

Ідентифікатори

План:

1. Ідентифікатори.
2. Селектори нащадків.
3. Псевдокласи і псевдоелементи.
4. Оформлення гіперпосилань.
5. Селектори атрибутів.

Ідентифікатори

Ідентифікатор – це унікальне ім'я для елемента на веб-сторінці, яке використовується для зміни його стилю і звернення до нього через скрипти. Ідентифікатор починається з символу # і повинен містити латинські символи, дефіс або підкреслення. Використання українських букв у іменах ідентифікаторів заборонено.

Ідентифікатори повинні бути унікальними, тобто зустрічатися в коді документа тільки один раз. Щоб звернутися до елемента за його ідентифікатором, використовують атрибут id із значенням, яке відповідає імені ідентифікатора, але без символу "#".

Наприклад, в коді HTML можна визначити стиль елемента div з ідентифікатором "help" наступним чином:

<div id="help">

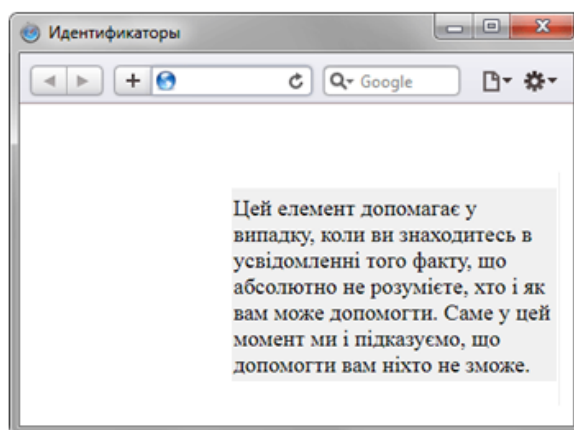
Цей елемент допомагає у випадку, коли ви знаходитесь в усвідомленні того факту, що абсолютно не розумієте, хто і

як вам може допомогти. Саме у цей момент ми і підказуємо, що допомогти вам ніхто не зможе.

</div>

А в CSS можна задати стиль для цього елемента за допомогою ідентифікатора:

```
#help {  
    position: absolute;  
    left: 160px;  
    top: 50px;  
    width: 225px;  
    padding: 5px;  
    background: #f0f0f0;  
}
```



Результат використання ідентифікатору

Якщо використовувати ідентифікатори для конкретного тегу, то синтаксис буде наступним: **Тег # ім'я ідентифікатора {властивість1: значення; властивість2: значення; ...}** Спочатку потрібно вказати ім'я тега, а потім без пробілів – символ решітки і назву ідентифікатора. У

прикладі нижче показано, як використовувати ідентифікатори відносно тегу `<p>`.

Приклад. Ідентифікатор спільно з тегом

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>Ідентифікатори</title>
  <style>
    P {
      color: green; /* Зелений колір тексту */
      font-style: italic; /* Курсивне накреслення тексту */
    }
    P#opa {
      color: red; /* Червоний колір тексту */
      border: 1px solid #666; /* Параметри рамки */
      background: #eee; /* Колір фону */
      padding: 5px; /* Поля навколо тексту */
    }
  </style>
</head>
<body>
  <p> Звичайний параграф</p>
  <p id="opa"> Параграф незвичайний</p>
</body>
</html>
```

Звичайний параграф

Параграф незвичайний

Вид текста после применения стиля

У цьому прикладі ми вводимо стиль для тегу `<p>` та для тегу `<p>` з ідентифікатором "opa". Селектори нащадків, також відомі як контекстні селектори, використовуються, коли необхідно додати стиль до певних тегів на всій сторінці. Наприклад, щоб зробити всі посилання `<a>` не підкресленими, можна використати таке правило:

`a {text-decoration: none;}`

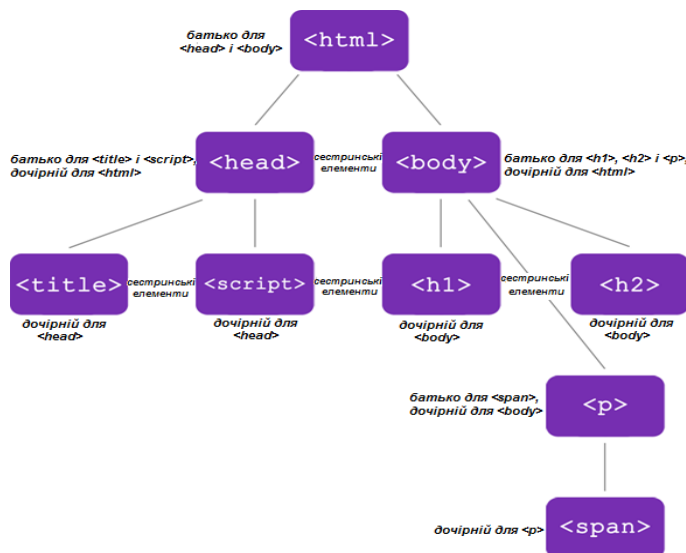
А якщо ми хочемо підкреслити посилання тільки в межах тегу `<p>`, можна використати селектори нащадків і записати:

p a {text-decoration: underline;}

Тут ми говоримо, що всі посилання `<a>`, які знаходяться в тегах `<p>`, повинні бути підкреслені. А на всі інші посилання це правило не поширюється. Це використовується для зміни стилю, використовуючи вкладеність тегів та їхні родинні зв'язки.

Ієрархія HTML-тегів включає теги предків, нащадків, батьків, дочірніх тегів та сестринських тегів.

Наприклад, тег **head** та **body** є нащадками тегу **html**; **h1**, **h2** та **p** є нащадками тегу **body**; **span** є нащадком для тегів **p**, **body** та **html**. Сусідніми тегами називаються елементи, у яких є спільний батько, наприклад, **head** і **body**.



Ієрархія HTML-тегів: родинні зв'язки

При створенні селектора нащадків спочатку потрібно записати назву батьківського елемента, а потім – назву дочірнього елемента.

Наприклад, якщо ми хочемо застосувати стилі до всіх посилань, які знаходяться всередині тега **<p>**, який знаходиться всередині тега **<body>**, який знаходиться всередині тега **<html>**, то можна записати селектор нащадків такими способами:

```
html body p a {};  
body p a {};  
p a {};
```

Всі три варіанти працюватимуть, але в даному випадку можна обмежитися лише ім'ям батька, тобто записати тільки один ідентифікатор, який вказує на тег **<p>**.

Також можна посилатися не тільки на назву батьківського тега, але також і на клас або ідентифікатор, які були присвоєні батьківському тегу. Це дає ще більш гнучкі можливості для написання стилів.

Для того, щоб застосувати стилі до посилань всередині елементів з класом "blogcontent", можна використати наступний код:

```
/* для класів */  
.blogcontent a {  
    color: blue;  
}  
  
/* для ідентифікаторів */  
#mobilenav a {  
    font-size: 12px;  
}
```

Псевдокласи і псевдоелементи

Зазвичай в стилі CSS задаються правила для елементів веб-сторінки, які видно в її коді. Але іноді потрібно змінити вигляд елемента в певний момент, наприклад, коли на нього наводять курсор. Також іноді потрібно

стилізувати елементи, які не є конкретно визначеними в кодї сторінки, наприклад, перший символ в абзаці.

Щоб змінювати стиль елементів в певний момент, в CSS використовуються псевдокласи. Вони дозволяють задавати стилі для елементів, коли з ними відбуваються певні дії, наприклад, коли на них наводять курсор.

Псевдоелементи, на відміну від псевдокласів, можуть задавати стилі для неіснуючого вмісту або для елементів, які не є конкретно визначеними в кодї сторінки, наприклад, для першого символу в абзаці.

Популярні псевдокласи CSS

Можна надавати елементам стиль, який буде залежати від стану цих елементів. Список деяких псевдокласів:

- **:link** – задає стиль посилань, за якими користувач ще не перейшов;
- **:visited** – застосовує стиль до вже відвіданих посилань;
- **:hover** – визначає стиль елемента, коли на нього наведений курсор (може застосовуватися не тільки до посилань);
- **:active** – задає стиль активним посиланням (тобто, в момент кліка по ній);
- **:focus** – застосовує стиль до елемента при фокусуванні на ньому (наприклад, при установці курсору в рядок пошуку);
- **:not()** – цей корисний псевдоклас дозволяє вибрати і стилізувати тільки ті елементи, які не містять селектор, вказаний в дужках.

Приклад:

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

```
a:hover {  
  color: #26A65B;  
}
```

```
a:visited {  
  color: #CCC;  
}
```

У записаному стилі сказано, що звичайні, невідвідані посилання повинні бути червоного кольору: **link {color: red;}**, посилання при наведенні повинна міняти колір на інший **a: hover {color: # 26A65B;}**, а відвідана посилання повинна мати третій кольору: **visited {color: #CCC;}**.

Популярні псевдоелементи CSS

Якщо псевдокласи записуються з одним двокрапкою, то псевдоелементи – з двома. Це було впроваджено в CSS3 для того, щоб розрізняти псевдокласи і псевдоелементи між собою. Однак раніше цієї різниці не існувало і з псевдоелементами використовувалося одне двокрапка. Зараз браузері підтримують обидва варіанти написання (але не для всіх випадків). Розглянемо деякі псевдоелементи:

- **::after** – використовується разом з властивістю **content** і дозволяє вивести необхідні дані після вмісту елемента;
- **::before** – виконує схожу функцію, що і попередній, тільки виводить дані перед вмістом елемента;
- **::selection** – цей псевдоелемент розпізнається браузерами тільки при використанні двох двокрапок і дозволяє встановити колір і фон для тексту, який виділений Користувачем;
- **::first-letter** – використовується для зміни стилю першого символу в тексті елемента;
- **::first-line** – використовується для зміни стилю першого рядка тексту елемента.

Приклад:

```
blockquote::before {  
  content: "«";
```

```

}

blockquote::after {
  content: "»";
}

blockquote::selection {
  color: #C8F7C5;
  background-color: #1E824C;
}

```

Тут заданий стиль для довгих цитат, який додає лапки «ялинки» на початку і у кінці вмісту тега **<blockquote>**, а також змінює колір і фон вибраного користувачем тексту цитати.

Селектори атрибутів

В HTML багато тегів можуть створювати різні елементи, залежно від того, які атрибути в них використовуються. Наприклад, тег **<input>** може створювати кнопку або текстове поле, залежно від того, який атрибут `type` ми використовуємо. Якщо ми застосуємо стиль до селектору **INPUT**, то цей стиль буде застосовуватися до всіх створених за допомогою цього тега елементів.

Щоб бути гнучким у керуванні стилем подібних елементів, в CSS існують селектори атрибутів. Вони дозволяють встановити стиль за наявністю певного атрибуту тега або його значення. Наприклад, ми можемо встановити стиль для всіх елементів **<input>**, які мають атрибут **type="text"**, або для тих, які мають атрибут **type="submit"**. Це дає нам більше можливостей для керування стилем нашої веб-сторінки.

Простий селектор атрибута

Цей метод дозволяє встановити стиль для елемента, якщо в нього доданий певний атрибут тега, і незалежно від його значення. Синтаксис використання такого селектора виглядає так:

[атрибут] {опис правил стилю} Селектор [атрибут] {опис правил стилю}

Цей стиль застосовується до всіх тегів, в яких є зазначений атрибут. Пробіл між ім'ям селектора та квадратними дужками не допускається.

Наприклад, у цьому прикладі показано, як змінити стиль тега <q>, якщо до нього додано атрибут "title":

Приклад. Вид елемента в залежності від його атрибута

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Селектори атрибутів</title>
    <style>
      Q {
        font-style: italic; /* Курсивне накреслення */
        quotes: "\00AB" "\00BB"; /* Міняємо вид лапок в цитаті */
      }
      Q[title] {
        color: maroon; /* Колір тексту */
      }
    </style>
  </head>
  <body>
    <p> Продовжуючи відомий закон Мерфі, який говорить: <q>Якщо неприємність
      може статися, то вона обов'язково трапиться </q>, можемо ввести своє спостереження: <q
      title="Із законів Фергюсона-Мержевича"> Після того, як веб-сторінка буде коректно
      відображатися в одному браузері, з'ясується, що вона неправильно показується в іншому
    </q>.</p>
  </body>
</html>
```

Продовжуючи відомий закон Мерфі, який говорить: *«Якщо неприємність може статися, то вона обов'язково трапиться»*, можемо ввести своє спостереження: *«Після того, як веб-сторінка буде коректно відображатися в одному браузері, з'ясується, що вона неправильно показується в іншому»*.

Зміна стилю елемента в залежності від застосування атрибута title

У цьому прикладі змінюється колір тексту всередині контейнера **<q>**, коли до нього додається **title**. Зверніть увагу, що для селектора **Q[title]** немає потреби повторювати стильові властивості, оскільки вони успадковуються від селектора **Q**.

Атрибут зі значенням

Цей спосіб дозволяє встановити стиль для елемента лише у тому випадку, якщо в нього задано певне значення атрибута. Синтаксис застосування такого селектора наступний: [атрибут="значення"] {Опис правил стилю} Селектор[атрибут="значення"] {Опис правил стилю}

Якщо вказано лише назву атрибута в квадратних дужках, то стиль буде застосований до всіх тегів, в яких є цей атрибут. А якщо додати до нього значення через знак дорівнювання, то стиль буде застосовуватися тільки до тегів, в яких атрибут має вказане значення.

У прикладі показано, як змінити стиль посилання (тег **<a>**), якщо в нього задано атрибут **target** зі значенням **_blank**. При цьому посилання буде відкриватися в новому вікні, і щоб це показати, перед текстом посилання додано малюнок за допомогою стилів.

Приклад. Стиль для відкриття посилань у новому вікні

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>Селектори атрибутів</title>
  <style>
    A[target="_blank"] {
```

```

background: url(images/blank.png) 0 6px no-repeat; /* Параметри фонового рисунку */
padding-left: 15px; /* Зміщуємо текст вправо */
}
</style>
</head>
<body>
  <p><a href="1.html"> Звичайне посилання </a> |
  <a href="link2" target="_blank"> Посилання в новому вікні </a></p>
</body>
</html>

```

[Звичайне посилання](#) | [🔗 Посилання в новому вікні](#)

Зміна стилю елемента в залежності від значення target

У цьому прикладі ми додаємо рисунок як фон для посилання за допомогою властивості background.

Функція цієї властивості полягає в тому, щоб створити повторювану фонову картинку, яку можна побачити більше, ніж раз, на всій площині елемента. Але якщо нам не потрібне повторення картинки, ми можемо скасувати його, задавши значення no-repeat. В результаті ми отримаємо єдине зображення, яке буде використовуватися як фон для елемента.

Значення атрибута починається з певного тексту

У цьому прикладі ми хочемо змінити стиль звичайних посилань та зовнішніх посилань, які ведуть на інші сайти. Ми можемо використати селектори атрибутів, щоб знайти ті посилання, які починаються з ключового слова http://. Для цього ми можемо використати синтаксис [атрибут^="значення"], де ^ показує, що ми шукаємо атрибут, який починається з певного значення.

Наприклад, якщо ми хочемо змінити стиль посилання, яке веде на інший сайт, ми можемо використати селектор **a[href^="http://"]**, щоб

знайти всі посилання, які починаються з `http://` в атрибуті `href`. Якщо ми хочемо змінити стиль всіх посилань, які містять слово "зовнішній" в атрибуті `title`, ми можемо використати селектор `[title^="зовнішній"]`. А, щоб знайти всі посилання, у яких атрибут `title` починається зі слова "зовнішній".

Таким чином, ми можемо використати селектори атрибутів, щоб знайти ті елементи, які містять певне значення в атрибуті, та змінити їх стиль за допомогою CSS правил.

Приклад. Зміна стилю зовнішнього посилання

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title>Селектори атрибутів</title>
  <style>
    A[href^="http://"] {
      font-weight: bold /* Жирне накреслення */
    }
  </style>
</head>
<body>
  <p><a href="1.html"> Звичайне посилання </a> |
  <a href="http://htmlbook.ru" target="_blank">Зовнішнє посилання на сайт
htmlbook.ru</a></p>
</body>
</html>
```

[Звичайне посилання](#) | [Зовнішнє посилання на сайт htmlbook.ru](http://htmlbook.ru)

Зміна стилю для зовнішніх посилань

У цьому прикладі ми використовуємо жирний шрифт для зовнішніх посилань та можемо також додати маленьку іконку, яка показує, що посилання веде на інший сайт. Ми використовуємо значення атрибуту, яке

закінчується певним текстом, щоб застосувати стиль до елементів з таким значенням атрибуту.

Ось синтаксис: **[атрибут\$="значення"] {Опис правил стилю}**
Селектор[атрибут\$="значення"] {Опис правил стилю}

Якщо ми хочемо застосувати стиль до всіх елементів з атрибутом, який закінчується заданим текстом, то використовуємо перший варіант, а якщо хочемо застосувати стиль тільки до певних селекторів, то використовуємо другий варіант.

Можна використовувати цей спосіб, щоб автоматично розділяти стиль для посилань на сайти домену **"ua"** та для посилань на сайти інших доменів, наприклад **"com"**.

Приклад. Стиль для різних доменів

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Селектори атрибутів</title>
<style>
A[href$=".ua"] { /* Якщо посилання закінчується на .ua */
background: url(images/ua.png) no-repeat 0 6px; /* Додаємо фоновий рисунок */
padding-left: 12px; /* Зміщуємо текст вправо */
}
A[href$=".com"] { /* Якщо посилання закінчується на .com */
background: url(images/com.png) no-repeat 0 6px;
padding-left: 12px;
}
</style>
</head>
<body>

<p><a href="http://www.yandex.com">Yandex.Com</a> |
<a href="http://www.yandex.ru">Yandex.ua</a></p>
</body>
</html>
```

У цьому прикладі є два посилання на різні домени: **com** і **ua**. Для кожного посилання створено свою фонову картинку за допомогою стилів. Стили будуть застосовуватися тільки для посилань, атрибут href яких закінчується на **".ua"** або **".com"**. Якщо додати слеш або адресу сторінки до імені домену, то закінчення зміниться, і стилі вже не будуть застосовуватися. У цьому випадку краще використати селектор, у якому зазначений текст з'являється у будь-якому місці значення атрибута.

 [Yandex.Com](#) |  [Yandex.ua](#)

Додавання картинки до посилань

Значення атрибута містить вказаний текст

Уявіть, що у веб-сторінці є посилання на інші сторінки, але вам потрібно змінити вигляд тільки для тих посилань, які містять певний текст. Але не відомо, в якому місці текст знаходиться – на початку, середині чи кінці посилання. В такому випадку потрібно використовувати наступний синтаксис:

[атрибут*="значення"] {Опис правил стилю}

Селектор[атрибут*="значення"] {Опис правил стилю}

Наприклад, у прикладі показано, як змінити стиль для посилань, які містять слово «**htmlbook**» в атрибуті **href**. Таким чином, будь-яке посилання на сторінку з цим словом в адресі матиме змінений стиль.

Приклад. Стиль для різних сайтів

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

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

```
<title>Селектори атрибутів</title>
```

```

<style>
[ href*="htmlbook" ] {
  background: yellow; /* Жовтий колір фону */
}
</style>
</head>
<body>
<p><a href="http://www.htmlbook.ru/html/">Теги HTML</a> |
<a href="http://stepbystep.htmlbook.ru">Крок за кроком </a> |
<a href="http://webimg.ru">Графіка для Веб</a></p>
</body>
</html>

```

[Теги HTML](#) | [Крок за кроком](#) | [Графіка для Веб](#)

Зміна стилю для посилань, в адресі яких зустрічається «htmlbook»

Одне з декількох значень атрибута

Іноді значення атрибутів можуть бути перераховані через пробіл, наприклад, якщо ми додаємо кілька класів до елемента. Щоб задати стиль для елемента, який має певне значення атрибуту серед перерахованих, використовують такий синтаксис:

[атрибут~="значення"] {Опис правил стилю}

Селектор[атрибут~="значення"] {Опис правил стилю}

Стиль буде застосовано, якщо у значенні атрибуту присутнє потрібне значення або воно входить до списку значень, розділених пробілами.

Приклад. Стиль в залежності від імені класу

```

<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Блок</title>
<style>
[class~="block"] h3 { color: green; }

```

```

</style>
</head>
<body>
  <div class="block tag">
    <h3>Заголовок</h3>
  </div>
</body>
</html>

```

В даному прикладі зелений колір тексту застосовується до селектору **H3**, якщо ім'я класу у шару задано як **block**. Відзначимо, що аналогічний результат можна отримати, якщо використовувати конструкцію ***=** замість **~=**.

Дефіс у значенні атрибута

Дозволено використовувати символ дефісу (-) в іменах класів та ідентифікаторів для створення значущих назв. Щоб змінити стиль елементів з такими назвами, використовується наступний синтаксис:

[атрибут]="значення" {Опис правил стилю}

Селектор[атрибут]="значення" {Опис правил стилю}

Стиль застосовується до елементів, у яких атрибут починається з вказаного значення або з фрагмента значення, після якого йде дефіс. Наприклад, якщо потрібно застосувати стиль до елемента з класом **"example-class"**, то використовується селектор **".example-class"** або **"[class='example']"**.

Приклад. Дефіси в значеннях

```

<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Блок</title>
  <style>

```



```

DIV[class="block"] {
  background: #306589; /* Колір фону */
  color: #acdb4c; /* Колір тексту */
  padding: 5px; /* Поля */
}
DIV[class="block"] A {
  color: #fff; /* Колір посилань */
}
</style>
</head>
<body>
<div class="block-menu-therm">
<h2>Терміни</h2>
<div class="content">
  <ul class="menu">
    <li><a href="t1.html">Буквиця</a></li>
    <li><a href="t2.html">Виворіт </a></li>
    <li><a href="t3.html">Виключення </a></li>
    <li><a href="t4.html">Інтерліньяж</a></li>
    <li><a href="t5.html">Капітель</a></li>
    <li><a href="t6.html">Накреслення</a></li>
    <li><a href="t7.html">Відбиття </a></li>
  </ul>
</div>
</div>
</body>
</html>

```

У цьому прикладі для імені класу використано назву **"block-menu-therm"**. Щоб змінити стиль для елементів, які мають таке значення атрибута, використовується конструкція **l="block"**, оскільки значення починається зі слова **"block"** і містить дефіси.

Можна поєднувати різні методи, щоб задавати стиль для елементів, які мають два або більше атрибути. У таких випадках квадратні дужки йдуть підряд, наприклад:

**[атрибут1="значення1"][атрибут2="значення2"] {Опис правил
стилю} Селектор[атрибут1="значення1"][атрибут2="значення2"]
{Опис правил стилю}**