# 4.2. 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**.

Селектор повідомляє браузеру, який саме елемент форматувати, в блоці оголошення перелічено властивості форматування та їх значення.



Рис. 2.3. 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 |
| <h1 class="headline">Інструкція користування персональним комп'ютером</h1><p class="headline">Примітка. Це важливо для роботи</p> | headline { text-transform: uppercase; color: lightblue;} |

**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 Нащадку.** 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 відноситься 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 форматувати 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 невідвідуваного посилання.

**a: visited** - описує стиль вже відвіданого посилання.

**a: hover** - описує вигляд елементу, над яким проходить вказівник мишки.

**a: focus** - описує вигляд елементу, над яким знаходиться (сфокусований) вказівник.

**a: active** - описує вигляд елементу, який активовано користувачем.

**Структурні псевдокласи.** Структурні псевдокласи форматують 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 елементу.

**Псевдоелементи.** Псевдоелементи не є елементами сторінки, їх використовують для додавання вмісту, який генерується за допомогою властивості content, і для зміни зовнішнього вигляду частини елементу:

**: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 елементу. 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 елементу і псевдоелемента додається 0, 0, 0, 1;
* для стилю, який доданого безпосередньо до елементу - 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 елементу 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.

## Контрольні питання

1. Для чого застосовують каскадні таблиці стилів?
2. З чого складається правило створення стилю
3. Яким чином можна додати стилі до html-сторінки?
4. Перелічити різні типи селекторів.
5. Які комбінації селекторів можна застосувати?
6. Яким чином визначають пріоритет стилю?

## Використані джерела

1. Сучасний підручник CSS - <https://idg.net.ua/blog/uchebnik-css>
2. Селектори CSS. Види, групування і специфічність - <https://itchief.ru/html-and-css/selectors>
3. Псевдо-класи і псевдо-елементи в CSS - [https://coderoad.ru/8069973/В-чем-разница-между-псевдоклассом-и-псевдо-элементом-в-CSS](https://coderoad.ru/8069973/%D0%92-%D1%87%D0%B5%D0%BC-%D1%80%D0%B0%D0%B7%D0%BD%D0%B8%D1%86%D0%B0-%D0%BC%D0%B5%D0%B6%D0%B4%D1%83-%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%BC-%D0%B8-%D0%BF%D1%81%D0%B5%D0%B2%D0%B4%D0%BE-%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%BC-%D0%B2-CSS)
4. CSS по методології БЕМ - <https://ru.bem.info/methodology/css/>
5. Специфічність CSS - <https://developer.mozilla.org/ru/docs/Web/CSS/Specificity>
6. Каскадність CSS. Пріоритети стилів - <https://idg.net.ua/blog/uchebnik-css/azy-css/kaskadnost>