**CSS**

**Каскадування стилів CSS**

Каскадування стилів CSS (Cascading Style Sheets) - це процес визначення стилів елементів HTML, який використовується для зміни вигляду веб-сторінок. Каскадування стилів дозволяє відокремити опис вигляду сторінки від її змісту, що дозволяє робити зміни відображення сторінки без зміни її структури.

CSS використовує каскадні таблиці стилів для визначення, які стилі будуть застосовані до конкретного елементу на сторінці. Це означає, що якщо багато стилів застосовуються до одного елементу, то будуть застосовані тільки ті, які мають вищий пріоритет або більш специфічні селектори.

Пріоритет стилів визначається наступним чином:

1) Inline стилі (задані в атрибуті style елементу) мають найвищий пріоритет і будуть застосовані першими.

2) Селектори ID мають вищий пріоритет, ніж селектори класу або тегу.

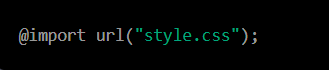
3) Селектори, що використовуються більш специфічно (наприклад, селектор, який має ID і клас), мають вищий пріоритет, ніж більш загальні селектори.

4) Стилі, які були визначені пізніше, мають вищий пріоритет, ніж ті, що були визначені раніше.

У разі конфлікту двох або більше стилів, що мають однаковий пріоритет, буде застосований останній визначений стиль. Таким чином, каскадування стилів CSS дозволяє досить гнучко визначати вигляд сторінки та швидко змінювати його за потреби.

**@import**@import є директивою мови CSS, яка дозволяє включити зовнішні файли CSS в інший файл CSS. Це дозволяє вам зручно організовувати стилі вашого веб-сайту та уникнути повторення коду.

Синтаксис @import такий:



Це означає, що файл style.css буде імпортований у поточний файл CSS. Шлях до файлу може бути відносним або абсолютним.

Директива @import повинна бути поміщена на початок файлу CSS, перед будь-яким іншим кодом. Якщо ви включаєте декілька зовнішніх файлів CSS за допомогою @import, вони будуть завантажуватися послідовно в порядку вказання.

Важливо зауважити, що імпортування зовнішніх файлів CSS за допомогою @import може призвести до збільшення часу завантаження сторінки, тому краще обмежуватися тільки тими стилями, які дійсно потрібні на сторінці.

**Adaptive vs Responsive веб-дизайн**

Adaptive – сайт, який на моніторі, планшеті, смартфоні, ... відображається по-різному. Змінюється розміщення блоків, розміри кнопок, полів, текстів. Для чого? Щоб користувачу було комфортніше користуватись сайтом. Особливість: сайт має кілька варіантів відображення, залежно від ширини екрану або типу дивайса.

Responsive – різновид адаптивного дизайну, який теж має кілька варіантів відображення, але переходи між ними плавні (резинові).

При адаптивній верстці можливе використання різного HTML/CSS коду для різних розмірів екрану або дивайсів. При респонсів – один HTML/CSS, але модифікація сайту досягається шляхом CSS маніпуляцій.

**Browser and CSS**

**Graceful degradation** - можливість роботи сайту при відключеному JS; достатньо акуратне відображення інтерфейсу в браузері, який не підтримує нові властивості CSS; адекватне відображення сайту при відключених зображеннях. У всіх цих випадках робота користувача з інтерфейсом має бути можлива, хоча і не така зручна.

**Progressive enhancement** говорить про те, що веб-інтерфейси повинні створюватись поетапно, від простого до складного. На кожному етапі має виходити завершений вен-інтерфейс, який буде кращий, зручніший, ніж попередній. Зараз таких етапів 4:

* Логічна розмітка. Технологія: HTML.
* Зовнішнє представлення. Технологія: CSS.
* Бездоганне зовнішнє представлення. Технологія: CSS.

Взаємодія, інтерактивність, зручність. Технологія: JavaScript

**Медіа запити (Media queries)**

Медіа квері бажано додавати окремим файлом останнім серед css файлів або вкінці документа, після решти стилей.

Якщо додавати окремим файлом:

***<link rel="stylesheet" media="screen and (max-width: 600px)" href="small.css" />***

***<link rel="stylesheet" media="only screen and (-webkit-min-device-pixel-ratio: 2)" type="text/css" href="iphone4.css" />***

***<link rel="stylesheet" media="all and (orientation:portrait)" href="portrait.css">***

***<link rel="stylesheet" media="all and (orientation:landscape)" href="landscape.css">***

Media Queries правила:

***@media screen***

***@media print***

***@media tv***

***@media all***

***and***

***(max-width: 600px)***

***(min-width: 900px)***

***and (min-width: 600px) and (max-width: 900px)***

***(max-device-width: 480px)***

***(-webkit-min-device-pixel-ratio: 2)***

***(orientation:portrait)***

***(orientation:landscape) {***

***.class {***

***background: #000;***

***}***

***}***

По замовчуванню Safari стискає сторінки, щоб помістити їх на екрані.

Наступний мета тег говорить Сафарі використовувати ширину дивайса як ширину вікна.

***<meta name="viewport" content="width=device-width; initial-scale=1.0">***

***Приклади:*** [***https://mediaqueri.es/***](https://mediaqueri.es/)

**CSS методології:.БЕМ**

БЕМ = Блок-Елемент-Модифікатор

Принципи:

* особливе іменування класів
* незалежність блоків
* повторне використання існуючого коду
* точкові зміни з мінімальними затратами

Блок – частина сторінки, яка являється логічно незалежною від іншого наповнення (меню)

Елемент – частина блока, яка відповідає за окрему функцію. Вона може знаходитись тільки в складі блока і не має сенсу у відриві від нього (кожна кнопка меню).

Модифікатор – властивість блока або елемента, що відповідає за його зовнішнє відображення або поведінку (активна кнопка меню)

Система іменування:

- назва блока формується як префікс-назва-блока

.b-menu { … }

.b-text-input { … }

- назва елемента створюється по схемі: префікс-назва-блока\_\_назва-елемента

.b-menu\_\_item { … }

.b-text-input\_\_label { … }

-назва модифікатора елемента:

.b-menu\_disabled { … }

-назва модифікатора елемента

.b-menu\_\_item\_state\_current { … }

Плюси:

* достатньо одного класа
* специфічність CSS правил
* абсолютно незалежні блоки

Мінуси:

* довгі назви класів
* складність освоєння командою

***<ul class="ui-list">***

***<li class="ui-list\_\_item">***

***<div class="media">***

***<img src="" alt="" class="media\_\_img" />***

***<div class="media\_\_body">***

***<p>...</p>***

***<ul class="bare-list">***

***<li>...</li>***

***</ul>***

***</div>***

***</div>***

***</li>***

***...***

***</ul>***