

Що таке CSS

CSS (*Cascading Style Sheets* - каскадні таблиці стилів) - одна з базових технологій у сучасному Інтернеті. Нечасто можна зустріти сайт, зверстаний без примінення CSS.

CSS-код - це список інструкцій для браузера, - *як і де* відображати елементи веб-сторінки, написаний особливим чином. Під «елементами» зазвичай маються на увазі теги XHTML / HTML і їх вміст.

Що таке CSS

Інструкції CSS зручно зберігати у вигляді окремого текстового файлу з розширенням `.css` , або у вигляді окремого текстового фрагмента на початку XHTML/HTML-документа.

Основна ідея CSS в тому, щоб відокремити дизайн документа від його вмісту. CSS відповідає за оформлення і зовнішній вигляд, а XHTML/HTML - за зміст і логічну структуру документа.

Подивимося на фрагмент XHTML-документа:

<h1 > Казка </ h1 >

<p > В одній далекій країні, на краю болота, під пеньком, жив їжачок. І ось одного разу ... </ p >

Що таке CSS

Зі службової XHTML розмітки ми бачимо тільки елемент заголовка h1 та абзацу p , і ні слова про оформлення - шрифтах, кольорі тексту, тлі, відступи та інші елементи дизайну. Все це покладено на CSS:

/* Оформляємо заголовки: */

```
h1 {  
  color : red ;  
  background-color : yellow ;  
  font : Tahoma 2em ;  
}
```

/* оформляємо абзаци тексту: */

```
p {  
  color : grey ;  
  line-height : 150 % ;  
}
```

Включення CSS в HTML документ

Для того, щоб застосувати таблицю стилів до HTML-документу, ми можемо обрати один з трьох способів, або комбінувати їх:

- застосувати *зовнішні стилі* (у вигляді окремого текстового .css - файлу) за допомогою елемента link;
- *вбудувати стилі* безпосередньо в HTML-документ (у вигляді блоку css-тексту) за допомогою елемента style;
- застосувати *inline-стиль* , тобто призначити стиль конкретному HTML-елементу безпосередньо в документі, за допомогою HTML-атрибуту style.

Розберемо ці способи докладніше.

Зовнішні стилі (external style sheets)

Застосовуються за допомогою елемента `link` , який повинен розташовуватися всередині елемента `head` і ніде більше.

`<Link REL = "stylesheet" type = "text / css" href = "mystyle.css" media = "all" />`

Зустрівши в HTML-документі цей тег, браузер завантажить з сайту CSS-файл (у нашому випадку це *mystyle.css*) і застосує до документа стилі, що містяться в ньому. Файл не повинен містити нічого, крім CSS-інструкцій.

Зовнішній файл зі стилями зручний тим, що одні й ті ж стилі можна застосовувати до безлічі документів на сайті - в кожному з них достатньо лише вписати один рядок з елементом `link`.

Таблиці стилів документа (document style sheets)

Називаються так тому, що розташовуються безпосередньо в HTML-документі і застосовуються лише до нього. Іноді називаються *embedded style sheet* (вбудований стиль).

CSS-стилі та коментарі розташовуються між відкриваючим і закриваючим тегамі елемента style :

```
<Style Type = "text / css" >  
... </ style >
```

Сам тег style (на відміну від link) може знаходитися в будь-якій частині документа, але зазвичай його розміщують всередині елемента head.

Вбудовані стилі безпосередньо в HTML-документ

```
<html>
```

```
<head>
```

```
<title>Классы</title>
```

```
<style type="text/css"> P { /* Обычный параграф */ text-align: justify; /*  
Выравнивание текста по ширине */ } P.cite { /* Параграф с классом cite  
*/ text-align: justify; /* Выравнивание текста по ширине */ color: navy; /*  
Синий цвет текста */ font-style: italic; /* Курсивное начертание */ }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p> тест, тест </p>
```

```
<p class="cite"> тест, тест.</p>
```

```
</body>
```

```
</html>
```

Застосування зовнішнього *стилю*

```
<html>
```

```
<head>
```

```
<title>Стили</title>
```

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

```
  <link rel="stylesheet" type="text/css" href="http://www.htmlbook.ru/main.css">
```

```
</head>
```

```
  <body>
```

```
    <h1>Заголовок</h1>
```

```
    <p>Текст</p>
```

```
  </body>
```

```
</html>
```


Стилі, підставляються в рядок (inline styles)

Іноді потрібно призначити стиль окремого елемента на сторінці, не застосовуючи зовнішніх стилів і елемента style. Типовий випадок - елемент зустрічається один раз в документі або на сайті, але вимагає особливого оформлення. Скористаємося атрибутом style (саме атрибутом елементів, а не елементом!):

<P Style = "color: red" > Я абзац, виділений червоним кольором, інших таких на сайті немає </ p >

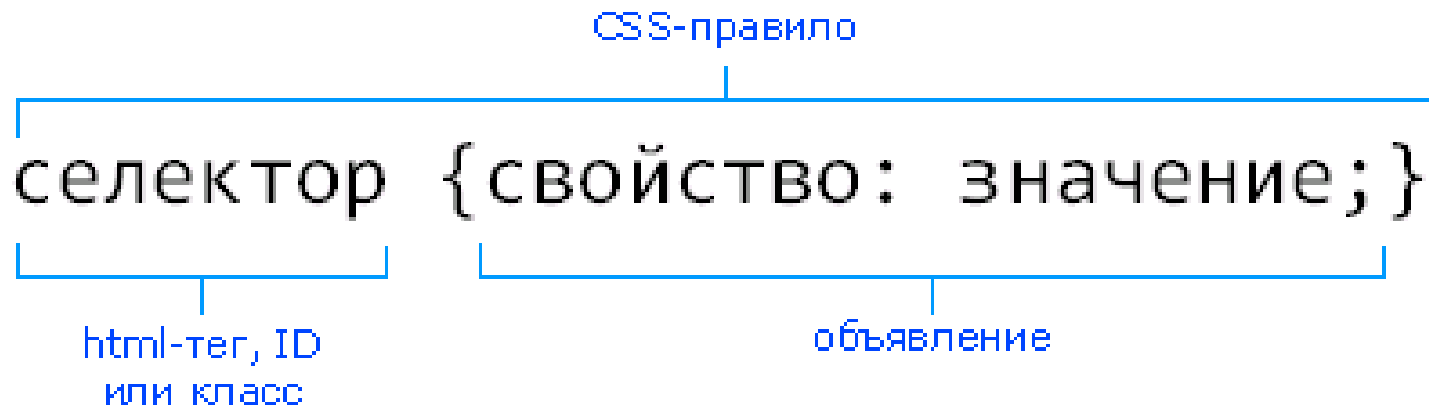
Атрибут style є майже у всіх HTML-елементів. Крім тих, що розташовуються поза елементом body.

Усередині атрибута style можна написати кілька CSS оголошень, розділених крапкою з комою, фігурні дужки не використовуються.

Зауваження: уникайте використання вищеописаного способу. Inline-стилі змішують вміст документа і його дизайн, у той час як ідеологічно більш правильно відокремлювати вміст документа та інформацію про його оформлення.

Синтаксис CSS

Всі CSS-правила складаються з селектора і блоку оголошень (укладеного у фігурні дужки). Всередині блоку оголошень (усередині фігурних дужок, простіше кажучи) може знаходитися одне або кілька оголошень, розділених крапкою з комою. Оголошення - це рядок, складений з css-властивості та її значення.



Як це може виглядати на практиці? Ось три приклади CSS-правил:

```
a {text-decoration: none;}
```

```
p.announce {border: 1px dashed black;}
```

```
/ * тут селектор - p.announce * /
```

```
p.note { display: block; /*так, оголошень може бути кілька*/
```

```
float: right; / * і їх не обов'язково писати на одному рядку * /
```

```
}
```

Кожне правило починається з селектора (показчика), що вказує на ті html-елементи, до яких ми збираємося застосувати css-правило.

У блоці оголошень відбувається найцікавіше - ми встановлюємо правила оформлення обраних нами елементів, перевизначення їх властивостей - розміри, колір, бордюри, поля, положення на екрані і т.д.

```
<p class="note">Спеціальна пропозиція для дівчат! Рожеві слони!! тільки у нас!!!</p>
```

Селектори

Щоб застосувати css-оформлення до HTML-елементу або безлічі елементів, зазвичай використовуються селектори - спеціальні покажчики на HTML-об'єкти, до яких ми плануємо застосувати css-правило.

Ось три основних види селекторів.

HTML селектори

Це найпростіший випадок - в якості селектора ми використовуємо ім'я того html-елемента, який хочемо змінити. Наприклад, для тега `` селектором буде `strong`. Відповідно, для тега `<h1>` селектором буде `h1`, і так далі. Тепер ми можемо перевизначити зовнішній вигляд всіх таких елементів в нашому документі:

```
strong {font-weight: normal; color: red;}
```

```
h1 { font: bold 10pt verdana; }
```

Селектори класу

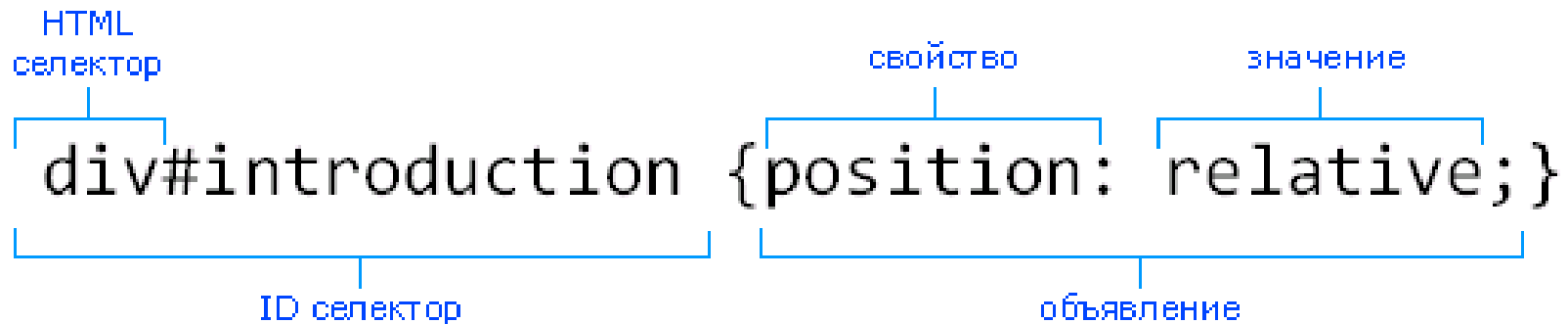
«Клас» - це якесь ім'я, рядок, який ми можемо застосувати до будь-яких HTML-тегів, щоб згодом посилатися на них - на ім'я класу. Як ім'я класу ви можете використовувати практично будь-який рядок. Зручність таких селекторів в тому, що можна привласнити одне ім'я класу безлічі html-тегів у документі і потім управляти їх зовнішнім виглядом, звертаючись до них по імені класу:

The diagram illustrates the components of a CSS class selector and declaration. On the left, the text `.currentDate` is shown with a bracket underneath it labeled "Селектор класу" (Class selector). To its right is a declaration block `{font-size: 8pt;}`. A large bracket underneath the entire declaration block is labeled "объявление" (Declaration). Above the declaration, two smaller brackets are present: one over `font-size:` labeled "свойство" (Property) and another over `8pt;` labeled "значение" (Value).

. myClass {font: bold 10pt verdana;} / * міняємо шрифт для всіх тегів з класом myClass * /

ID селектори (або ідентифікатори)

Будь-який ідентифікатор (ID) - це якесь ім'я, яке ви, так само, як і у випадку з класами, можете застосувати до будь-якого HTML-тегу. Основна відмінність - ID повинен бути унікальний в рамках html-документа:



`# myObj {margin: 1em;} / *` змінюємо поля для елемента, у якого `id = "myObj"` `* /`

`span # today {margin: 1em;} / *` змінюємо поля для елемента `span`, у якого `id = "today"` `* /`

```
#top {  
background-color: #ccc;  
padding: 1em  
}  
  
.intro {  
    color: red;  
font-weight: bold;  
}
```

```
<div id="top">  
<h1>Chocolate curry</h1>  
<p class="intro">This is my recipe for making curry purely with chocolate</p>  
<p class="intro">Mmm mm mmmmm</p>  
</div>
```

Різниця між ідентифікатором `id` і класом полягає в тому, що ідентифікатор може застосовуватися для визначення одного елемента, а клас може використовуватися для визначення декількох елементів.

Як застосувати один стиль до кількох селекторів

Дуже поширене завдання - застосувати один набір правил до декількох різних селекторів. Це робиться елементарно - достатньо перерахувати селектори через кому:

The diagram illustrates the components of a CSS rule. On the left, the selector `h1,p.note,#footer` is shown with a bracket underneath labeled "Список селекторів через запяту". To its right is the declaration `{font-size: 8pt;}`. A bracket underneath the entire declaration is labeled "объявление". Above the declaration, a bracket for `font-size` is labeled "свойство" and a bracket for `8pt` is labeled "значение".

Згодом оформлення для селекторів можна перевизначити індивідуально:

```
/ * Всі параграфи і списки робимо червоними, шрифтом  
Tahoma * /
```

```
p, li, ul, ol {color: red; font-face: Tahoma, sans-serif;}
```

```
/* Перевизначає колір на синій для нумерованих списків */
```

```
ol {color: blue;}
```


Селектори, що залежать від контексту

Ми навчилися встановлювати стилі для елементів HTML незалежно від того, де саме в документі вони розташовані. Але CSS трохи гнучкіше, ніж здається. Ми можемо «прицілюватися» точніше і визначати стилі для елементів залежно від контексту (англ. Contextual Selectors).

Ось, подивіться:

```
/ * Всі посилання, що знаходяться всередині списків, стануть червоними: * /
```

```
li a {color: red;}
```

```
/ * Всі посилання в параграфах, що знаходяться всередині таблиць, стануть зеленими: * /
```

```
table p a {color: green;}
```

Це найпоширеніший метод створення контекстуальних селекторів, який називається «селектором нащадків». Розберемо його докладніше.