуючі оголошення властивостей, до уваги береться правило, що має найбільшу специфічність.

Значення специфічності складається з чотирьох частин: 0, 0, 0, 0. Специфічність селектора визначається наступним чином:

- для id додається 0, 1, 0, 0;
- для class додається 0, 0, 1, 0;
- для кожного елемента і псевдоелемента додається 0, 0, 0, 1;
- для стилю, який доданого безпосередньо до елемента - 1, 0, 0, 0;
- універсальний селектор не має специфічності.

```

h1 {color: lightblue;} /* специфічність 0, 0, 0, 1 */
em {color: silver;} /* специфічність 0, 0, 0, 1 */
h1 em {color: gold;} /* специфічність: 0, 0, 0, 1 + 0, 0, 0, 1 = 0, 0,
0, 2 */
#sidebar {color: orange;} /* специфічність 0, 1, 0, 0 */
li # sidebar {color: aqua;} /* специфічність: 0, 0, 0, 1 + 0, 1, 0, 0
= 0, 1, 0, 1 */

```

В результаті до елемента застосуються ті правила, специфічність яких більше, наприклад, якщо на елемент діють дві специфічності зі значеннями 0, 0, 0, 2 і 0, 1, 0, 1, то виграє друге правило.

Вагу правила також можна задати за допомогою ключового слова **!important**, яке додається після значення властивості, наприклад, **font-weight: bold!important**. Таке оголошення буде мати пріоритет над всіма іншими правилами.

Більше інформації по правилах CSS можна дізнатися з специфікації мови та довідників.

1. Довідник з HTML, CSS, JavaScript, Bootstrap, SVG <https://www.w3schools.com/>
2. Довідник з HTML, CSS, JavaScript, Bootstrap <https://html5css.ru/>
3. Довідник з HTML, CSS, JavaScript, Bootstrap <https://html5book.ru>
4. Довідник з HTML та CSS <http://htmlbook.ru/>

## Додаток В. Перелік поширених елементів HTML5

<!--...-->	Використовується для додавання коментарів.
<!DOCTYPE>	Оголошує тип документа та надає основну інформацію для браузера – його мову та версія.
<a>	Створює гіпертекстові посилання.
<abbr>	Визначає текст як аббревіатуру чи акронім. Пояснючий текст задається за допомогою атрибута title.
<address>	Задає контактні дані автора/власника документа чи статті. Відображається у браузері курсивом.
<article>	Розділ контенту, який утворює незалежну частину документа чи сайту, наприклад, стаття у журналі, запис у блозі, коментар.
<aside>	Контент сторінки, який має непряме відношення до основного контенту сторінки/сайту. Зазвичай застосовується до бічних колонок
<audio>	Завантажує звуковий вміст на веб-сторінку.
<b>	Задає напівжирне зображення уривка тексту, не надаючи акценту або важливості виділеному.
<base>	Задає базову адресу (URL), щодо якої обчислюються всі відносні адреси. Це допоможе уникнути проблем при перенесенні сторінки в інше місце, оскільки всі посилання працюватимуть, як і раніше.
<blockquote>	Виділяє текст як цитату, застосовується до опису великих цитат.
<body>	Представляє тіло документа (вміст, який не відноситься до метаданих документа).
 	Перенесення тексту на новий рядок.
<button>	Створення інтерактивної кнопки. Елемент може містити текст або зображення.
<canvas>	Полотно-контейнер для динамічного відображення зображень, таких як прості зображення, діаграми, графіки тощо. Для малювання використовується скриптова мова JavaScript.
<caption>	Додає підпис до таблиці. Вставляється відразу після тегу <table>.

<cite>	Використовується для визначення джерела цитування. Відображається курсивом.
<code>	Представляє фрагмент програмного коду, що відображається шрифтом сімейства monospace.
<col>	Вибирає для форматування один або кілька стовпців таблиці, які не містять інформацію одного типу.
<colgroup>	Створює структурну групу стовпців, що виділяє множину логічно однорідних комірок.
<datalist>	Елемент-контейнер для списку елемента <input>. Варіанти значень поміщаються в елементи <option>.
<dd>	Використовується для опису терміну з елемента <dt>.
<del>	Позначає текст як перекреслений.
<div>	Елемент-контейнер для розділів HTML-документів. Використовується для групування блокових елементів для форматування стилями.
<dl>	Елемент-контейнер, у якому знаходяться термін та його опис.
<dt>	Використовується для завдання терміну.
<em>	Виділяє важливі фрагменти тексту, відображаючи курсивом.
<embed>	Елемент-контейнер для вбудовування зовнішнього інтерактивного контенту або плагіну.
<fieldset>	Групує пов'язані елементи у вигляді, малюючи рамку навколо них.
<figcaption>	Заголовок/підпис для елемента <figure>.
<figure>	Елемент-контейнер для такого контенту як ілюстрації, діаграми, фотографії, приклади коду, зазвичай, з підписом.
<footer>	Визначає завершальну область (нижній колонтитул) документу чи розділу.
<form>	Форма для збору та надсилання на сервер інформації від користувачів. Не працює без атрибуту action.
<h1-h6>	Створюють заголовки шести рівнів для пов'язаних із ними розділів.
<head>	Елемент-контейнер для метаданих HTML-документів, таких як <title>, <meta>, <script>, <link>, <style>.

<header>	Секція для введення інформації сайту або групи навігаційних посилань. Може містити один або більше заголовків, логотип, інформацію про автора.
<hr>	Горизонтальна лінія для тематичного розподілу параграфів.
<html>	Кореневий елемент документа HTML. Повідомляє браузер, що це HTML-документ. Є контейнером для решти html-елементів.
<i>	Виділяє фрагмент тексту курсивом.
<iframe>	Створює контейнер, в який можна завантажуючи в поточний документ HTML інший документ.
<img>	Вбудовує зображення в HTML-документ за допомогою атрибута src, значенням якого є адреса зображення, що вбудовується.
<input>	Створює багатофункціональні поля форми, які користувач може вводити дані.
<ins>	Виділяє текст підкресленням. Застосовується виділення змін, внесених у документ.
<label>	Додає текстову позначку для елемента <input>.
<legend>	Заголовок елементів форми, що згруповані за допомогою елемента <fieldset>.
<li>	Елемент маркованого чи нумерованого списку.
<link>	Використовується для підключення зовнішніх файлів.
<main>	Контейнер для основного унікального вмісту. На одній сторінці має бути не більше одного елемента <main>.
<mark>	Виділяє фрагменти тексту, позначаючи їх жовтим тлом.
<meta>	Використовується для збереження додаткової інформації про сторінку. Цю інформацію використовують браузери для обробки сторінки, а пошукові системи для її індексації. У блоці <head>може бути кілька елементів <meta>, оскільки в залежності від атрибутів, що використовуються, вони несуть різну інформацію.
<nav>	Розділ документа, що містить навігаційні посилання на сайті.
<noscript>	Визначає секцію, яка не підтримує сценарій (скрипт).
<ol>	Упорядкований нумерований перелік. Нумерація може бути числова чи алфавітна.

<p>	Параграфи у тексті.
<picture>	Елемент-контейнер, який містить один елемент <img>та нуль або кілька елементів <source>. Сам собою нічого не відображає. Дає можливість браузеру вибрати найбільш відповідне зображення.
<pre>	Виводить текст без форматування, із збереженням пробілів та переносів тексту. Можна використовувати для відображення комп'ютерного коду, повідомлення електронної пошти тощо.
<s>	Відображає текст, який не є актуальним, перекресленим.
<script>	Використовується для визначення сценарію на стороні клієнта (зазвичай JavaScript). Містить або текст скрипта, або вказує зовнішній файл сценарію за допомогою атрибута src.
<section>	Визначає логічну область (розділ) сторінки, зазвичай із заголовком.
<select>	Елемент управління, що дозволяє вибрати значення із запропонованої множини. Варіанти значень містяться в <option>.
<small>	Відображає текст меншим шрифтом.
<source>	Вказує розташування та тип альтернативних медіаресурсів для елементів <picture>, <video>, <audio>.
<span>	Контейнер для малих елементів. Можна використовувати для форматування уривків тексту, наприклад, виділення кольором окремих слів.
<strong>	Розставляє акценти в тексті, виділяючи жирним.
<style>	Підключає вбудовані таблиці стилів.
<sub>	Задає підрядкове написання символів, наприклад індексу елемента в хімічних формулах.
<summary>	Створює видимий заголовок елемента <details>. Відображається із зафарбованим трикутником, клацнувши яким можна переглянути подробиці заголовка.
<sup>	Задає надрядкове написання символів.
<table>	Елемент для створення таблиці.
<tbody>	Визначає тіло таблиці.
<td>	Створює комірку таблиці.



<textarea>	Створює великі поля для введення тексту.
<tfoot>	Визначає нижній колонтитул таблиці.
<th>	Створює заголовок комірки таблиці.
<thead>	Визначає заголовок таблиці.
<time>	Визначає дату/час.
<title>	Заголовок HTML-документа, який відображається у верхній частині рядка заголовка браузера. Також може відобразитись у результатах пошуку, тому це слід брати до уваги надання назви.
<tr>	Створює рядок таблиці.
<u>	Виділяє уривок тексту накресленням, без додаткового акценту.
<ul>	Створює маркований список.
<video>	Додає на сторінку відеофайли. Підтримує 3 відео формати: MP4, WebM, Ogg.

# Термінологічний словник

Ajax – технологія звернення до сервера без перезавантаження сторінки.

API – це загальний термін у світі веб-розробки, що означає інтерфейс прикладного програмування. Коротше кажучи, щоразу, коли розробник згадує API, він шукає спосіб з'єднати дві частини програмного забезпечення для єдиної мети. Наприклад, якщо ваш веб-сайт або служба пропонує онлайн-платежі, вони повинні використовувати деякі сторонні платіжні послуги. Щоб увімкнути цю функцію, ваш веб-сайт повинен підключитися до стороннього веб-сайту, і ваша команда розробників скористається своїм API для встановлення з'єднання.

Backend - програмна частина програми, яка відповідає за обробку та збереження інформації, переданої через інтерфейс.

Backup - процес створення копії даних на носії (твердому диску), призначеному для відновлення даних в оригінальному або новому місці їх розташування у разі пошкодження або руйнування.

Bitrix - середовище розробки веб-додатків, так званий, Bitrix Framework.

Bootstrap – фреймворк для створення сучасних, кросбраузерних та стандартизованих інтерфейсів. Узгодженість шаблонів дизайну на основі HTML та CSS для типографіки, форм, кнопок, таблиць, навігації, модальних вікон, а також плагінів JavaScript.

CMS - це система управління контентом, набір скриптів для створення, редагування та управління контентом сайту. Опції CMS дозволяють налаштовувати веб-сайт відповідно до потреб замовника.

Cookie - це текстовий файл з даними (інформація про логіну та паролі, статистика відвідувань тощо), який записується в браузер сервером відвідуваного сайту.

CSS - це мова, яка відповідає за візуальне подання та структуру контенту. CSS повідомляє браузеру, як мають відобразитися елементи HTML: кольори, шрифти, блоки тощо.

DNS - це мережний сервіс, що дозволяє перетворювати символічну доменну адресу на IP-адресу і навпаки.

Favicon - це невелика іконка сайту, що відображається перед назвою сторінки у вкладці браузера, закладках, у сніпетах результатів пошукової видачі.

Footer (футер, підвал) – блок у нижній частині сторінки, нижній колонтитул.

Framework (фреймворк) - інструмент, що полегшує процес написання та запуску веб-програми.

Frontend (клієнтська частина програми) – це створення коду та розмітки, що відповідає за візуальний дизайн веб-сайту.

FTP - це сервер, який працює за File Transfer Protocol (протоколу передачі файлів)

Header (Хедер, шапка) – це блок у верхній частині сторінки сайту, який видно на всіх сторінках сайту, верхній колонтитул .

HTML – стандартна мова розмітки гіпертекстових сторінок в Інтернеті.

IP - адреса - унікальна адреса комп'ютера в мережі, необхідна для знаходження, передачі та отримання інформації від одного вузла до іншого.

Java – об'єктно-орієнтована мова програмування.

Javascript - це мова програмування, яка додає інтерактивність на веб-сайт (наприклад, відгук при натисканні кнопок або при введенні даних у форми, анімація).

Landing Page – односторінковий сайт, що зазвичай пропонує певний товар чи послугу.

MODX – це система управління контентом з відкритим кодом.

MySQL – це система управління базами даних.

PHP – це мова програмування загального призначення з відкритим вихідним кодом, що спеціально розроблена для створення веб-проектів і код може впроваджуватися безпосередньо в HTML.

URL -адреса сторінки в інтернеті, що складається з доменної адреси, шляху до сторінки на сайті та імені файлу сторінки.

WordPress – це система управління контентом, яка дозволяє власникам сайтів, редакторам, авторам керувати їх сайтами та публікувати контент без знань програмування.

XML - мова опису даних, подібна до HTML.

Бази даних - організована структура, призначена для зберігання, зміни та обробки взаємопов'язаної інформації, переважно більших обсягів.

Блог - це сайт або веб-сторінка, на якій автор веде онлайн-щоденник, основним наповненням якого є записи, що систематично додаються.

Веб-сервер - комп'ютер, який зберігає файли сайту (HTML-документи, CSS-стилі, JavaScript-файли, картинки та ін.) та надсилає їх за запитом до пристрою кінцевого користувача (веб-браузер тощо).

Верстка – процес опису візуальної частини сайту програмними кодами.

Гіперпосилання - елемент веб-сторінки, при натисканні на якому відбувається перехід на іншу адресу, яка може розташовуватися як на тому самому сайті, так і на іншому.

Дедлайн - кінцевий термін (зазвичай, дата) виконання будь-якої однієї задачі або проекту.

Динамічна веб-сторінка - веб-сторінка, що генерується або змінюється у процесі виконання запиту користувача.

Інтернет-магазин - це сервіс, розроблений для комерції в інтернеті.

Інтерфейс – набір інструментів для взаємодії людини та комп'ютерної системи.

Карта сайту - схема всіх сторінок, які є складовими веб-сайту або проекту.

Кеш - це сховище швидкого доступу до пам'яті комп'ютера, яке створюється та використовується програмою з метою прискорити свою роботу.

Кеш - своєрідне сховище елементів, яке забезпечує швидше завантаження веб-сайту для відвідувачів. Іноді, щоб побачити нові коригування та зміни потрібно його очистити.

Кросбраузерність - властивість сайту, за допомогою якої сторінки сайту однаково відображаються в різних браузерах.

Плагін - частина програмного забезпечення, яке виконує функції чи функцію і використовується для розширення можливостей веб-сайту.

Баг - помилка, яка не дозволяє веб-сайту або застосунку працювати належним чином.

Редизайн – це вдосконалення зовнішнього вигляду (оформлення) сайту. Часто позначає створення нового дизайну.

Сайт-візитка – невеликий сайт з інформацією про діяльність компанії.

Скрипт - невелика програма, що виконується на стороні сервера та написана спеціальною мовою сценаріїв або в браузері користувача

Фреймворки - призначені для створення веб-сайтів набагато швидшим процесом. Фреймворк пропонує розробникам набір інструментів та рішень для покращення веб-сайту. Приклади: Symfony, Angular, React, Laravel або Ruby on Rails.

Хостинг - послуга розміщення веб-сайту на сервері.