# 5.2. Веб-формати графічних файлів

У підготовці зображень для веб-сторінок варто бути обізнаним в популярних графічних форматах та їх властивостях.

Комп’ютерні графічні файли можна поділити на дві великі гілки: растрову і векторну.

* **Векторні файли** представляють математичний опис об'єктів відносно точки відліку координат. Наприклад, для того, щоб відобразити пряму потрібно вказати координати двох точок, які об'єднаються 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. Складність при переведенні чи перенесенні даних з одного векторного формату до іншого полягає у використанні в програмах різних алгоритмів, різних математичних формул для побудови векторних примітивів та опису растрових об'єктів.
* **Растровий файл** влаштовано простіше. Він представляє прямокутну матрицю (*bitmap*), що поділена на пікселі. Растрові формати різняться між собою здатністю містити додаткову інформацію: різні колірні моделі, вектори, Альфа-канали, прошарки різних типів, інтерліньяж (черезрядкове довантаження), 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).

Метод стиснення LZW (Lempel-Ziv-Welch) полягає у пошуку однакових послідовностей - фраз у всьому файлі. Виявлені послідовності зберігаються в таблиці, їм привласнюються короткі маркери - ключі. Так, якщо в зображенні є набори з рожевого, оранжевого і зеленого пікселів, що повторюються 50 разів, LZW виявляє цей набір, привласнює йому окреме число (наприклад, 7) і зберігає ці дані 50 разів у вигляді числа 7.

Метод LZW ефективно діє на ділянках однорідних, вільних від шуму кольорів, добре стискає довільні графічні дані, але процес кодування і розпаковування відбувається відносно повільно.

GIF дозволяє завантажувати зображення «через рядок» (Interlaced), завдяки чому, маючи лише частину файлу, можна побачити зображення цілком, але з меншою роздільністю. Це досягається за рахунок завантаження, спочатку 1, 5, 10 і далі рядків, за другим проходом підвантажуються 2, 6, 11 рядки, і згодом зображення набуває початкового вигляду. Черезрядковий запис дещо збільшує розмір файлу, але надає більшої зручності для користувачів.

У GIF можна застосовувати повністю прозорі ділянки, вони лишаються прозорими в браузерах та інших програмах і через них просвічується фоновий колір. Прозорість забезпечується за рахунок додаткового Alpha-каналу, що зберігається разом з файлом. Але, якщо ділянка є напівпрозора, то вона буде зафарбовано в колір, який обрано за замовченням.

Файл GIF спроможні містити кілька растрових картинок, які браузери відображаються 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.
* GIF підтримує покадрову зміну зображень, що робить формат популярним для створення банерів і простої анімації.
* Використовує метод стиснення без втрат.

#### 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 пікселами.

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 8х8 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.

Метод стиснення Хафмана (Huffman) розроблено в 1952 році і використовується як складова частина в ряді інших схем стиснення, в тому числі і у LZW. В методі Хафмана аналізується набір символів для визначення частоти кожного символу. Для символів, що найчастіше зустрічаються, використовується позначення у вигляді мінімальної можливої кількості бітів. Наприклад, найчастіше в англійських текстах зустрічається буква "е". Використовуючи кодування Хафмана можна представити літеру "е" лише двома бітами (1 і 0), замість вісьмох бітів, необхідних для представлення букви "е" в кодуванні ASCII.

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 ВМР.

Формат JPEG є апаратно незалежним, повністю підтримується на РС і Macintosh. JPEG відтворює спектр кольорів TrueColor (224).

JPEG краще стискає растрові картинки 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 8х8 пікселів переходи виходять м'якшими, за рахунок того, що таких квадратів є більше.

Даний формат називають стисненням з втратами, оскільки алгоритм JPEG вибірково відкидає дані. Не бажано зберігати у JPEG-форматі будь-які зображення, де важливими є тонкі нюанси кольорів, оскільки під час стиснення відбувається відкидання колірної інформації. У JPEG слід зберігати лише кінцевий варіант роботи, оскільки кожне повторне збереження призводить до нових втрат (відкидання) даних і початкове зображення може бути вкрай зіпсованим.

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)

<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 браузерах.

## PNG-8

PNG-8 — формат подібний до GIF і має покращений формат стиснення даних.

#### Особливості

* Використовує 8-бітову палітру (256 кольорів) в зображенні, за що і отримав в своїй назві цифру вісім. При цьому можна вибирати, скільки кольорів буде задіяно у файлі — від 2 до 256.
* На відміну від GIF, не відображає анімацію.

#### Область застосування

Текст, логотипи, ілюстрації з чіткими краями, зображення з градієнтною прозорістю.

## PNG-24

PNG-24 — формат, аналогічний до PNG-8, але використовує 24-бітову палітру кольору Подібно до формату JPEG, зберігає яскравість і відтінки кольорів у фотографіях. Подібно до форматів GIF і PNG-8, зберігає деталі зображення та прозорість.

#### Особливості

* Використовує понад 16 млн. кольорів, тому, застосовується для повноколірних 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.

## 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.

**Особливості APNG:**

* Основною характеристикою APNG є здатність відтворення анімації. На відміну від статичних зображень PNG, APNG може містити кілька кадрів, які можна відтворювати як послідовність, створюючи ефект анімації.
* APNG підтримує альфа-канал, що робить його ідеальним для зображень із прозорим тлом або шарами.
* Багато сучасних браузерів підтримують APNG, що дозволяє створювати анімовані зображення, які можуть бути відтворені у веб-середовищі без необхідності використовувати додаткові плагіни або інструменти.

**Можливі проблеми APNG:**

* Деякі застарілі браузери та програми можуть не коректно відтворювати анімації APNG. Це може спричинити проблеми сумісності під час перегляду зображень на різних пристроях.
* APNG-зображення можуть бути досить великими, особливо якщо вони містять багато кадрів і складну анімацію. Це може вплинути на швидкість завантаження веб-сторінки, особливо при використанні великих файлів APNG.
* У деяких випадках анімацію можна створити більш ефективно за допомогою інших форматів, таких як GIF чи WebP. Тому, перш ніж використовувати APNG, варто оцінити, чи він підходить для конкретного завдання.
* APNG надає обмежені можливості керування анімацією (наприклад, швидкістю відтворення). Деякі інші формати, такі як GIF, можуть надавати більші можливості для налаштування анімації.

В цілому, APNG є корисним форматом для створення анімованих зображень за допомогою прозорості та сумісності з більшістю сучасних браузерів. Однак, при його використанні слід враховувати потенційні проблеми сумісності та розмір файлів, особливо під час створення веб-контенту.

## WebP_LogoWebP

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

WebP – це сучасний формат стиснення зображень, розроблений Google. Він є форматом зображень із втратою даних, який покликаний забезпечити високу якість стиснення при порівняно низькому розмірі файлів.

**Основні характеристики WebP:**

* WebP використовує стиснення з втратою даних, що означає, що частина інформації видаляється із зображення для скорочення розміру файлу. Однак якість зображення залишається високою, і втрата візуальних даних є мінімальною.
* WebP забезпечує більш ефективне стиснення зображень порівняно з іншими форматами, такими як JPEG. Це зменшує час завантаження веб-сторінок та заощаджує пропускну здатність.
* WebP підтримує прозорість (альфа-канал), що дозволяє створювати зображення з напівпрозорими або прозорими областями.
* WebP може використовуватися для створення анімованих зображень, подібних до GIF або формату APNG.
* WebP підтримує різні колірні профілі, що дозволяє зберігати зображення з різною колірною моделлю.
* Багато сучасних браузерів, включаючи Google Chrome, Mozilla Firefox, Microsoft Edge та інші, підтримують формат WebP.

WebP є універсальним і ефективним форматом зображень для веб-розробки, який забезпечує високу якість стиснення і добре підтримується сучасними браузерами.

## FLIF Free Lossless формат зображення

<http://flif.info/>

FLIF (Free Lossless Image Format) – це сучасний формат стиснення зображень, розроблений для забезпечення високої якості стиснення без втрати даних.

**Особливості FLIF:**

* Основною характеристикою формату FLIF є можливість стиснення зображень без втрати даних. Якість зображення зберігається у вихідній формі, і під час стиснення не відбувається жодної деградації.
* FLIF використовує адаптивні методи стиснення, що дозволяє досягати високого ступеня стиснення для різних типів зображень, включаючи фотографії, малюнки та скріншоти.
* FLIF підтримує багаторівневий стиск, що означає, що можна зберігати зображення з різними рівнями якості та роздільності.
* FLIF підтримує альфа-канал, що дозволяє створювати зображення з прозорістю.
* Формат FLIF підтримує колірні профілі, що дозволяє зберігати точність кольору.

**Проблеми FLIF:**

* Обмежена підтримка браузерами. FLIF не так широко підтримується, як інші формати зображень, такі як JPEG та PNG.
* Для використання FLIF-зображень у веб-середовищі може знадобитися їх попередня конвертація в інші формати, такі як JPEG або WebP, щоб забезпечити сумісність із браузерами.
* FLIF-зображення можуть мати більший розмір у порівнянні з іншими форматами, такими як JPEG. Це може вплинути на швидкість завантаження веб-сторінок.
* Низька поінформованість про формат FLIF серед розробників та дизайнерів у порівнянні з більш відомими форматами стиснення зображень.

FLIF - це цікавий формат стиснення зображень, що обіцяє високу якість стиснення без втрат даних. Однак його обмежена підтримка браузерами та необхідність конвертації можуть створити проблеми під час використання на веб-сайтах та інших проектах.

# svgВекторні 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 браузери. Формат є відкритим стандартом, на відміну від більшості інших форматів, SVG не є чиєюсь 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 зазвичай важать менше за растрові зображення.
* Формат масштабується, що забезпечує чіткість за любої 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.

Занадто складні SVG-зображення (з втіленими разтровими елементами) збільшують розмір файлу. SVG не застосовний до фотографій, тут краще використати формати JPG і webP.

## Інші формати графічних файлів

### PSD (Adobe Photoshop Document)

Внутрішній формат популярного растрового редактора Photoshop останнім часом підтримується великою кількістю програм. Він дозволяє записувати зображення з багатьма прошарками, їх масками, додатковими Альфа-каналами і каналами базових кольорів, контурами та іншою інформацією.

Для стиснення застосовують метод RLE (Run Length Encoding), кодування із змінною довжиною рядка. Дія методу полягає в пошуку однакових пікселів в одному рядку. Якщо в рядку, наприклад, є 3 піксела білого кольору, 21 - чорного, 14 - білого, то застосування RLE надає можливість не запам'ятовувати кожен з них (38 пікселів), а записати як 3 білих, 21 чорний і 14 білих в першому рядку.

Подібно до методу LZW, алгоритм RLE добре працює з штучними і ілюстрованими картинками і гірше з фотографіями. У випадку, якщо фотографія має багато дрібних деталей, RLE може навіть збільшити розмір файлу.

### TIFF (Tagged Image File Format)

Апаратно незалежний формат TIFF, на сьогоднішній, день є одним з найпоширеніших і надійніших, його підтримують практично всі програми на РС і Macintosh, що пов’язані з графікою. TIFF є кращим вибором при імпорті растрової графіки у векторні програми та видавничі системи. Йому доступно весь діапазон колірних моделей від монохромної до RGB, CMYK і додаткових кольорів Pantone. TIFF може зберігати контури, Альфа-канали та інші додаткові дані.

TIFF має два різновиди: для Macintosh і РС. Це пов'язано з тим, що процесори Motorola читають і записують числа зліва направо, а процесори Intel - навпаки. Сучасні програми можуть без проблем використовувати обидва варіанти формату. Зазвичай, дані у форматі TIFF не стискаються, але може бути використано LZW-стиснення.

### BMP (Windows Device Independent Bitmap)

Рідний формат Windows, який підтримується всіма графічними редакторами, що працюють під управлінням цієї операційної системи. Застосовується, в основному, для збереження растрових зображень, що призначені для використання в Windows. Може зберігати як індексовані (до 256 кольорів), так і RGB-кольори (понад 16 млн. відтінків). Можливе застосування стиснення за алгоритмом RLE.

### Adobe_PDFPDF (Portable Document Format)

PDF запропоновано фірмою Adobe, як платформо-незалежний формат для створення електронної документації, презентацій, передачі верстки та графіки через мережі.

Односторінкові файли PDF відмінної якості може створювати Photoshop. Багатосторінкові PDF можуть створювати програми Adobe Acrobat, PageMaker і програми пакету MS Office.

PDF спочатку проектувався як компактний формат для електронної документації. Тому, всі дані в ньому можуть стискатися, причому, до різного типу інформації застосовуються різні, найбільш прийнятні для них типи стиснення: JPEG, RLE, CCITT, ZIP. Програма Acrobat дозволяє розставляти гіперпосилання, заповнені поля, додавати у файл PDF відео чи звук, інші дії.

Файл PDF може бути оптимізованим. З нього видаляються елементи, що повторюються, встановлюється посторінковий порядок завантаження сторінок через Веб, з пріоритетом спочатку для тексту, потім для графіки. Якщо елементів, що повторюються, немає, файл, після оптимізації, як правило, дещо збільшується.

PDF найчастіше використовується для передачі по мережах графіки і зверстаного тексту в компактному виді. Він може зберігати всю інформацію для пристрою виведення, яка була в початковому файлі.

### WMF (Windows Metafile)

Векторний формат WMF є рідним форматом Windows і використовує його графічну мову. Призначений для передачі векторних даних через буфер обміну (Clipboard). Розпізнається практично всіма програмами Windows, так чи інакше пов'язаними з векторною графікою. Користуватися форматом WMF варто лише у випадках передачі «чистих» векторів. WMF спотворює колір, не зберігає певні параметри, які привласнюються об'єктам в різних векторних редакторах, не містить растрові об'єкти, не розпізнається багатьма програмами на Macintosh.

### aiAI (Adobe Illustrator Document)

Adobe Illustrator – популярний графічний редактор від Adobe. Векторний формат Illustrator можна безпосередньо відкрити у Photoshop, його підтримують майже всі програми Macintosh і Windows так чи інакше пов'язані з векторною графікою і графікою взагалі.

Формат Illustrator є найкращим посередником при передачі векторів з однієї програми в іншу, з РС на Macintosh і навпаки. Втілені або пов'язані з документом растрові файли при обміні через формат Illustrator втрачаються.

### CRD (CorelDraw Document)

Векторний формат, що має лідерство на платформі РС. Багато програм на РС (FreeHand, Illustrator, PageMaker тощо) можуть імпортувати файли CorelDraw.

У файлах застосовується окрема компресія для векторів і растрів, можуть втілюватися шрифти, файли CorelDraw мають величезне робоче поле 45х45 метрів (цей параметр є важливим для зовнішньої реклами), підтримується багатосторінковість.

# Зображення, що створюються програмним шляхом

Є чотири способи створення зображення на веб-сторінці програмним кодом: CSS, SVG, Canvas, JS-анімація.

## CSS

### cssСтворення зображення на чистому CSS

Замість програмного забезпечення для створення векторних ілюстрацій (такого, як Illustrator, Affinity Designer або Sketch) використовується безпосередньо CSS-код. Створення зображень на CSS може вплинути на креативність мислення і надихнути на підкорення нових вершин цієї мови.

#### Приклади CSS-зображень:

* <https://a.singlediv.com/>
* <https://habrahabr.ru/post/323086/>

### Анімація за домомогою CSS

Властивість CSS - animation дозволяє анімувати переходи від однієї конфігурації стилю CSS до іншої. Анімація складається з двох компонентів: стилю, що описує анімацію і набору ключових кадрів, які вказують початковий і кінцевий стани елементу, а також можливі проміжні точки.

Існують три ключові переваги використання CSS - анімації перед традиційними методами анімації, керованими скриптами:

* Вони прості у використанні для простих анімацій, їх можна створювати не знаючи JavaScript.
* Анімація працює добре, навіть при помірному завантаженні системи. Механізм рендерингу може використовувати пропуски кадрів і інші методи, щоб підтримувати оптимальну продуктивність.
* Надання браузеру управління послідовністю анімації дозволяє йому оптимізувати продуктивність, наприклад, зменшуючи частоту оновлення анімацій, запущених на вкладках, які в даний час не відображаються.

## svgЗображення SVG

SVG - це формат масштабованої векторної графіки. В векторних форматах зберігається не саме зображення, а інструкція з його побудови по точках і кривих.

В растрових форматах інформація про конкретне число точок зображення щільно упакована в бінарний код. В нього марно заглядати і міняти його можна лише в редакторах.

Формат SVG також можна створювати і змінювати в редакторах, наприклад, Illustrator, Sketch або Inkscape. Але ще він текстовий, а значить його можна відкрити в любому текстовому редакторі.

<svg width = "20">

 <rect fill="#fc0" width="20" height="20" />

 <line stroke="black" x1="0" y1="0" x2="20" y2="20" />

</svg>

При втіленні SVG-файлу, насправді, уставляється не просто картинка, а ціла сторінка. Зі своєю системою координат, стилями, скриптами і особливостями. Існує чотири основних способів втілення SVG-зображень зі своїми особливостями.

### Уставляння в елементі img HTML-коду

Це самий ефективний спосіб завантажити будь-яку картинку.

<img src="picture.svg" alt="За склом">

В такому SVG-зображенні не працюватимуть скрипти і будь-які спроби взаємодії з елементами всередині не є успішними. Файл буде як за склом: дивитися можна, а чіпати не можна. Хоча всередині все інше прекрасно працює, включаючи CSS-анімації.

Такий спосіб найкраще підходить для контентних зображень, яким не потрібна взаємодія: логотипи, графіки, схеми.

### Фонова картинка в CSS

Уставляння основною картинкою до елементу, псевдоелементу або контенту. В такому SVG-зображенні також не можна змінювати оригінальний вміст.

.picture {background-image:url (picture.svg);}

Цей спосіб підходить для оздоблювальної графіки, для якої не потрібна взаємодія: фони, іконки та інші нескладні зображення.

### Уставляння через тег <object>

Третій спосіб, уставляння через тег object. У цьому випадку працюють скрипти, взаємодія, анімація - якщо вони описані всередині SVG. Між тегами object можна вставити код, що повідомить користувача про використання застарілої версії браузера.

<object type = "image/svg + xml" data="picture.svg">

 <img src = "picture.png">

</object>

Замість тегу object можна використовувати тег iframe, що буде сприймати SVG-код як іншу сторінку. Але object працює краще і підлаштовується під розміри картинки.

За гнучкість доводиться платити: через те, що це вже не просто графіка і там можна додавати скрипти, до такого способу висуваються інші вимоги безпеки. Наприклад, картинку з іншого домену просто так вже не вставити.

Цей спосіб підходить, коли потрібно додати інтерактивну графіку: ігри, графіки, діаграми.

### Пряме уставляння вмісту SVG-файлів на сторінку

Цей спосіб став можливим при впровадженні стандартів HTML5 і вміст SVG-файлів тепер можна вставляти прямо на сторінку, як будь-які інші теги.

<h1>Квадрат</h1>

<svg>

 <rect fill="#fc0" width="120" height="120" />

</svg>

З таким SVG-елементом можна маніпулювати як зі звичайними HTML-елементами: стилі, скрипти. Наприклад, можна змінювати колір заповнення при наведенні і описувати все в загальних стилях.

<style>

rect:hover {fill:#090;}

</style>

Нажаль, такі зображення не можна зберігати.

Формат SVG є значно більшим, ніж графічний формат, і є достатньо матеріалів, які розкривають його можливості

* Курс «Знайомство з SVG» — <https://htmlacademy.ru/courses/130>
* Курс «Оздоблення SVG-фігур» — <https://htmlacademy.ru/courses/187>

## Canvas

HTML елемент <canvas> створює область на веб-сторінці, яку можна використати для відтворення графіки в режимі реального часу за допомогою сценаріїв (наприклад, за допомогою мови JavaScript). Він може бути використаний для створення онлайн-ігор, годинників, анімації тощо.

### Додавання на сторінку

Елемент <canvas> є звичайним HTML-елементом:

<canvas id="test" width="600" height="200"> </canvas>

Цей елемент має три атрибути: id, width і height. Атрибут id (ідентифікатор) надає елементу унікальне ім'я - test, це потрібно для того, щоб мова сценаріїв могла отримати до нього доступ. Атрибут width визначає кількість пікселів по горизонталі, яке <canvas> буде займати на веб-сторінці. Аналогічним чином height визначає вертикальну займану область. Таким чином, браузер зможе виділити простір потрібного розміру на сторінці, відповідно до зазначених значень.

За замовчуванням полотно є прозорим і не має рамки, тобто є порожнім простором у вікні браузера, в якому можна малювати.

### Використання CSS для вказування розміру

За замовчуванням елемент canvas має ширину 300 пікселів і висоту 150 пікселів. Якщо не поставити атрибути width і height, то елемент отримає розміри за замовчуванням. Якщо потім задати розміри, використовуючи CSS, припустимо, 600х200 пікселів, то <canvas> розміром 300х150 пікселів піддасться масштабування відповідно до заданих параметрів, так само і все, що буде в ньому намальовано.

При використанні атрибутів width і height можна задавати розміри для елемента, які будуть більше (або менше), ніж 300х150 пікселів, і все створене в ньому буде відображатися нормально. Таким чином, використовувати CSS для налаштування розміру буде доцільно, якщо потрібно дійсно масштабувати елемент.

### Робота зі старими версіями браузерів

Не всі браузери підтримують елемент <canvas>, тому, для інформування користувачів, чиї браузери не підтримують його, можна помістити спеціальне повідомлення або альтернативний вміст, який буде відображатися тільки в тому випадку, якщо елемент не підтримується, в іншому випадку проігнорується:

<! - Розміщуємо інформацію для користувачів ->

<canvas>

 Ваш браузер не підтримує елемент <canvas>

</canvas>

<! - або можна помістити картинку рівну розміру елемента ->

<canvas>

 <img src="image1.png">

</canvas>

### Атрибути

* **height:** Визначає висоту елемента в пікселах (наприклад: "400"). Значення за замовчуванням - "150".
* **width:** Визначає ширину елемента в пікселfх (наприклад: "150"). Значення за замовчуванням - "300".

<canvas id="myCanvas" width="400" height="150"></canvas>

<script type="text/javascript">

 var canvas=document.getElementById('myCanvas');

 var ctx=canvas.getContext('2d');

 ctx.fillStyle='#CC5200';

 ctx.fillRect(0,0,100,100);

</script>

Елемент <canvas> підтримує [Глобальні атрибути](https://puzzleweb.ru/html/globalattributes.php) и [Події](https://puzzleweb.ru/html/eventattributes.php)

## Контрольні питання

1. Назвати основні типи комп’ютерної графіки.
2. Які графічні формати використовуються у веб? В чому їх відмінності та особливості?
3. Які відмінності між растровими та векторними графічними форматами.
4. Перелічити переваги формату JPG?
5. Перелічити переваги формату WebP?
6. Перелічити переваги формату SVG?
7. Які особливості притаманні векторним форматам зображень?
8. Які формати застосовують для зображень з прозорим фоном? Їх особливості?
9. Перелічити можливі підходи для створення анімаційного зображення.
10. Для чого призначений HTML елемент <canvas>?