

# Каскадні таблиці стилів

✘ CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних.

CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript[1].

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці вебсторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

CSS використовується авторами та відвідувачами вебсторінок, щоб визначити кольори, шрифти, верстку та інші аспекти вигляду сторінки. Одна з головних переваг — можливість розділити зміст сторінки (або контент, наповнення, зазвичай HTML, XML або подібна мова розмітки) від вигляду документу (що описується в CSS).

## Переваги

- Інформація про стиль для усього сайту або його частин може міститися в одному .css-файлі, що дозволяє швидко робити зміни в дизайні та презентації сторінок;
- Різна інформація про стилі для різних типів користувачів: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на принтер, стиль для мобільних пристроїв;
- Сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з урахуванням їх;
- Прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузері здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

## Синтаксис

CSS має порівняно простий синтаксис і використовує небагато англійських слів для найменування різних складових стилю.

Стили складаються зі списку правил. Кожне правило має один або більше селектор (англ. selector) та блок визначення (англ. declaration block). Блок визначення складається з оточеного фігурними дужками списку властивостей.

Властивості в списку оформлюються у вигляді назва властивості, двокрапка (:), значення, крапка з комою (;). Приклад:

```
p {
  font-family: Verdana, sans-serif;
}
h2 {
  font-size: 110%;
  color: red;
  background: white;
}
.note {
  color: red;
  background: yellow;
  font-weight: bold;
}
p.warning {
  background: url(warning.png) no-repeat fixed top;
}
#paragraph1 {
  margin: 0;
}
a:hover {
  text-decoration: none;
}
#news p {
  color: red;
}
```

В прикладі використано 7 правил, селектори **p**, **h2**, **.note**, **p.warning**, **#paragraph1**, **a:hover** та **#news p**. Приклад властивості: `color: red`, де властивості з назвою `color` присвоєно значення `red`.

**В перших двох правилах** визначаються властивості HTML-елементів `p` (абзац, скорочення від англ. paragraph) та `h2` (Заголовок другого рівня, скорочення від англ. header). Абзац буде відображено шрифтом Verdana або, у разі відсутності Verdana, іншим шрифтом sans-serif. Заголовок другого рівня буде відображено червоним кольором на білому тлі.

Третє правило буде застосовано до всіх елементів, властивість `class` визначена як `'note'`. Наприклад:

```
<p class="note">Абзац буде відображено червоним жирним шрифтом на жовтому тлі.</p>
```

**Третє правило** також можна записати як `*.note`. Щоб третє правило застосовувалося лише до абзаців, його слід переписати як:

```
p.note {  
  color: red;  
  background: yellow;  
  font-weight: bold;  
}
```

**Четверте правило** стосується тих елементів рівня абзацу (p), що мають властивість class рівну 'warning'. Такі абзаци матимуть фонове зображення warning.png згори.

Властивість .class може мати не лише порівняння, але й перелік. Коли властивість class містить перелік значень, розділених пробілом, то до цього елемента застосовуються правила усіх перелічених класів. Наприклад, до <p class="note warning"> будуть застосовані правила note та warning.

**П'яте правило** застосовується лише до одного елемента в документі HTML, що має ідентифікатор id рівний paragraph1. Цей елемент не буде мати межі навколо, тому що властивість margin дорівнює 0. Наприклад:

```
<p id="paragraph1">Цей абзац не має межі тому що межа  
дорівнює 0. На сторінці може бути лише один такий елемент.  
</p>
```

**Шосте правило** визначає стиль наведення мишкою (hover) для a (anchor) елементів. Стандартно більшість браузерів підкреслюють елементи a (посилання в межах одного документа або на інші HTML-документи). Це правило прибирає підкреслювання з посилань, коли користувач наводить на них мишкою.

**Останнє правило** застосовується до тих абзаців, що знаходяться всередині (мають батьківський елемент) з ідентифікатором news. Це приклад наслідування властивостей.

Завдяки каскадній структурі CSS, ці абзаци (p) будуть наслідувати font-family: Verdana, sans-serif; як і всі елементи p (абзаци). Ті з них, що мають class рівний warning будуть мати також і фонове зображення.

CSS може також містити коментарі. Синтаксис коментарів подібний до синтаксису, що використовується в багатьох мовах програмування (наприклад C, PHP). Слід звернути увагу на те, щоб в коментарях CSS-файлів не використовувалися кириличні літери (деякі браузери некоректно обробляють CSS-файли з українськими літерами).

```
/* comment */
```

## Селектори та Псевдо-класи

CSS має спеціальні позначення (селектори, від англ. selector) для визначення діапазону застосування правила. **Всі елементи**

використовується селектор \* (часто не пишеться, наприклад \* p та p — рівнозначні) **За назвою елемента**

наприклад для всіх p або h2 **Наслідування**

наприклад для всіх a елементів які містяться в li елементах (посилання всередині списків),

селектор матиме вигляд li а **Властивості class та id**

наприклад .class або #id для елементів class="class" або id="id" **Суміжні елементи**

наприклад для всіх елементів p, попереднім до яких є елемент h2 (перший абзац після заголовка), селектор буде h2 + p **Пряме наслідування**

наприклад для всіх span елементів всередині p, але не глибших за перший рівень наслідування

span, селектор буде p > span **За атрибутами**

наприклад для всіх <input type="text"> селектор буде input[type="text"]

наприклад для всіх <input type="text"> селектор буде input[type="text"]

Для точнішого визначення стилю можуть використовуватися псевдо-класи (англ. pseudo-classes). Мабуть найвідомішим є псевдо-клас :hover, що застосовується коли користувач вкаже на елемент, зазвичай наведенням курсору. Стиль визначається для a:hover або #elementid:hover. Інші псевдо-класи, наприклад, :first-line (перший рядок тексту), :first-letter (перша літера) :visited (посилання, що вже відкривалось) або :before (визначає стиль перед елементом). Спеціальний псевдо-клас :lang© буде застосовано до елементів якщо їхня мова (англ. language) «с».

Підтримка псевдо-класів залежить від браузера, наприклад Internet Explorer 6 підтримує псевдо-клас :hover лише для посилань.

Селектори можуть поєднуватися й іншим чином для досягнення більшої гнучкості (див. визначення селекторів на сайті W3C [Архівовано 23 квітня 2006 у Wayback Machine.]). В наступному прикладі кілька селекторів об'єднуються через кому[3]. Правило встановлює шрифт для HTML-заголовків всіх рівнів.

## Джерела

-  CSS

From:  
<https://library.vpuhluhiv.com.ua/> - **Wiki Глухівського ВПУ**

Permanent link:  
<https://library.vpuhluhiv.com.ua/subjects:basic:informatika:web:css?rev=1666306113>

Last update: **21.10.2022 01:48**

