

FRONT-END SkillUP

заняття 2

План заняття

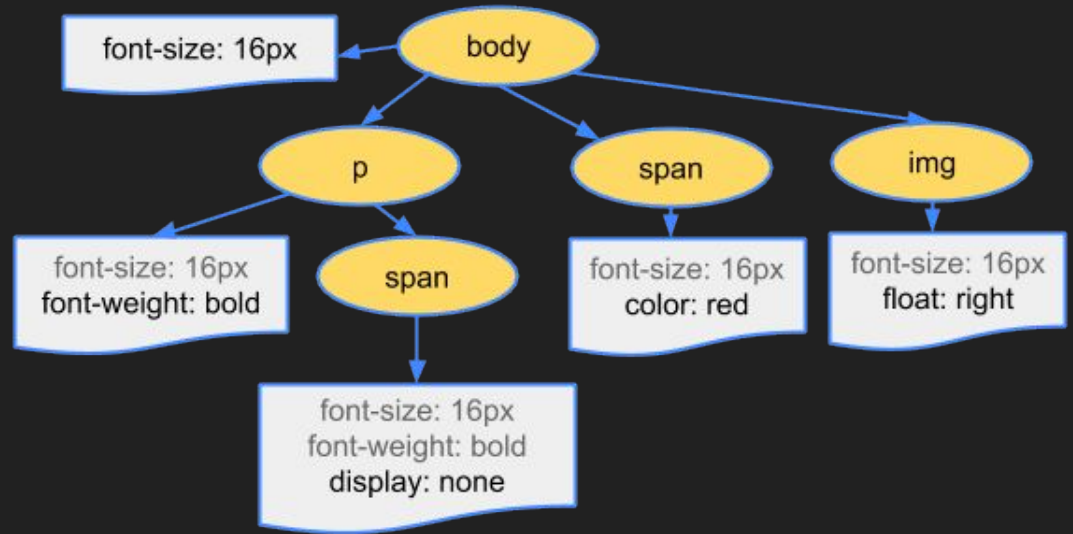
1. Процес відображення сторінки
2. CSS
3. Практика таблиці, блоки

Процес відображення сторінки

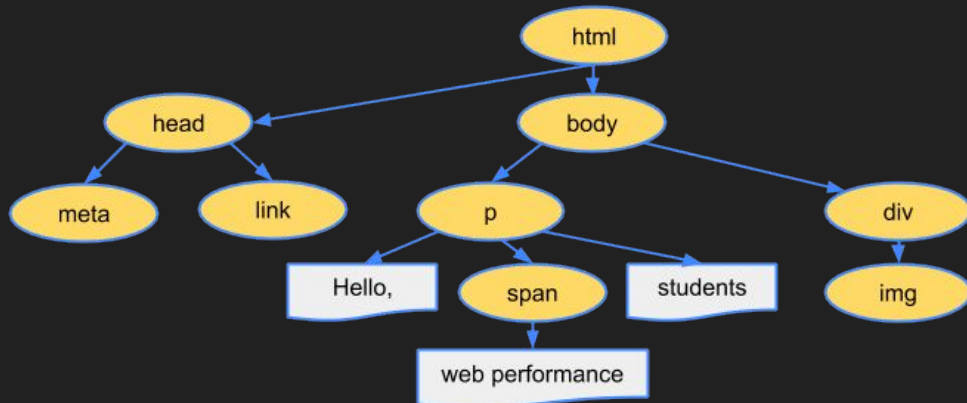
Послідовність роботи браузера при відображенні документа:

1. З отриманого HTML-документа формується DOM (Document Object Model).
2. Завантажуються і розпізнаються стилі, формується CSSOM (CSS Object Model).
3. На основі DOM і CSSOM формується дерево рендерингу, або render tree — набір об'єктів рендерингу (Webkit використовує термін «renderer», або «render object», а Gecko — «frame»). Render tree дублює структуру DOM, але сюди не потрапляють невидимі елементи (наприклад — <head> або елементи зі стилем display:none;)
4. Також, кожен рядок тексту представлений в дереві рендерингу як окремий renderer. Кожен об'єкт рендерингу містить відповідний йому об'єкт DOM (або блок тексту), і розрахований для цього об'єкта стиль. Простіше кажучи, render tree описує візуальне уявлення DOM.
5. Для кожного елемента render tree розраховується положення на сторінці — відбувається layout. Браузери використовують потоковий метод (flow), при якому в більшості випадків достатньо одного проходу для розміщення всіх елементів (для таблиць проходів потрібно більше).
6. Нарешті, відбувається відображення в браузері — painting.

CSSOM



DOM



Repaint

Repaint

У разі зміни стилів елемента, які не впливають на його розміри і положення на сторінці (наприклад, `background-color`, `border-color`, `visibility`) браузер просто промальовує його заново, з урахуванням нового стилю — відбувається `repaint` (або `restyle`).

Reflow

Якщо ж зміни зачіпають вміст, структуру документа, положення елементів — відбувається reflow (або relayout). Причинами таких змін звичайно є:

- Маніпуляції з DOM (додавання, видалення, зміна, перестановка елементів);
- Зміна вмісту, в т.ч. тексту в полях форм;
- Розрахунок або зміна CSS-властивостей;
- Додавання, видалення таблиць стилів;
- Маніпуляції з атрибутом class
- Маніпуляції з вікном браузера — зміни розмірів, прокрутка;
- Активація псевдо-класів (наприклад, :hover).

Оптимізація з боку браузера

Браузери по можливості локалізують repaint і reflow в межах елементів, що піддали зміні.

Наприклад, зміна розмірів абсолютно або фіксовано позиційованого елемента торкнеться тільки самого елемента і його нащадків.

Зміна статично позиційованого — спричинить reflow всіх елементів, наступних за ним.

[How browsers work](#)

CSS

Каскадні таблиці стилів (англ. **Cascading Style Sheets** або скорочено **CSS**) — спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

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.

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).

CSS також дозволяє адаптувати контент до різних умов відображення (на екрані монітора, мобільного пристрою (КПК), у роздрукованому вигляді і т.д.).

Один і той самий HTML або XML документ може бути відображений по-різному залежно від використаного CSS.

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
- Сторінки зменшуються в об'ємі та стають більш структурованими, оскільки інформація про стилі відділена від тексту;
- 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.



CSS підключення

Щоб включити таблицю стилів CSS, потрібно зберегти код CSS у файл (це можна зробити будь-яким текстовим редактором), наприклад `example.css`, і потім включити або імпортувати його в HTML сторінку.

```
<link rel="stylesheet" href="example.css" type="text/css">
```

CSS підключення

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {color:red;}
p {color:blue;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

This is a heading

This is a paragraph.

CSS підключення

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:left">This is a header</h1>
<p style="color:green">This is a paragraph.</p>

</body>
</html>
```

This is a header

This is a paragraph.

CSS 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 та блок визначення.

Блок визначення складається із оточеного фігурними дужками списку властивостей.

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

A CSS rule has two main parts: a selector, and one or more declarations:

Selector	Declaration	Declaration		
<code>h1</code>	<code>{ color:blue; font-size:12px; }</code>			
	<code>color:blue;</code>	<code>font-size:12px;</code>		
	Property	Value	Property	Value

The selector is normally the HTML element you want to style.

Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

CSS селектори

ID селектори - самий потужний тип селекторів. Це означає, що вони "витісняють" інші типи селекторів, а стилі вказані з цим селектором застосовуються першочергово.

```
<div id="news">  
  some text inside div  
</div>
```

```
#news {  
  some styles for div  
}
```

Це, можливо, самий використовуваний і універсальний тип селектора. Ви можете додавати декілька класів (через пробіл) до HTML елемента.

```
<div class="news">  
  some text inside div  
</div>
```

```
.news {  
  some styles for div  
}
```

Цей тип селекторів дуже корисний при зміні унікальних властивостей HTML документа. Не зловживайте цим типом селекторів. Краще використовувати стилювання за класом.

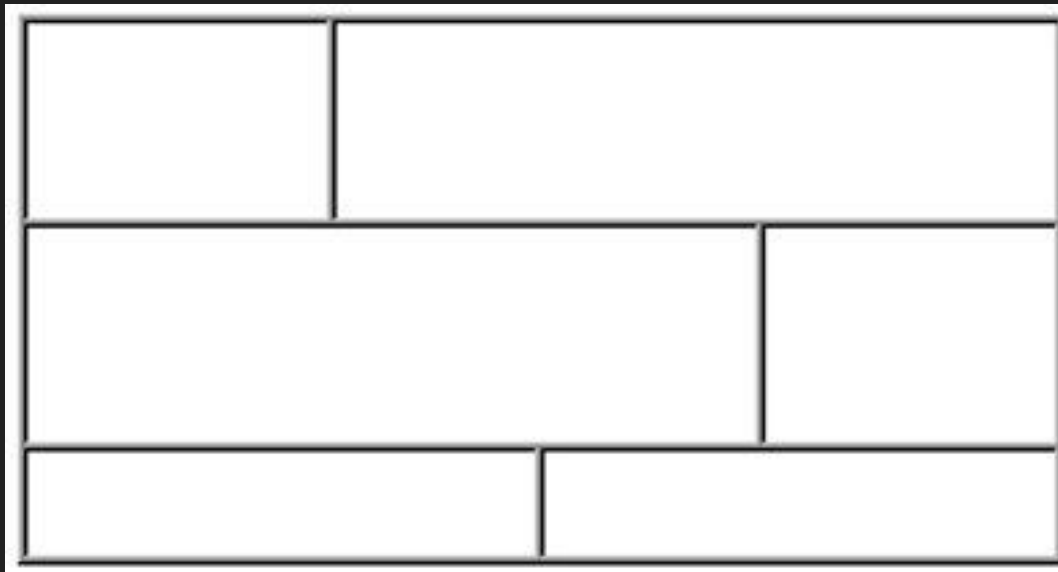
```
<div>  
  some text inside div  
</div>
```

```
div {  
  some styles for div  
}
```

Додаткова інформація

- CSS коментарі
- CSS одиниці виміру
- CSS кольори
- CSS box model

Практика



Домашнє завдання

1. <https://coderepen.io/tholman/pen/EwIKd> - переглянути, і спробувати пояснити чому саме такий колір браузер відображає.
2. Використовуючи блочну і табличну верстку (результатом має бути два варіанти), реалізувати:

1	2	5	6	7
3	4	8		

Література

CSS 3:

<http://ptgmedia.pearsoncmg.com/images/9780321888938/samplepages/0321888936.pdf>

CSS History: http://en.wikipedia.org/wiki/Cascading_Style_Sheets

Non-standard fonts:

<http://www.acumensoftwaredesign.com/css/non-standard-fonts-css-tricks>

Efficient CSS:

<http://speckyboy.com/2011/03/08/website-speed-part-1-write-more-efficient-css/>

Media Queries: <http://frontender.com.ua/media-queries/#media-1>