

# Internet- технології

Каскадні таблиці стилів.  
Селектори

# Селектори

Селектор визначає, для яких елементів використовуються набір стильових властивостей та їх значень.

Комбінації селекторів:

- **вкладені** селектори;
- **дочірні** селектори;
- **сусідські** селектори;
- **родинні** селектори.

Вид селектора	Приклад	Приклад стиля
універсальний	*	* { color:green; }
селектор тегів	div	div { margin:7px; padding:7px; }
селектор атрибуту	q[title]	q[title] { color: maroon; }
селектор класу	.important	.important { font-weight:bold; color:red; }
селектор id	#onlyThisOne	#onlyThisOne { font-family:courier; }
псевдоклас	a:link	a:link { color:blue; }
псевдоелемент	p:first-letter	p:first-letter { color:red }

# Селектори атрибутів

Використання селектора атрибута (загальний синтаксис – Селектор[атрибут] { Опис стильових властивостей }, пробіл між ім'ям селектора та квадратними дужками не допускається) дозволяють встановити стиль для елементів, для яких визначено атрибут (або за його значенням).

```
<style>
  a[target="_blank"] {
    background: url(image/blank.png) no-repeat;
    padding-left: 15px;
  }
</style>
```

```
<body>
```

```
<p><a href="1.html"> Звичайне посилання </a> <br>
```

```
<a href="2.html" target="_blank"> Посилання для відкриття у новій вкладці
```

```
</a></p>
```

```
</body>
```

Звичайне посилання

 Посилання для відкриття у новій вкладці

# Селектори атрибутів

Селектор атрибутів дозволяє скоротити кількість ID та класів та зробити верстку легшою для підтримки. Крім простого селектора атрибутів, CSS доступні такі різновиди:

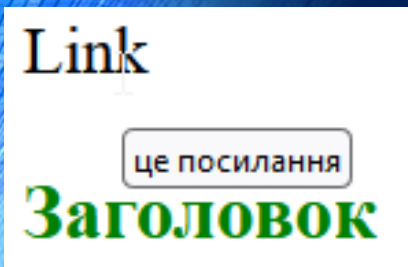
**[att^="value"]** – вибір елементів за атрибутом, значення якого **починається** із заданого рядку.

**[att]="value"** – вибір елементів за атрибутом, значення якого **або збігається** з зазначеним value, **або починається** з value, за яким слідує дефіс (-)

**[att\$="value"]** – за атрибутом, що **закінчується** заданим значенням.

**[att\*="value"]** – за атрибутом, в якому міститься задане значення.

**[att ~="value"]** – за атрибутом, у якого серед **перелічених** значень є збіги з необхідним.



```
a[title$="посилання"] { display:
block; }
[class~="block"] h3 { color: green; }
```

```
<a title="це посилання">Link</a>
<div class="block tag">
  <h3>Заголовок</h3>
</div>
```



# Селектори. Класи

Щоб віднести елемент розмітки до того чи іншого класу, потрібно скористатися його атрибутом `class`. Це дозволить розробнику визначити клас елементів розмітки (множину елементів сторінки), які відображатимуться однаково.

Імена класів повинні починатися з латинського символу і можуть містити символ дефісу (-) і підкреслення (\_). Використання не-латинських букв у іменах класів неприпустимо.

```
.important {  
  font-weight:bold; color:red;  
}
```

```
<p class="important"> Important Text</p>  
<p> Ordinary Text </p>  
<div class="important"> Important </div>
```

**Important Text**

Ordinary Text

**Important**

# Селектори. Класи. Приклад

```
<head>
  <style>
    div {
      writing-mode:tb-rl; /* Окремі стилі для IE */
      font-size: 196%;
    }
    .vertical {
      width: 1em;   line-height: 1em;
    }
  </style>
</head>
<body>
  <div style="text-indent:0px" class="vertical">
В е р т и к а л ь </div>
</body>
```

В  
е  
р  
т  
и  
к  
а  
л  
ь

# Селектори. Ідентифікатори

Селектори id (ідентифікатори, #<ім'я\_ідентифікатора> { переік стильвоих властивостей }), застосовуються для елементів з атрибутом id та відповідним його значенням. Основна відмінність класів від ідентифікаторів полягає в тому, що імена других повинні бути унікальними, не повторюватися, що дозволяє їх використовувати разом з JavaScript.

```
div#First {  
    background-color: silver;  
}
```

```
<div id="First">Ідентифікатори  
широко використовуються в  
JavaScript, щоб знайти унікальний  
елемент у документі.</div>
```

Ідентифікатори широко використовуються в JavaScript, щоб знайти унікальний елемент у документі.

# Селектори. Псевдокласи

**:checked** – застосовується до елементів інтерфейсу, таких як прапорці (checkbox) та перемикачі (radio), коли вони перебувають у положенні «ввімкнено».

**:indeterminate** – задає стиль для елементів форм, таких як прапорці та перемикачі, коли вони перебувають у невизначеному стані.

**:disabled** – застосовує стиль до **заблокованих** елементів форм.

**:enabled** – використовується для застосування стилю до доступних до редагування користувачу (не заблокованих) елементів форм.

**:required** – застосовує стильові правила до елемента input, у якого встановлено атрибут required.

**:optional** – застосовує стильові правила до елементів форми, у яких не заданий атрибут required.

**:invalid** – застосовується до полів форми, вміст яких не відповідає зазначеному типу.



# Селектори. Псевдокласи

**:empty** – усі порожні елементи, тобто ті, які не містять дочірніх елементів, тексту або пропусків.

**:not** – задає правила стилів для елементів, які не містять зазначеного селектора. `input:not([type="submit"])`

**:first-of-type** – задає правила стилів для першого елемента у списку дочірніх елементів свого батька.

**:nth-of-type** – використовується для додавання стилю до елементів зазначеного типу на основі нумерації у дереві елементів.

**:nth-last-child** – використовується для додавання стилю до елементів на основі порядку у дереві елементів.

**:first-child** – застосовує стильове оформлення до першого дочірнього елемента свого батька.

**:only-child** – застосовується до дочірніх елементів, тільки якщо він **єдиний** у батька.

# Селектори. Псевдокласи. Приклад

CSS:

```
table {  
  border-spacing: 0;  
}  
td {  
  border: 1px solid #333;  
  padding: 3px;  
}  
td:nth-last-child(2n+1) {  
  background: #fofofo;  
}  
td:nth-child(1) {  
  background: #cfc;  
}
```

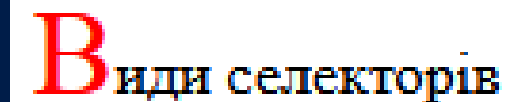
```
<table>  
  <tr> <td>&nbsp;</td><td>ITex</td><td>СПЗ</td><td>КСх</td></tr>  
  <tr> <td>Іванов</td><td>79</td><td>90</td><td>75</td> </tr>  
  <tr> <td>Петров</td><td>60</td><td>76</td><td>90</td> </tr>  
  <tr> <td>Сідоров</td><td>79</td><td>95</td><td>78</td> </tr>  
</table>
```

	ITex	СПЗ	КСх
Іванов	79	90	75
Петров	60	76	90
Сідоров	79	95	78

# Селектори. Псевдоелементи

Псевдоелементи дозволяють встановити стиль елементів, **не визначених у дереві елементів**, а також **генерувати вміст**, якого немає у вихідному коді тексту. У CSS3 псевдоелементи позначаються з двома двокрапками, щоб синтаксис відрізнявся від визначення псевдокласів (на відміну від попередньої версії, де визначалось просто двокрапкою).

```
<head>
  <style>
    p::first-letter {
      font-family: "Times New Roman", Times, serif;
      font-size: 2em;
      color: red;
    }
  </style>
</head>
<body><p> Види селекторів </p></body>
```



# Селектори. Псевдоелементи. Приклад

Псевдоелемент `::after` використовується разом із властивістю `content`, дозволяє генерувати вміст (успадковує стиль від елемента, якого він додається).

Псевдоелемент `::selection` застосовує стиль до виділеного користувачем тексту. У правилах стилів **для цього псевдоелемента** допускається використовувати такі властивості: `color`, `background`, `background-color`, `text-shadow`.

```
p::selection {  
    color: #ff0;  
    background: #000;  
}  
.cite::after {  
    font-style: italic;  
    content: "(див. розділ 4.2)";  
}
```

При виділенні цього тексту він **змінить колір**.

20-й Ювілейний міжнародний молодіжний форум(див. розділ 4.2)

```
<body><p> При виділенні цього тексту він змінить  
колір.</p>20-й Ювілейний міжнародний молодіжний <span  
class="cite">форум</span></body>
```



# Селектори. Псевдоелементи. Приклад

```
<ol>
  <li>Іванов</li>
  <li>Петров</li>
  <li>Сідоров</li>
</ol>
```

1) Іванов  
2) Петров  
3) Сідоров

CSS:

```
ol {
  list-style-type: none;
  counter-reset: item;
}
li:before {
  content: counter(item) ')';
  counter-increment: item;
}
```

# Селектори. Приклад. Нумерація рядків таблиці

```
<style>
```

```
table {
```

```
  counter-reset: cnt;
```

```
}
```

```
tr {
```

```
  counter-increment: cnt;
```

```
}
```

```
.trnumber::before {
```

```
  content: counter(cnt); //"."
```

```
}
```

```
</style>
```

```
<table border="1" class="mytable">
```

```
  <tr><td><span
```

```
class="trnumber">.</span></td><td>1.2</td></tr>
```

```
  <tr><td><span
```

```
class="trnumber">.</span></td><td>2.2</td></tr>
```

```
  <tr><td><span
```

```
class="trnumber">.</span></td><td>2.3</td></tr>
```

```
</table>
```

Або, з використанням JavaScript:

```
var firstcells = document.querySelectorAll('.mytable tr td:first-child');
```

```
for (i = 0; i < firstcells.length; i++) firstcells[i].innerHTML = i + 1 + ".";
```

1.	1.2
2.	2.2
3.	2.3

# Комбінація селекторів. *Вкладені* селектори

Допускається довільний рівень вкладення. Слід окремо розрізняти **дочірні** селектори, коли елемент є **безпосереднім** нащадком.

```
p b {  
  font-weight: bold;  
  color: red;  
}  
  
p > b {  
  color: green;  
}
```

**Текст без впливу стилю**

**Демонстрація використання вложеного селектора**

**Демонстрація використання дочірнього селектора**

<div><b>Текст без впливу стилю</b></div>

<p><span><b>Демонстрація використання **вложеного** селектора</b></span></p>

<p><b>Демонстрація використання **дочірнього** селектора</b></p>

# Комбінація селекторів. Родинні селектори

Сусідніми (+) називаються елементи веб-сторінки, які йдуть безпосередньо один за одним у коді документа.

Родинні селектори (~) застосовуються до дочірніх елементів одного елемента-батька.

```
h1 ~ p { color: red; }  
h1 + p { color: green; }
```

## Заголовок

Сосідний селектор (абзац одразу після заголовка, `h1 + p`).

Просто абзац.

Родинний селектор (абзац з заголовком мають безпосереднього спільного предка, `h1 ~ p`).

```
<h1>Заголовок</h1>
```

```
<p>Сосідний селектор (абзац одразу після заголовка, h1 + p).</p>
```

```
<div><p>Просто абзац.</p></div>
```

```
<p>Родинний селектор (абзац з заголовком мають безпосереднього спільного предка, h1 ~ p).</p>
```



# *Контрольні запитання*

- Що таке селектори? Види селекторів.
- Різновиди селекторів атрибутів.
- У чому різниця між застосуванням селекторів класу та селектора ідентифікатора?
- Що таке псевдоеlementи?