

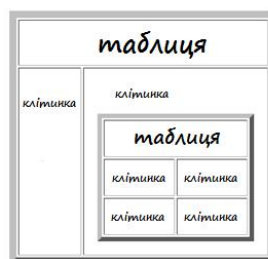
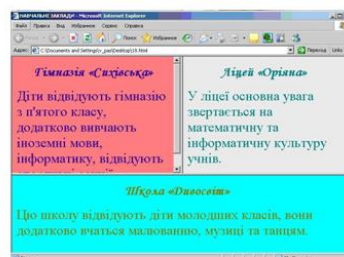
Змістовий модуль 3. Каскадні аркуші стилів CSS3

Тема 6: Верстка сторінки

Верстка сторінок може виконуватись різними способами, і найбільш поширеними є таблиці, фрейми та стилі.

Принципи верстки

- Фрейми
- Таблиці
- Блоки



В той час, як фрейми на нинішній день використовуються дуже рідко, порівнюючи верстку таблицями та стилями, можна виділити наступні переваги та недоліки кожного із цих способів:

	Таблиця	Блоки та стилі
Колонки	Колонки формуються елементами таблиці, їх висота однакова і взаємопов'язана.	Колонки створюються різними блоками, їх висота різна і залежить від змісту.
Ширина	Якщо ширина таблиці явно не вказана, вона обчислюється на основі вмісту таблиці.	За замовчуванням блок займає всю доступну йому ширину.
Розташування	Рядки таблиці відображаються в тому порядку, як вони представлені в коді. Клітинки йдуть зліва направо і зверху вниз.	Порядок блоків у коді може не відповідати їх становищу в браузері.
Завантаження	Як правило, поки таблиці не завантажиться повністю, вміст її не відображатиметься. Якщо на веб-сторінці розміщена велика таблиця, завантаження сторінки істотно сповільнюється.	Блоки відображаються послідовно, у міру завантаження документа.

HTML-код	Код для створення таблиць може бути громіздкий, особливо якщо потрібно об'єднати багато клітинок або зробити кілька вкладених таблиць.	Код, як правило, компактний.
Вирівнювання по висоті	Вміст всередині клітинки легко вирівнювати по її верхньому, нижньому краю або середині.	Вирівнювання нетривіальне.

Основними принципами верстки стилями та блоками є:

- розділення змісту та оформлення;
- використання тегу `<div>`;
- виділення елементів `<header>` `<footer>` `<nav>` `<aside>`;
- використання стилів CSS;
- використання таблиць лише для подання табличних даних.

Блоки створюються або спеціальними тегам `<header>` `<footer>` `<nav>` `<aside>` `<p>` `<h1>` `<table>`, або універсальним тегом `<div>`. Такі блоки розміщуються по вертикалі один під одним і займають всю доступну площу за шириною. Якщо в атрибутах задана ширина блоку, то вона окрім власне ширини вмісту має розмістити границі, поля та відступи. Висота блоку визначається браузером автоматично за його вмістом.

При потребі визначити блок у межах рядка, використовується тег `<span>`, наприклад:

`<span class="pose">Виділена частина</span>` рядка

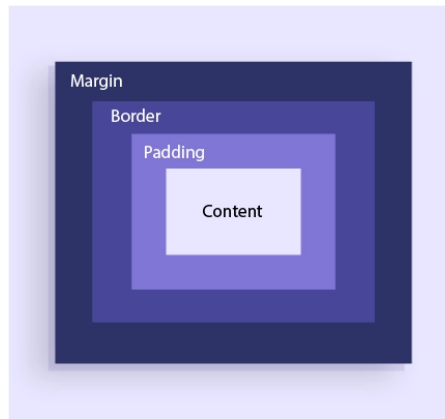
За рекомендаціями W3C, не можна вкладати блоковий елемент (визначає абзац) всередину рядкового (визначає сукупність символів, не обов'язково окремий абзац):

`<a><h1>Заголовок</h1></a>` – неправильно
`<h1><a>Заголовок</a></h1>` – правильно

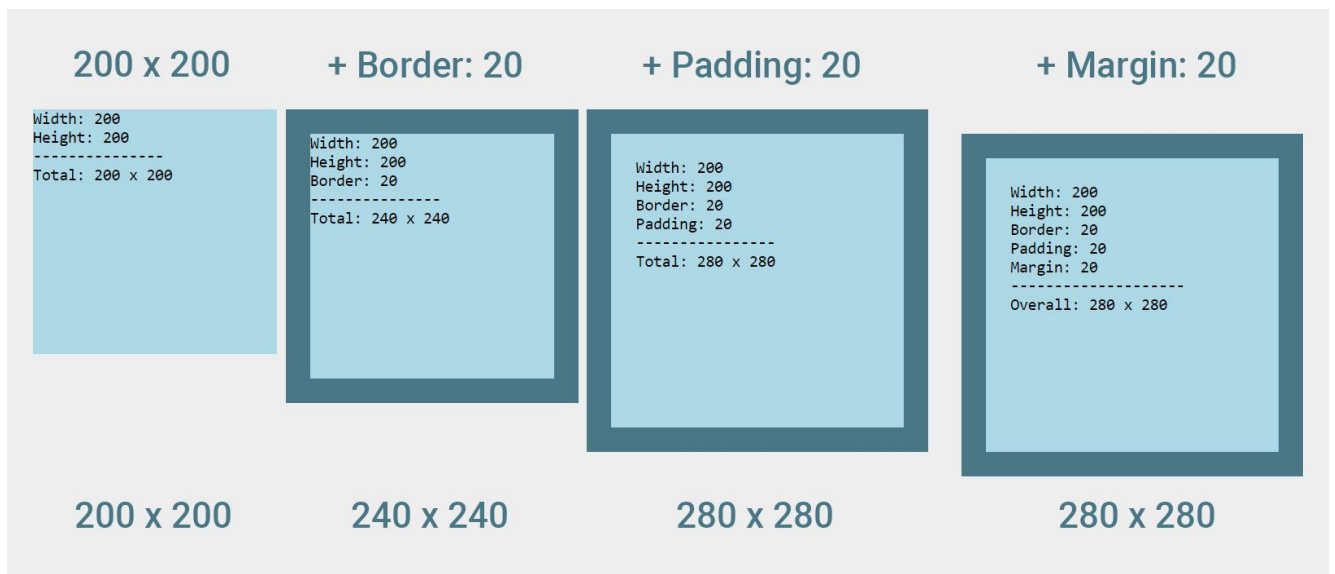
Поля та відступи

Для кожного блоку можна налаштувати параметри полів та відступів:

margin зовнішні поля
border межі – ширина, тип, кути
padding внутрішні відступи
background тло вмісту div



Приклад:



Показ елемента (властивість display)

Ще однією важливою властивістю верстки є властивість показу **display** - вона вказує ЯК і ЧИ елемент показується на сторінці.

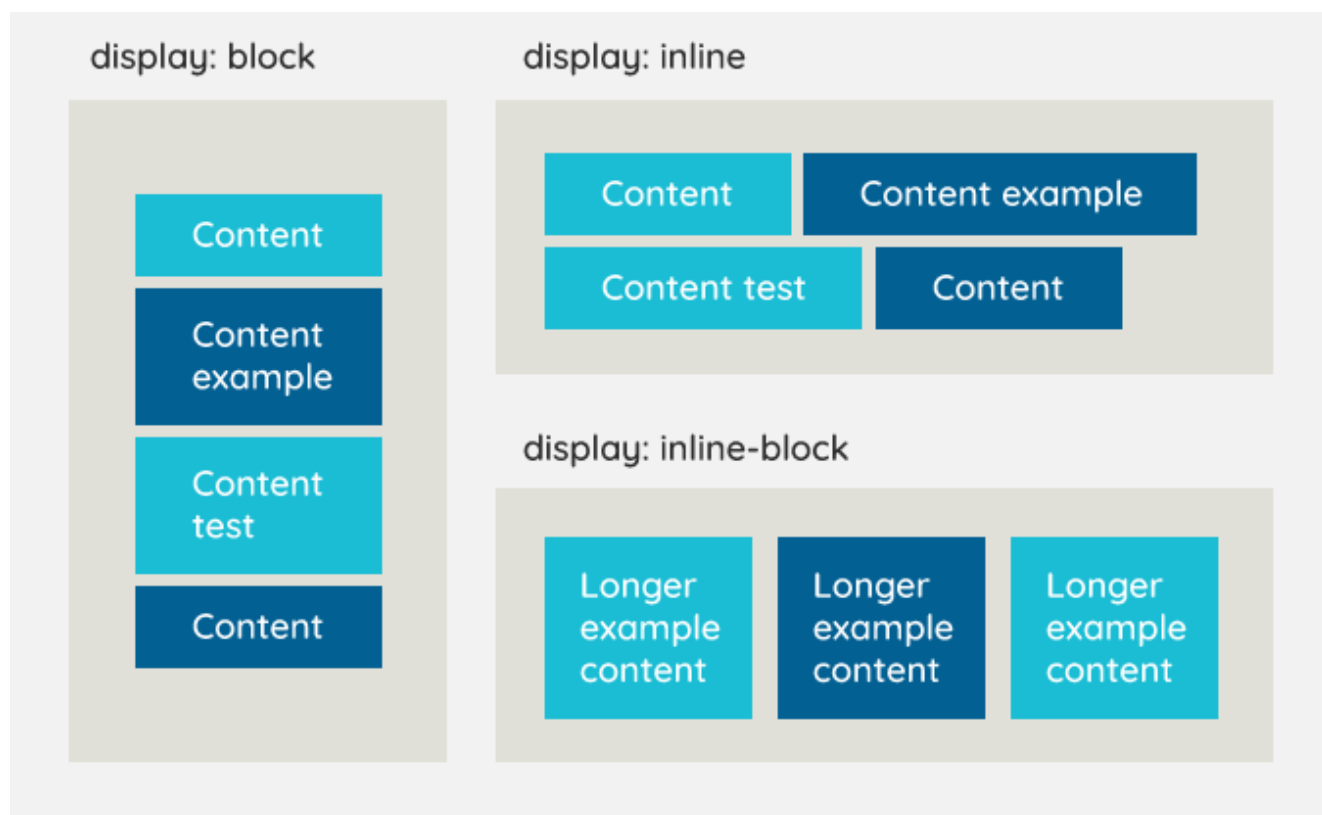
Так властивість **display: none;** приховує елемент, не залишаючи для нього місця (є ще одна схожа властивість: **visibility: hidden;** проте вона передбачає відведення простору під прихований елемент).

display: inline; перетворює елемент на рядковий. Якщо застосувати його, наприклад до тегу `<div>`, він стане вмістом, аналогічним `<span>` (наприклад, неможливо задати висоту).

display: block; - перетворює елемент на блок. Якщо застосувати його, наприклад до тегу `<span>`, він стане блоком, аналогічним `<div>`

display: inline-block; - елемент, форматування вмісту якого відбувається блоком, а сам елемент розміщується на сторінці рядково.

Приклад:

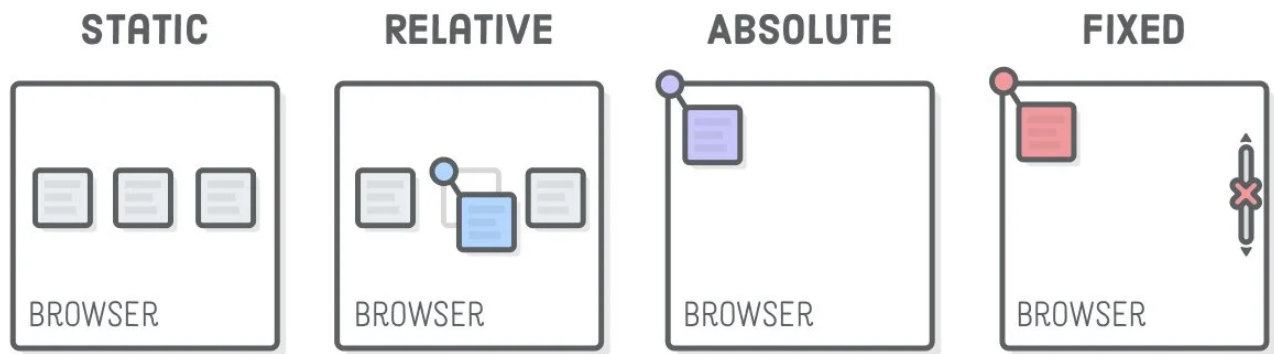


Позиціонування (властивість **position**)

При розміщенні елементів на сторінці можна використовувати один із таких режимів:

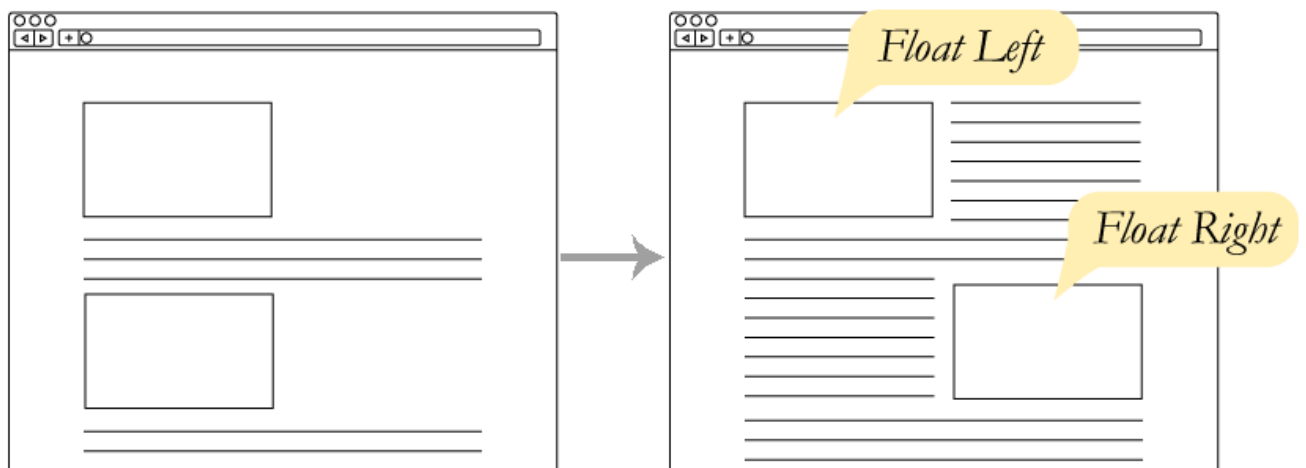
- **static** – на вільне місце після попереднього об'єкта;
- **absolute** – точне місце розміщення;
- **fixed** – не рухається при прокручуванні;
- **relative** – розташування об'єкта відносно того місця, яке би він зайняв, якби для нього використовувалось статичне позиціонування.

Приклад:



Для **відносного** та **абсолютного** позиціонування можна задавати відступи для розміщення об'єкта по горизонталі **left** (ліворуч) та **right** (праворуч), та по вертикалі **top** (згори) та **bottom** (знизу).

Властивість **float** визначає, до якої сторони сторінки «прикріплюється» вміст:

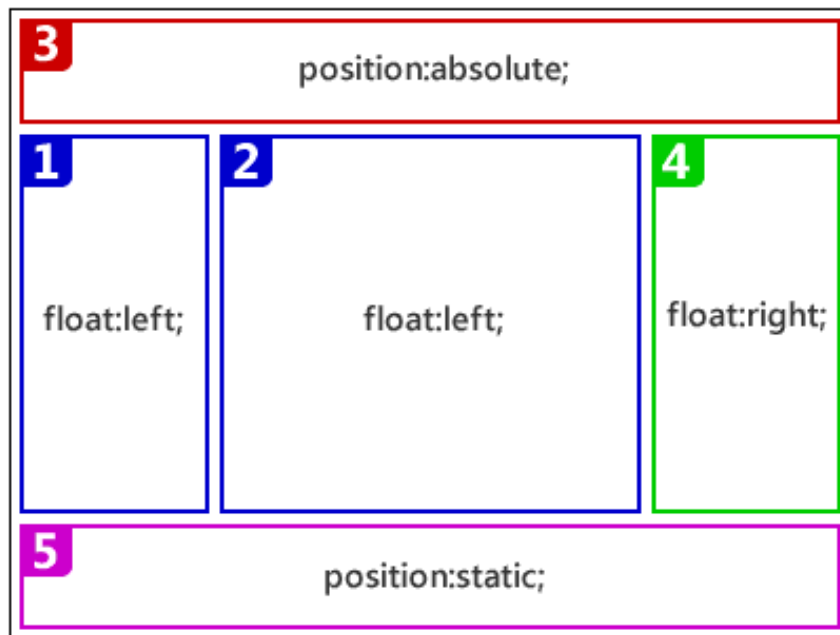


Для вирівнювання по центру: **{margin: auto; width: 50%;}**

Налаштування тла блоку:

- **background: url(header.jpg)** зображення
- **repeat-x** повтор по горизонталі
- **background-attachment: fixed;** не прокручується
- **no-repeat** без повтору
- **center** вирівнювання по центру

На зображенні можна переглянути, як буде розміщено елемент у різних режимах:



Наприклад:

- при використанні

.first {position:fixed; top:450; left:500;}

елемент буде нерухомим при прокручуванні;

- при використанні

.second {position:absolute; top:100; left:400}

елемент буде показаний поверх інших у заданому розміщенні;

- при використанні

.third {position:relative; top:200;}

на місці елемента буде порожньо, а сам елемент буде показаний зі зміщенням.

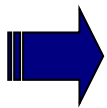
Стиль

```
.feedback {  
  position: fixed; /* Фіксоване розміщення */  
  right: 0; top: 50%; /* Координати */  
  background: red; /* Колір тла*/  
  width: 70px; /* Ширина блоку*/  
  height: 70px; /* Висота блоку*/  
}
```

визначає створення квадрату червоного кольору 70×70 пікселів посередині висоти сторінки і з нульовим відступом справа. Він може бути зафіксованим при прокручуванні, розміщений абсолютно **absolute** чи плаваюче **float**. Так можна додавати, так звані, візки: цитати, відгуки, коментарі.

При позиціюванні варто враховувати, що існують стилі, визначені стандартно, наприклад відступи між заголовками чи на початку сторінки. Для того, щоб скинути ці значення варто задати стиль, який стосуватиметься усіх елементів (або вказувати конкретні теги, для яких видаляти стандартні відступи та поля):

*** {margin: 0; padding: 0; }**



Приклади та заготовки CSS-розміток – за посиланням:

<http://www.dynamicdrive.com/style/layouts/>.

Гرادієнти

За допомогою стилів CSS можна створювати градієнти, це атрибут **linear-gradient**, у якому слід вказати напрямок градієнта та назви кольорів (два і більше).

Наприклад, **background: linear-gradient(red, yellow, green);** - лінійний градієнт з трьох кольорів.

Варіанти градієнта: **to right** (зліва направо), **to bottom right**, (кутовий – до правого нижнього кута), або за допомогою вказання кута повороту градієнта, наприклад **-90deg**.

background: linear-gradient(to right, red,orange,yellow,green,blue,indigo,violet); - горизонтальна веселка

background: radial-gradient(red, yellow, green); - радіальний градієнт

Наприклад, стиль

.content {background: linear-gradient(orange, yellow);}

створює градієнт на всю висоту сторінки (чи блоку) – вгорі буде оранжевий колір, внизу – жовтий.

Задавши додаткову властивість **background-attachment: fixed;** отримаємо зафіксований градієнт тла, яке не рухається при прокручуванні сторінки.

У стилі можна вказати адресу зображення властивістю **background: url(header.jpg)**. У такому разі за умовчанням це зображення замостить простір по горизонталі та вертикалі. Якщо потрібно розмістити зображення лише по горизонталі, задають атрибут **repeat-x** (**repeat-y** – відповідно, по вертикалі). Зверніть увагу на скорочений запис того факту, що цей атрибут стосується властивості **background**, тому перед repeat-x **НЕМАЄ** крапки з комою.

.header {background: url(header.jpg) repeat-x; }

Можна, наприклад, вказати відсутність замощення і розташувати зображення по центру блоку **no-repeat center**;

Якщо ж додамо **background-attachment: fixed**; зображення тла буде зафіксоване при прокручуванні (щоправда, поки відвідувач не прокрутить вміст сторінки до наступного блоку).

Наприклад, **body { background: url(background.jpg) fixed no-repeat center; }** розміщує нерухоме зображення по центру сторінки.

Поради

Іноді елементи сторінки можуть накладатися один на одний. У нижній частині такої групи об'єктів буде розташований об'єкт, який описали першим, а зверху – той об'єкт, який описали останнім. Можна також задати властивість **z-index: 2**

Іноді потрібно задати обтікання об'єкту текстом: **float: right**;

Оскільки серед властивостей стилів немає можливості задати висоту блоку, то для розтягування блоку на всю висоту вікна можна скористатись комбінацією стилів **position: fixed; min-height: 100%**;

У разі, якщо вміст блоку перевищує його розмір (або може перевищити), варто задати властивість переповнення **overflow: auto**; (блок при цьому буде розтягуватись).



Додаткова інформація по CSS: <https://learnlayout.com/>.

ВПРАВИ ТА ПРАКТИЧНІ ЗАВДАННЯ ДО ВИКОНАННЯ

1. Практичне завдання «Позиціювання»:

Побудувати роботу на основі заготовки:
<https://trinket.io/library/trinkets/af5c5fb2e20c> .

Наприклад, можна задати такі стильові властивості:

```
#eyes1 {  
  width: 200px;  
  position: absolute;  
  top: 200px;  
  left: 100px;  
}
```

Відомості та рекомендації щодо виконання даного завдання — див. Лекцію 6.