

Cascade Style Sheets

- Основні типи значень у CSS
- Спеціалізовані директиви CSS
- Властивості керування потоком документу, заданням розмірів елементів

Основні типи значень у CSS

Важливим є розуміння, які значення можуть набувати властивості у селекторах. Загалом їх ділять на такі основні групи: абсолютні і відносні числові одиниці; ключові слова; кольори; функції та рядки

Абсолютні величини задають точно значення властивостей, для яких вони є характері, у пікселях

height: 100px;

висота елемента - 100 пікселів

Відносні одиниці вимірювання дозволяють вказати значення розмірів по відношенню до батьківського елемента. До них відносять: відсотки; *vw* – розраховується як одна сота від ширини екрана; *vh* – аналогічна до попередньої відносна одиниця, тільки у цьому випадку за основу береться висота екрана; *em* – одиниця вимірювання для шрифтів, яка задає розмір тексту по відношенню до розміру шрифту батьківського елемента; *rem* – працює аналогічно, але розрахунок відбувається на основі базового розміру шрифту, який встановлений для тегу `body`

Кольори у CSS

✓ за допомогою ключових слів, які позначають колір (наприклад **color:red;** – задає червоний колір тексту);

✓ у вигляді шістнадцяткового значення (**color:#23aaff;**);

✓ за допомогою ключового слова **rgb(255,40,50),**

дужках подаються значення трьох кольорів (червоного, зеленого і синього) за допомогою числа із інтервалу від 0 до 255 включно.

Спеціалізовані директиви CSS

Вони являються спеціалізованими правилами, які дозволяють змінити відображення або ж поведінку елементів сторінки у певній, завчасно визначеній ситуації. Директиви розпочинаються із запису символу **@**, після якого записується одне із ключових слів. Їх ділять на стандартні правила та вкладені. Коротко охарактеризуємо перші.

@charset

задає кодування, яке у подальшому буде використовуватися браузером. Наприклад правило `@charset "UTF-8";` вказує, що сторінка використовує кодування символів UTF-8.

@import

дозволяє завантажити та використовувати зовнішній CSS файл (`@import "new.css";`).

@namespace

дозволяє застосовувати CSS для таких мов розмітки як XML та XHTML.

@document

спеціалізована директива, яка визначає умови застосування стилей для всієї сторінки

```
@document (
  url(https://sample.com/)
  regexp("https:// ..."))
{body{font-family: Arial;}}
```

Спеціалізовані директиви CSS

@font-face

дозволяє завантажувати користувацькі шрифти

```
@font-face{  
font-family:'MyWebFont';  
url('Fonts/MyWebFont.woff2')  
}
```

@media

містить умовні інструкції, які застосовують задані стилі, у залежності від певної умови

```
@media (max-width: 600px){  
.screen{  
display:none;  
}
```

наведений приклад скрипту дозволяє вимикати властивість display для значень ширини екрану 600px і менше





Властивості керування потоком документу, заданням розмірів елементів

Основною одиницею для відображення матеріалу (текстового, графічного, відео та аудіо) є бокс – прямокутна область сторінки, у якій розміщується контент.

Поведінку блока на сторінці задається ключовою властивістю **display**. Серед основних значень, яких може набувати ця характеристика, є такі: **block, inline-block, inline, flex, grid, table**.

Значення **block**. За замовчуванням браузер висловлює таке значення для тегів **header, section, footer, div, h1, ..., h6, p, ul, ol** тощо.

**Для всіх тегів HTML, до яких застосовується
властивість із описаним значенням характерні такі
особливості:**

-  Висота боксу підлаштовується під висоту контенту
-  Займає весь доступний простір по ширині
-  Для такого тегу є можливість задати значення висоти, ширини, внутрішні та зовнішні відступи
-  Обов'язковим є примусове перенесення слів на новий рядок, якщо у попередньому речення не поміщається повністю



Основні властивості, які задають параметри бокса, якщо для нього встановлено значення властивості **DISPLAY: BLOCK**

width

задає ширину контенту
(зазвичай використовуються такі одиниці як px, %, auto)

height

висота контенту
(px, auto)

padding

внутрішні відступи
у боксі від меж до
контенту

margin

задає зовнішні
відступи

Аналогічно дозволяється використовувати і такі властивості як: **margin-left, margin-right, margin-top, margin-bottom**

також часто використовуються властивості **padding-left, padding-right, padding-top, padding-bottom**. Вони дозволяють задати внутрішній відступ з однієї сторони (зліва, справа, зверху, знизу)

Властивість **border** задає для боксу рамки. Їх основними компонентами є: товщина, стиль накреслення і колір. Вони задаються за допомогою властивостей **border-width**, **border-style** та **border-color**. Окремі сторони дозволяється модифікувати за допомогою команд **border-left**, **border-right**, **border-top**, **border-bottom**.

Значення **inline** характерне для тегів, які використовуються для форматування фразового контенту. Типовими інструкціями HTML є: **span**, **a**, **strong**, **em**, **b**, **i**, **time** тощо

Значення **inline-block**. Це гібридне значення, яке дозволяє розміщувати елементи із значенням block у рядок. Фактично поєднує особливості застосування першого та другого.



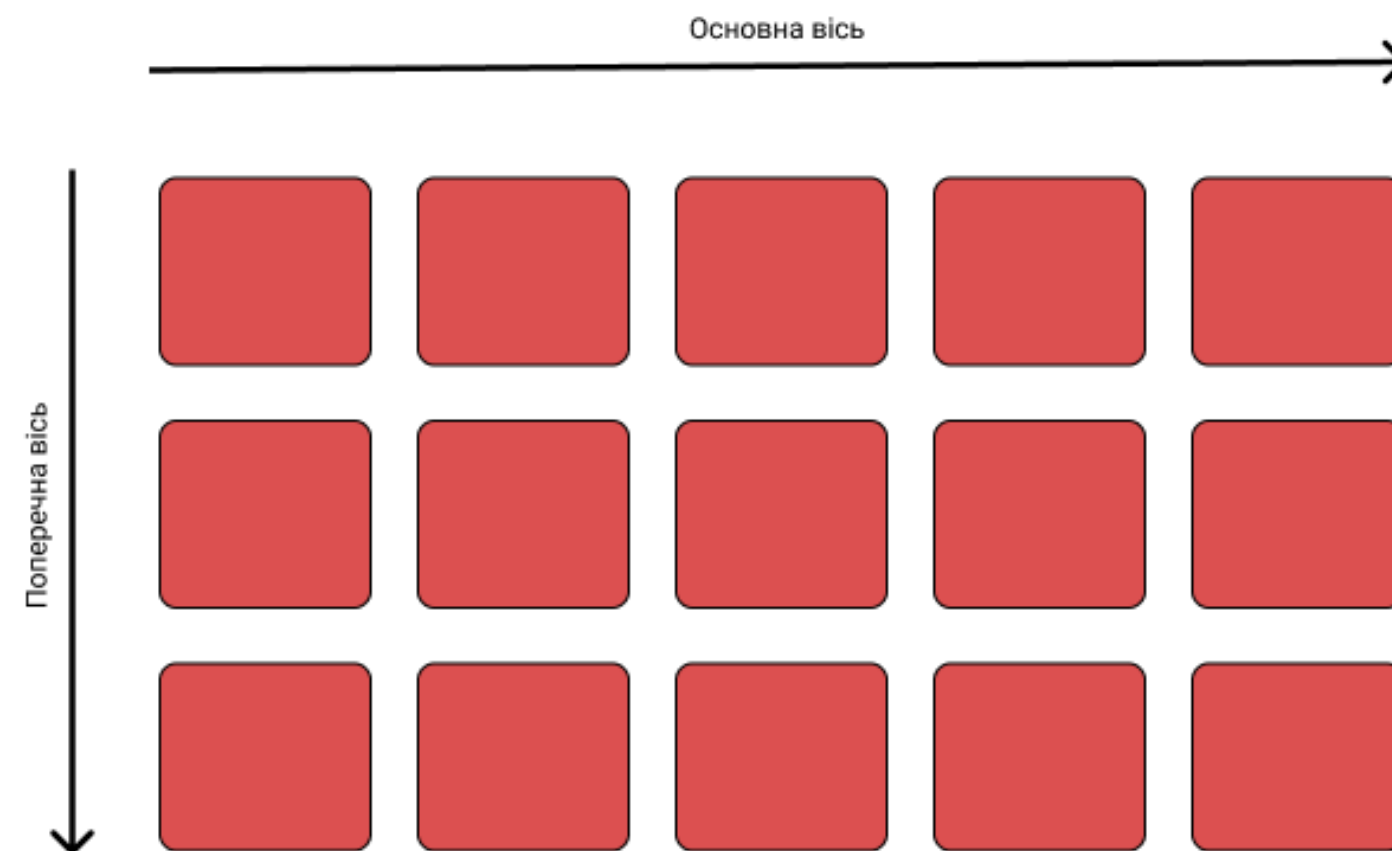


- технологія, яка спростила створення гнучких та адаптивних сторінок для сайтів.

Вона була розроблена як модель однорядкового макетування елементів інтерфейсу і, що також важливо, як один з методів розподілу простору між елементами у рамках контейнеру flexbox. Для неї характерно гнучкість у розміщенні компонентів на сторінці.

При верстці за допомогою цієї технології варто оперувати елементами із точки зору двох осей – основної та поперечної осей.

Основна вісь визначається властивістю **flex-direction**, а поперечна вісь проходить перпендикулярно їй. Все, що ми робимо із компонентами **flexbox**, відноситься до цих осей, тому варто з самого початку зрозуміти, як вони працюють.



До основних властивостей, які дозволяють налаштувати розміщення елементів у flex-контейнері, відносять:

flex-direction, justify-content, flex-wrap, order, align-items, align-self, align-content, flex-grow, flex-shrink і flex-basis

grid

є прикладом ще однієї інтегрованої технології, яка дозволяє виконувати позиціонування об'єктів на сторінці як по горизонталі так і по вертикалі. По своїй суті вона дуже схожа із таблицею, але має значно гнучкіший та простіший по свої суті функціонал.

Серед основних властивостей, які використовуються у рамках CSS Grid варто назвати:

grid-template-columns, grid-template-rows, grid-template-areas, grid-template, grid-column-gap, grid-row-gap, grid-gap, justify-items, align-items, justify-content, align-content, grid-auto-columns, grid-auto-rows, grid-auto-flow, grid, grid-column-start, grid-column-end, grid-row-start, grid-row-end, grid-column, grid-row, grid-area, justify-self, align-self

приклад застосування технології Flex

ФАЙЛ
HTML

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Flexbox Example</title>
```

```
<link rel="stylesheet" href="технологія Flex.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="box"> Каталог товарів </div>
```

```
<div class="box"> Акційні товари </div>
```

```
<div class="box"> Нові товари</div>
```

```
<div class="box"> Цікаві пропозиції</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

визначає відображення веб-сторінки на різних пристроях, особливо на мобільних



name="viewport"

Цей атрибут вказує, що тег управляє властивостями області перегляду (viewport), яка визначає, як веб-сторінка відобразиться на різних пристроях — від мобільних телефонів до настільних комп'ютерів.

content="width=device-width"

Ця частина говорить браузеру, що ширина області перегляду повинна відповідати ширині екрана пристрою. Іншими словами, сторінка буде автоматично підлаштовуватися під ширину екрана користувача, будь то смартфон чи комп'ютер.

initial-scale=1.0

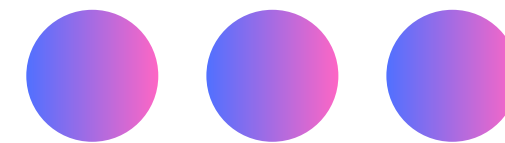
Це встановлює початковий масштаб відображення сторінки на рівні 1.0, тобто вміст сторінки відобразиться в 100% масштабу. Це означає, що сторінка не буде зменшеною чи збільшеною при першому завантаженні, вона відобразиться у своєму реальному розмірі.

ФАЙЛ CSS

```
/* Стили для контейнера */  
.container { display: flex;  
justify-content: space-between;  
align-items: center;  
height: 100vh;  
padding: 20px;  
background-color: #f0f0f0; }
```

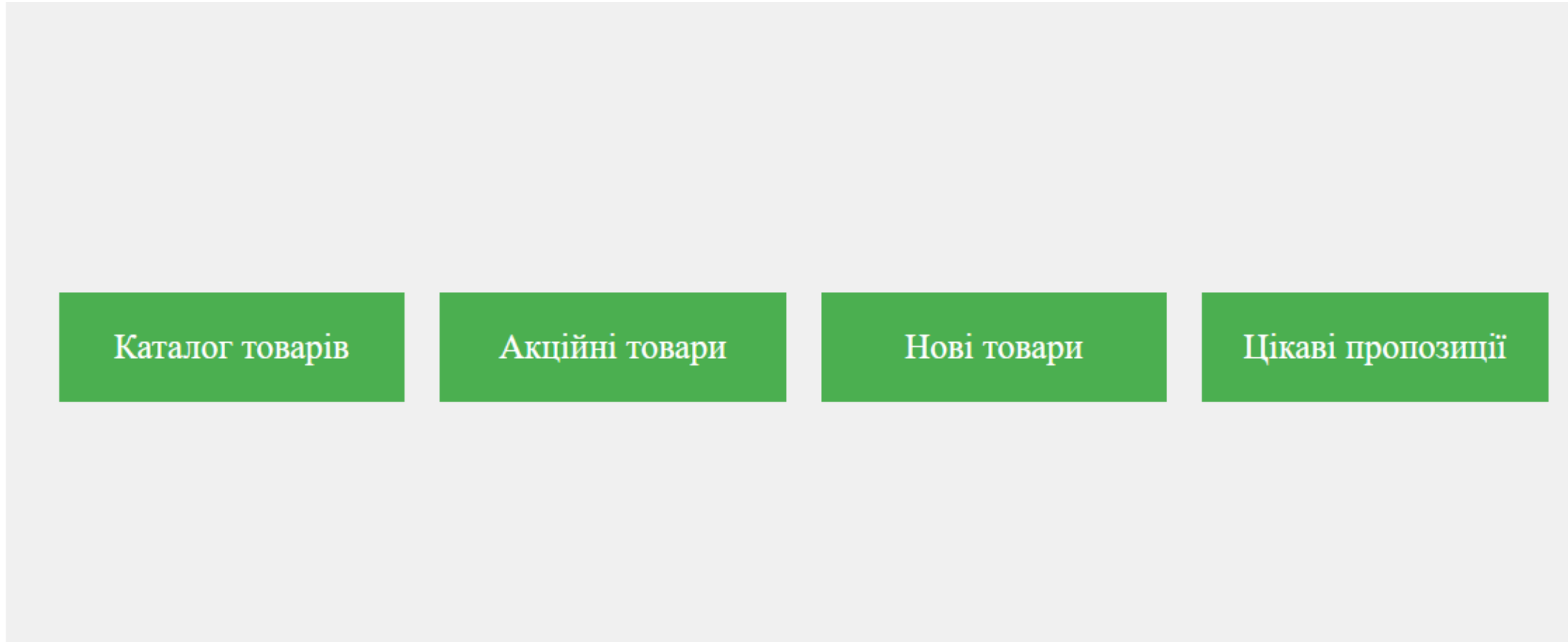
```
/* Стили для элементов */  
.box { background-color: #4caf50;  
color: white;  
font-size: 20px;  
padding: 20px;  
text-align: center;  
flex: 1;  
margin: 10px; }
```

```
/* Використання Flexbox */  
/* Розподіл простору між елементами */  
/* Вирівнювання елементів по вертикалі */  
/* Задаємо висоту контейнера */
```



```
/* Елементи будуть рівномірно розподілені */
```

ВИГЛЯД СТОРІНКИ



Чотири блоки будуть розташовані в один ряд з однаковим простором між ними, а також будуть адаптивно змінюватися в залежності від розміру екрану.

