

СИНТАКСИС CSS

План:

1. Формат стильового запису.
2. Правила застосування стилів.
3. Значення стильових властивостей.
4. Ключові слова.
5. Селектори тегів.
6. Класи.

Стилі в HTML записуються в окремому форматі, відмінному від HTML. Головним поняттям є селектор, який визначає назву стилю та параметри форматування, які застосовуються до вибраних елементів. Селектором можуть бути теги, класи або ідентифікатори. Загальна форма запису стилів має такий вигляд: спочатку пишеться ім'я селектора, потім в фігурних дужках записуються стильові властивості та їх значення, які розділяються крапкою з комою.

<u>селектор</u>	<u>властивість</u>	<u>значення</u>
body	{ background:	#ffc910; }

При цьому CSS не чутливий до регістру, перенесення рядків, пропусків і символів табуляції, тому форма запису залежить від бажання розробника.

Наприклад, у HTML документі можна застосувати стилі до заголовків. Для цього потрібно використати тег **<style>** і написати потрібні стилі в ньому. Наприклад, можна змінити колір тексту та товщину шрифту для

заголовків різних рівнів. Код може мати різні форми запису, але головне, щоб стилі були зрозумілі для розробника і працювали правильно.

Приклад. Використання стилів

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

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

```
<title>Заголовки</title>
```

```
<style>
```

```
h1 { color: #a6780a; font-weight: normal; }
```

```
h2 {
```

```
color: olive;
```

```
border-bottom: 2px solid black;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Заголовок 1</h1>
```

```
<h2>Заголовок 2</h2>
```

```
</body>
```

```
</html>
```

В даному прикладі властивості селектора **h1** записані в один рядок, а для селектора **h2**. Кожна властивість знаходиться на окремому рядку. У другому випадку легше відшукувати потрібні властивості і правити їх за необхідності, але при цьому незначно зростає обсяг даних за рахунок активного використання пропусків і переносів рядків. Так що в будь-якому випадку спосіб оформлення стильових параметрів залежить від розробника.

Правила застосування стилів

Форма запису

Для оформлення веб-сторінки можна використовувати стилі, які задають вигляд різних елементів. Щоб задати стилі для елемента на сторінці, можна використовувати селектор – це така "мітка", за якою можна знайти потрібний елемент. Наприклад, якщо потрібно змінити колір тексту у всіх абзацах (тег **<p>**), можна використовувати такий селектор: **p**.

Існує декілька способів задавати стилі для селекторів. Один із них – додавати кожну властивість окремо, як у цьому прикладі:

```
td {background: olive;}
```

```
td {color: white;}
```

```
td {border: 1px solid black;}
```

Проблема цього підходу в тому, що потрібно повторювати селектор кілька разів і це може бути заплутано. Тому краще писати всі властивості для кожного селектора разом, використовуючи таку форму запису:

```
td {  
  
    background: olive;  
  
    color: white;  
  
    border: 1px solid black;  
  
}
```

Цей спосіб є більш зручним і наочним.

Також варто пам'ятати, що якщо використати один і той самий селектор з різними властивостями, то ті властивості, які були вказані нижче в коді, матимуть пріоритет. Наприклад:

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

p {color: red;}

У цьому прикладі колір тексту у всіх абзацах (тег **<p>**) спочатку встановлено зеленим, а потім червоним. Тому, оскільки значення **"red"** розташоване нижче, воно буде застосовано до тексту.

Проте, такі повторювані значення краще уникати, або ж видаляти. Наприклад, якщо треба встановити однаковий колір для всіх абзаців на сторінці, то варто написати:

p {color: green;}

і не дублювати це правило в коді ще раз.

Значення

У кожної властивості може бути тільки відповідне її функції значення. Наприклад, для **color**, яка встановлює колір тексту, як значення неприпустимо використовувати числа.

Коментарі

Коментарі в CSS-файлі допомагають пояснювати використання стильових властивостей, виділяти розділи або писати свої нотатки. Коментарі дозволяють легко згадувати логіку і структуру селекторів, і підвищують чіткість коду. Щоб помітити, що текст є коментарем, застосовують наступну конструкцію **/*...*/**.

Ось приклад коду з коментарем:

/*

Стиль для сайта

Зроблений для ознайомлювальних цілей

****/***

```
div {  
    width: 200px; /* Ширина блоку /  
    margin: 10px; / Поля навколо елемента /  
    float: left; / Обтікання по правому краю */  
}
```

Коментарі можна додавати в будь-яке місце CSS-документа, а також писати текст коментаря в кілька рядків. Проте додавання тексту збільшує обсяг документів, що негативно позначається на часі їх завантаження. Тому коментарі зазвичай застосовують в налагоджувальних або навчальних цілях, а при викладанні сайту в мережу їх видаляють. Не можна вкладати коментарі один в одного.

Значення стильових властивостей

Значення стильових властивостей – це конкретні значення, які встановлюються для властивостей CSS. Значення можуть бути різних типів, таких як рядок, число або відсотки.

Рядок – це послідовність символів, яку можна використовувати в CSS для властивостей, таких як font-family або background-color. Приклад:

```
h1 {  
    font-family: "Helvetica Neue", Arial, sans-serif;  
    background-color: white;  
}
```

Число – це числове значення, яке можна використовувати в CSS для властивостей, таких як width, height або font-size. Приклад:

```
img {  
    width: 100px;  
    height: 100px;  
    font-size: 16px;  
}
```

Відсотки – це значення, яке вказує відносний розмір відносно батьківського елемента. Наприклад, якщо ви встановлюєте ширину елемента в 50%, то він займе половину ширини свого батьківського елемента. Приклад:

```
.container {  
    width: 80%;  
}  
  
.box {  
    width: 50%;  
}
```

У цьому прикладі елемент з класом "box" займе 50% ширини свого батьківського елемента з класом "container", який має ширину 80%. Таким чином, ширина "box" буде 40% від загальної ширини сторінки.

Знання різних типів значень стильових властивостей допоможуть вам краще розуміти, як працює CSS і як ви можете контролювати вигляд веб-сторінок.

Розміри

Для того, щоб встановити розміри елементів на сторінці веб-сайту, в CSS використовуються різні одиниці виміру. Вони бувають абсолютними та відносними. Абсолютні одиниці не залежать від розміру екрану або пристрою, а відносні одиниці визначають розмір елемента відносно іншого елемента.

Наприклад, відносні одиниці зазвичай використовуються для роботи з текстом або коли потрібно встановити відсоткове співвідношення між елементами.

Одиниці виміру, що відносяться до відносних, мають такі назви та опис:

Відносні одиниці виміру

Одиниц а	Опис
em	Розмір шрифту поточного елемента
ex	Висота символу x
px	Піксель
%	Відсоток

Одиниця **em** це змінне значення, що залежить від розміру шрифту поточного елемента (розмір встановлюється через стильову властивість **font-size**). У кожному браузері закладений розмір тексту, застосовуваний в тому випадку, коли цей розмір явно не заданий. Тому спочатку **1em** дорівнює розміру шрифту, заданого в браузері за замовчуванням або

розміром шрифту батьківського елемента. Процентна запис ідентична **em**, в тому сенсі, що значення **1em** і **100 %** рівні.

Одиниця **ex** визначається як висота символу «х» в нижньому регістрі. На **ex** поширюються ті ж правила, що і для **em**, а саме, він прив'язаний до розміру шрифту, вказаного в браузері за замовчуванням, або до розміру шрифту батьківського елемента.

Піксель – це елементарна точка, що відображається монітором або іншим подібним пристроєм, наприклад, смартфоном. Розмір пікселя залежить від роздільної здатності пристрою і його технічних характеристик.

Приклад. Використання відносних одиниць

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Відносні одиниці</title>
    <style>
      H1 { font-size: 30px; }
      P { font-size: 1.5em; }
    </style>
  </head>
  <body>
    <h1>Заголовок розміром 30 пікселів</h1>
    <p>Розмір тексту 1.5 em</p>
  </body>
</html>
```

Заголовок розміром 30 пікселів

Розмір тексту 1.5 em

Розмір тексту при різних одиницях

Абсолютні одиниці

Абсолютні одиниці застосовуються рідше, ніж відносні і зазвичай при роботі з текстом.

Абсолютні одиниці виміру

Одиниця	Опис
in	Дюйм (1 дюйм рівне 2,54 см)
cm	Сантиметр
mm	Міліметр
pt	Пункт (1 пункт рівне 1/72 дюйма)
pc	Пика (1 пика рівна 12 пунктам)

Найбільш, мабуть, поширеною одиницею є пункт, який використовується для вказівки розміру шрифту. Хоча ми звикли вимірювати все в міліметрах і подібних одиницях, пункт, мабуть, єдина величина з не метричної системи вимірювання, яка використовується у нас повсюдно.

Приклад. Використання абсолютних одиниць

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>Абсолютні одиниці</title>
```

```
<style>
```

```
  H1 { font-size: 24pt; }
```

```
  P { margin-left: 30mm; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <h1>Заголовок розміром 24 пункта</h1>
```

```
  <p> Зсув тексту вправо на 30 міліметрів </p>
```

```
</body>
```

```
</html>
```

Заголовок розміром 24 пункта

Зсув тексту вправо на 30 міліметрів

Размір текста при різних одиницях

При установці розмірів обов'язково вказуйте одиниці виміру, наприклад **width: 30px**. В іншому випадку браузер не зможе показати бажаний результат, оскільки не розуміє, який розмір вам потрібно. Одиниці не додаються тільки при нульовому значенні (**margin: 0**).

Колір

Колір в стилях можна вибрати на трьох різних способах: за шістнадцятковим значенням, за назвою і в форматі RGB.

За шістнадцятковим значенням:

Колір можна задати за допомогою чисел в шістнадцятковому коді. Щоб уникнути плутанини у визначенні системи числення, перед шістнадцятковим числом ставлять символ решітки #, наприклад, #666999. Кожен з трьох кольорів – червоний, зелений і синій – може мати значення від 00 до FF. Таким чином, позначення кольору складається з трьох складових: #rrggbb, де перші два символи позначають червону компоненту кольору, два середні – зелену, а два останні – синю.

Якщо числа для кольору використовують лише одну цифру, то їх можна подвоїти. Наприклад, #fe0 розцінюється як #fee00.

За назвою:

деякі кольори мають назви, які браузер може розпізнати. Наприклад, "червоний" або "синій".

За допомогою RGB:

Колір можна задати в десятковому численні червоної, зеленої і синьої складових. Кожен з трьох кольорів може мати значення від 0 до 255. Також можна вказати значення кольору в відсотках. Для цього потрібно вказати

ключове слово "rgb", а потім в дужках, через кому, вказати компоненти кольору. Наприклад, `rgb(255, 0, 0)` або `rgb(100%, 20%, 20%)`.

Приклад. Представлення кольору

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Кольори</title>
    <style>
      BODY {
        background-color: #3366CC; /* Колір фону веб-сторінки */
      }
      H1 {
        background-color: RGB(249, 201, 16); /* Колір фону під заголовком */
      }
      P {
        background-color: maroon; /* Колір фону пуд текстом абзацу */
        color: white; /* Колір тексту */
      }
    </style>
  </head>
  <body>
    <h1>Lorem ipsum dolor sit amet</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
    nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat
    volutpat.</p>
  </body>
</html>
```



Адреси

Адрес (URI, Uniform Resource Identifiers, уніфікований ідентифікатор ресурсів) використовується для того, щоб знайти файл, наприклад, картинку, яку можна використовувати як фон на веб-сторінці. Щоб вказати адресу файлу, використовується ключове слово `url()`, а в середині дужок – відносна або абсолютна адреса. Адресу можна писати в одинарних або подвійних лапках.

Приклад. Адреса графічного файлу

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title> Додавання фону </title>
    <style>
      body {
        background: url('http://webimg.ru/images/156_1.png') no-repeat;
      }
      div {
        background: url(images/warning.png) no-repeat;
        padding-left: 20px;
        margin-left: 200px;
      }
    </style>
  </head>
  <body>
    <div> Увага, запитувана сторінка не знайдена! </div>
  </body>
</html>
```

У цьому прикладі для селектора `body` використовується абсолютна адреса до графічного файлу, а для селектора `div` – відносна.

Ключові слова

В якості значень активно застосовуються ключові слова, які визначають бажаний результат дії стильових властивостей. Ключові слова пишуться без лапок.

Правильно: P { text-align: right; }

Невірно: P { text-align: "right"; }

Селектори тегів

В HTML для тегів можна задавати правила форматування, які визначають їх вигляд, такі як колір, фон, розмір і т. д. Ці правила задаються за допомогою стилів.

Стиль містить ім'я тега, якому він буде застосований, і список властивостей з їх значеннями. Властивості розділяються між собою крапкою з комою.

Наприклад, стиль для тега <p> може виглядати так:

```
p {  
    text-align: justify; /* Вирівнювання по ширині */  
    color: green; /* Зелений колір тексту */  
}
```

У цьому стилі змінюється колір тексту на зелений і вирівнювання по ширині.

Щоб застосувати стиль до тегу, його треба включити всередині контейнера <style> в тілі HTML-документу.

Стили застосовуються тільки до тегів, які безпосередньо відображаються в контейнері <body>, тому, якщо ви застосуєте стиль до тегу, який знаходиться за межами цього контейнера, він не буде видно.

Універсальний селектор

Іноді для всіх елементів веб-сторінки потрібно встановити один і той же стиль, наприклад, шрифт або розмір тексту. У такому випадку можна використовувати універсальний селектор, який застосовується до будь-якого елементу на сторінці.

Універсальний селектор позначається символом зірочки (*), а синтаксис запису правил стилю виглядає наступним чином:

***{опис правил стилю}**

В деяких випадках можна не вказувати універсальний селектор, наприклад, якщо ми хочемо задати один і той же стиль для всіх елементів з певним класом. У такому випадку записи *.class і .class є ідентичними за результатом.

У прикладі показано, як використовувати універсальний селектор для задання шрифту та розміру тексту для всіх елементів на сторінці. Можна помітити, що весь текст на сторінці має однаковий шрифт та розмір.
Приклад:

Отримати аналогічний результат можна, якщо використати селектор BODY замість універсального селектора.

Приклад. Використання універсального селектора

```
<!DOCTYPE HTML>  
<html>  
  <head>
```

```

<meta charset="utf-8">
<title>Універсальний селектор</title>
<style>
* {
  font-family: Arial, Verdana, sans-serif; /* Рубаний шрифт для тексту */
  font-size: 96%; /* Розмір тексту */
}
</style>
</head>
<body>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
</body>
</html>

```

Аналогічний результат можна отримати, якщо в даному прикладі поміняти селектор * на **BODY**.

Класи

Класи застосовують, коли необхідно визначити стиль для індивідуального елемента веб-сторінки або задати різні стилі для одного тега. При використанні спільно з тегамі синтаксис для класів буде наступний:

Тег.Ім'я класу { властивість1: значення; властивість2: значення; ... }

Всередині стилю спочатку пишеться бажаний тег, а потім, через точку, користувальницьке ім'я класу. Імена класів повинні починатися з латинського символу і можуть містити в собі дефіс (-) і підкреслення (_). Використання українських букв в іменах класів неприпустимо. Щоб вказати в коді HTML, що тег використовується з певним класом, до них додається атрибут **class="Ім'я класу"**.

Приклад. Використання класів

```

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

```

<style>

```
P { /* Звичайний абзац */  
  text-align: justify; /* Вирівнювання тексту по ширині */  
}  
P.cite { /* Абзац з класом cite */  
  color: navy; /* Колір тексту */  
  margin-left: 20px; /* Відступ зліва */  
  border-left: 1px solid navy; /* Границя зліва від тексту */  
  padding-left: 15px; /* Відстань від лінії до тексту */  
}
```

</style>

</head>

<body>

<p>Для штучного освітлення приміщення застосовуються люмінесцентні лампи. Вони відрізняються високою світловою віддачею, тривалим терміном служби, малою яскравістю поверхні, що світиться, близьким до природного спектральним складом випромінюваного світла, що забезпечує хорошу передачу кольору.**</p>**

<p class="cite"> Для виключення засвічення екрану дисплея світловими потоками віконні прорізи забезпечені світлорозсіюючими шторами.**</p>**

</body>

</html>

Для штучного освітлення приміщення застосовуються люмінесцентні лампи. Вони відрізняються високою світловою віддачею, тривалим терміном служби, малою яскравістю поверхні, що світиться, близьким до природного спектральним складом випромінюваного світла, що забезпечує хорошу передачу кольору.

Для виключення засвічення екрану дисплея світловими потоками віконні прорізи забезпечені світлорозсіюючими шторами.

Вид тексту, оформленого за допомогою стильових класів

Перший абзац вирівняний по ширині з текстом чорного кольору (цей колір задається браузером за замовчуванням), а наступний, до якого застосовано клас з ім'ям **cite** відображається синім кольором і лінією ліворуч.

Можна також використовувати класи і без вказівки тега. Синтаксис в цьому випадку буде наступний.

.Ім'я класу { властивість1: значення; властивість2: значення; ... }

При такому записі клас можна застосовувати до будь-якого тегу.

Приклад


```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

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

```
<title>Класи</title>
```

```
<style>
```

```
.gost {
```

```
color: green; /* Колір тексту */
```

```
font-weight: bold; /* Жирне накреслення */
```

```
}
```

```
.term {
```

```
border-bottom: 1px dashed red; /* Підкреслення під текстом */
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

<p>Согласно ****ГОСТ 12.1.003-83 ССБТ "Шум. Загальні вимоги безпеки ****", шумовою характеристикою робітничих місць при постійному шумі є рівні звукових тисків в децибелах в октавних смугах. Сукупність таких рівнів називається **<b class="term">** граничним спектром ****, номер якого чисельно дорівнює рівню звукового тиску в октавній смузі з середньгеометричною частотою 1000 Гц.

```
</p>
```

```
</body>
```

```
</html>
```

Согласно **ГОСТ 12.1.003-83 ССБТ " Шум. Загальні вимоги безпеки "**, шумовою характеристикою робітничих місць при постійному шумі є рівні звукових тисків в децибелах в октавних смугах. Сукупність таких рівнів називається граничним спектром, номер якого чисельно дорівнює рівню звукового тиску в октавній смузі з середньгеометричною частотою 1000 Гц.

Вид тегів, оформлених за допомогою класів

Класи зручно використовувати, коли потрібно застосувати стиль до різних елементів веб-сторінки: комірок таблиці, посилань, абзаців та ін. У прикладі показано зміну кольору фону рядків таблиці для створення "зебри".

Приклад. Використання класів

```
<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <title> Каміння </title>
  <style>
    table.jewel {
      width: 100%; /* Ширина таблиці */
      border: 1px solid #666; /* Рамка навколо таблиці */
    }
    th {
      background: #009383; /* Колір фону */
      color: #fff; /* Колір тексту */
      text-align: left; /* Вирівнювання по лівому краю */
    }
    tr.odd {
      background: #ebd3d7; /* Колір фону */
    }
  </style>
</head>
<body>
  <table class="jewel">
    <tr>
      <th>Назва</th><th>Колір</th><th>Твердість по Моосу</th>
    </tr>
    <tr class="odd">
      <td>Алмаз</td><td>Білий</td><td>10</td>
    </tr>
    <tr>
      <td>Рубін</td><td>Червоний</td><td>9</td>
    </tr>
    <tr class="odd">
      <td>Аметист</td><td>Блакитний</td><td>7</td>
    </tr>
    <tr>
      <td>Смарагд</td><td>Зелений</td><td>8</td>
    </tr>
```



```

.level4 { font-size: 1.6em; }
.level5 { font-size: 1.8em; }
.level6 { font-size: 2em; }
A.tag {
  color: #468be1; /* Колір посилань */
}
</style>
</head>
<body>
<div>
  <a href="/term/2" class="tag level6">Paint.NET</a>
  <a href="/term/69" class="tag level6">Photoshop</a>
  <a href="/term/3" class="tag level5">колір</a>
  <a href="/term/95" class="tag level5">фон</a>
  <a href="/term/11" class="tag level4">палітра</a>
  <a href="/term/43" class="tag level3">шари</a>
  <a href="/term/97" class="tag level2">світло</a>
  <a href="/term/44" class="tag level2">панелі</a>
  <a href="/term/16" class="tag level1">лінія</a>
  <a href="/term/33" class="tag level1">прямокутник</a>
  <a href="/term/14" class="tag level1">піксел</a>
  <a href="/term/27" class="tag level1">градієнт</a>
</div>
</body>
</html>

```

[Paint.NET](#) [Photoshop](#) [колір](#)
[фон](#) [палітра](#) [шари](#) [світло](#) [панелі](#) [лінія](#)
[прямокутник](#) [піксел](#) [градієнт](#)

Хмара тегів

В стилях також допускається використовувати запис виду **.layer1.layer2**, де **layer1** і **layer2** являють собою імена класів. Стиль застосовується тільки для елементів, у яких одночасно задані класи **layer1** і **layer2**.

Псевдокласи

Псевдокласи визначають динамічний стан елементів, який змінюється за допомогою дій користувача, а також положення в дереві документа.

Прикладом такого стану служить текстове посилання, яке змінює свій колір при наведенні на нього курсора миші. При використанні псевдокласів браузер не перевантажує поточний документ, тому з допомогою псевдокласів можна отримати різні динамічні ефекти на сторінці.

Синтаксис застосування псевдокласів наступний.

Селектор: Псевдокласс { опис правил стилю }

Спочатку вказується селектор, до якого додається псевдоклас, потім слідує двокрапка, після якого йде назва псевдокласу. Допускається застосовувати псевдокласи до імен ідентифікаторів або класів (**A.menu: hover {color: green}**), а також до контекстним селекторам (**.menu A: hover {background: #fc0}**). Якщо псевдоклас вказується без селектора попереду (**:hover**), то він буде застосовуватися до всіх елементів документа.

Умовно всі псевдокласи діляться на три групи:

- що визначають стан елементів;
- що мають відношення до дерева елементів;
- що вказують мову тексту.

Псевдокласи, що визначають стан елементів

До цієї групи належать псевдокласи, які розпізнають поточний стан елемента і застосовують стиль тільки для цього стану.

:active

Відбувається при активації користувачем елемента. Наприклад, посилання стає активним, якщо навести на нього курсор і клацнути мишкою. Незважаючи на те, що активним може стати практично будь-який елемент веб-сторінки, псевдоклас **:active** використовується переважно для посилань.

:link

Застосовується до невідвіданих посилань, тобто таких посилань, на які користувач ще не натискав. Браузер деякий час зберігає історію

відвідувань, тому посилання може бути позначене як відвідане хоча б тому, що по ній був зафіксований перехід раніше.

Запис **A {...}** і **a: link {...}** за своїм результату рівноцінна, оскільки в браузері дає один і той же ефект, тому псевдоклас **:link** можна не вказувати. Винятком є якоря, на них дія **:link** не поширюється.

:focus

Застосовується до елемента при отриманні ним фокусу. Наприклад, для текстового поля форми отримання фокусу означає, що курсор встановлений в поле, і за допомогою клавіатури можна вводити в нього текст.

Приклад. Застосування псевдокласу

:focus

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

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

```
<title>Псевдокласи</title>
```

```
<style>
```

```
INPUT:focus {  
  color: red; /* Червоний колір тексту */  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form action="">
```

```
<p><input type="text" value="Чорний текст"></p>
```

```
<p><input type="text" value="Чорний текст"></p>
```

```
</form>
```

```
</body>
```

```
</html>
```

У другому рядку знаходиться курсор, тому текстове поле отримало фокус.

Зміна стилю тексту при отриманні фокусу

В даному прикладі в текстовому полі міститься попередній текст, він визначається значенням атрибута **value** тега **<input>**. При клацанні по елементу форми відбувається отримання полем фокусу, і колір тексту змінюється на червоний. Досить клацнути в будь-якому місці сторінки (крім текстового поля, природно), як відбудеться втрата фокусу і текст повернеться до первісного чорного кольору.

Результат буде видно тільки для тих елементів, які можуть отримати фокус. Зокрема, це теги **<a>**, **<input>**, **<select>** і **<textarea>**.

:hover

Псевдоклас **:hover** активізується, коли курсор знаходиться в межах елемента, але клацання по ньому не відбувається.

:visited

Даний псевдоклас застосовується до відвіданих посилань. Зазвичай таке посилання змінює свій колір за замовчуванням на фіолетовий, але за допомогою стилів колір і інші параметри можна задати самостійно.

Приклад. Зміна кольору посилань

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

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

```
<title>Псевдокласи</title>
```

```
<style>
```

```
A:link {
```

```
color: #036; /* Колір невідвіданих посилань */
```

```
}
```

```
A:visited {
```

```
color: #606; /* Колір відвіданих посилань */
```

```
}
```

```
A:hover {
```

```

    color: #f00; /* Колір посилань при наведенні на них курсору миші */
}
A:active {
    color: #ff0; /* Колір активних посилань */
}
</style>
</head>
<body>
<p>
<a href="1.html">Посилання 1</a> |
<a href="2.html">Посилання 2</a> |
<a href="3.html">Посилання 3</a></p>
</body>
</html>

```

[Посилання 1](#) | [Посилання 2](#) | [Посилання 3](#)

Вид посилання при наведенні на нього курсору миші

В даному прикладі показано використання псевдокласів спільно з посиланнями. При цьому має значення порядок проходження псевдокласів. Спочатку вказується **:visited**, а потім йде **:hover**, в іншому випадку відвідані посилання не будуть змінювати свій колір при наведенні на них курсору.

Селектори можуть містити більше одного псевдокласу, які перераховуються поспіль через двокрапку, але тільки в тому випадку, якщо їх дії не суперечать один одному. Так, запис **a:visited:hover** є коректною, а запис **A:link:visited** – ні. Втім, якщо підходити формально, то валідатор CSS вважає правильним будь-яке поєднання псевдокласів.

Псевдоклас **:hover** не обов'язково повинен застосовуватися до посилань, його можна додавати і до інших елементів документа. Так, в прикладі показана таблиця, рядки якої змінюють свій колір при наведенні на них курсору миші. Це досягається за рахунок додавання псевдокласу **:hover** до селектору **TR**.

Приклад. Виділення рядків таблиці

```

<!DOCTYPE HTML>
<html>

```



```
<head>
<meta charset="utf-8">
<title>Псевдокласи</title>
<style>
table { border-spacing: 0; }
td { padding: 4px; }
tr:hover {
  background: #fc0; /* Міняємо колір фону рядка таблиці */
}
</style>
</head>
<body>
<table width="400" border="1">
  <tr>
    <th>Назва</th>
    <th>Колір</th>
    <th>Твердість по Моосу</th>
  </tr>
  <tr>
    <td>Алмаз</td>
    <td>Білий</td>
    <td>10</td>
  </tr>
  <tr>
    <td>Рубін</td>
    <td>Червоний</td>
    <td>9</td>
  </tr>
  <tr>
    <td>Аметист</td>
    <td>Блакитний</td>
    <td>9</td>
  </tr>
  <tr>
    <td>Смарагд</td>
    <td>Зелений</td>
    <td>8</td>
  </tr>
</table>
</body>
</html>
```

Назва	Колір	Твердість по Моосу
Алмаз	Білий	10
Рубін	Червоний	9
Аметист	Блакитний	9
Смарагд	Зелений	8

Виділення рядків таблиці при наведенні на них курсору миші

Псевдокласи, що мають відношення до дерева документа

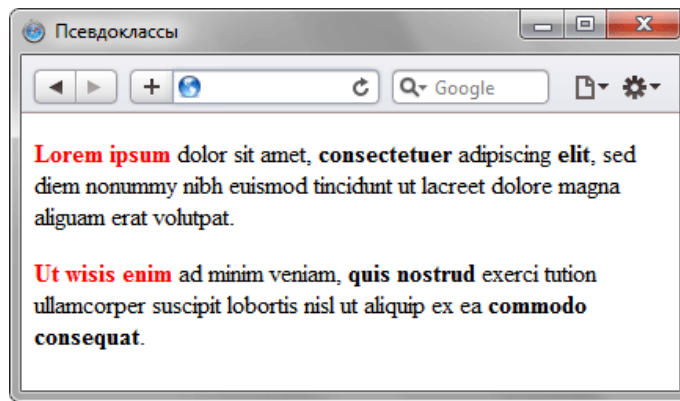
До цієї групи відносяться псевдокласи, які визначають положення елемента в дереві документа і застосовують до нього стиль в залежності від його статусу.

:first-child

Застосовується до першого дочірнього елемента селектора, який розташований в дереві елементів документа.

Приклад. Використання псевдокласу :first-child <!DOCTYPE HTML>

```
<html>
<head>
  <meta charset="utf-8">
  <title>Псевдокласи</title>
  <style type="text/css">
    B:first-child {
      color: red; /* Червоний колір тексту */
    }
  </style>
</head>
<body>
  <p><b>Lorem ipsum</b> dolor sit amet, <b>consectetur</b>
  adipiscing <b>elit</b>, sed diam nonummy nibh euismod tincidunt
  ut laoreet dolore magna aliquam erat volutpat.</p>
  <p><b>Ut wisis enim</b> ad minim veniam, <b>quis nostrud</b>
  exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea <b>commodo
  consequat</b>.</p>
</body>
</html>
```



Використання псевдокласу :first-child

В даному прикладі псевдоклас **:first-child** додається до селектору **B** і встановлює для нього червоний колір тексту. Хоча контейнер зустрічається в першому абзаці три рази, червоним кольором буде виділено лише перше згадування, тобто текст "Lorem ipsum". В інших випадках вміст контейнера **** відображається чорним кольором. З наступним абзацом все починається знову, оскільки батьківський елемент змінився. Тому фраза «Ut wisis enim» також буде виділена червоним кольором.

Псевдоклас **:first-child** найзручніше використовувати в тих випадках, коли потрібно задати різний стиль для першого і інших однотипних елементів. Наприклад, у деяких випадках червоний рядок для першого абзацу тексту не встановлюють, а для інших абзаців додають відступ першого рядка. З цією метою застосовують властивість **text-indent** з потрібним значенням відступу. Але щоб змінити стиль першого абзацу і прибрати для нього відступ потрібно скористатися псевдокласом **:first-child**.

Приклад. Відступи для абзацу

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

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

```
<title>Псевдокласи</title>
```

```
<style>
```

```
P {
```

```
text-indent: 1em; /* Відступ першого рядка */
```

```
}
```

```
P:first-child {
```

```

text-indent: 0; /* Для першого абзацу відступ прибираємо */
}
</style>
</head>
<body>
  <p> Історію цю вже почали забувати, хоча знаходилися городяни, які час від часу розповідали її новоприбулим в місто відвідувачам.</p>
  <p> Легенда обростала подробицями і вже зовсім не нагадувала подію насправді подія. І, тим не менш, жодна людина не наважувався заїкнутися про неї з настанням темряви.</p>
  <p> Але одного разу в місто знову увійшов незнайомец. Він кульгав на ліву ногу.</p>
</body>
</html>

```

Історію цю вже почали забувати, хоча знаходилися городяни, які час від часу розповідали її новоприбулим в місто відвідувачам.

Легенда обростала подробицями і вже зовсім не нагадувала подію насправді подія. І, тим не менш, жодна людина не наважувався заїкнутися про неї з настанням темряви.

Але одного разу в місто знову увійшов незнайомец. Він кульгав на ліву ногу.

Зміна стилю першого абзацу

У цьому прикладі перший абзац тексту не містить відступу першого рядка, а для інших його встановлено.

Псевдокласи, що задають мову тексту

Для документів, що одночасно містять тексти на декількох мовах має значення дотримання правил синтаксису, характерні для того чи іншого мови. З допомогою псевдокласів можна змінювати стиль оформлення іноземних текстів, а також деякі налаштування.

:lang

Визначає мову, яка використовується в документі або його фрагменту. В коді HTML мова встановлюється через атрибут **lang**, він зазвичай додається до тегу **<html>**. З допомогою псевдокласу **:lang** можна задавати певні установки, характерні для різних мов, наприклад, вид лапок в цитатах. Синтаксис наступний.

Елемент: **lang**(мова) { ... }

В якості мови можуть виступати наступні значення: **ua** – українська; **en** – англійська; **de** – німецька; **fr** – французька; **it** – італійська.

Приклад. Вид лапок в залежності від мови

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

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

```
<title>lang</title>
```

```
<style>
```

```
p {
```

```
font-size: 150%; /* Розмір тексту */
```

```
}
```

```
q:lang(de) {
```

```
quotes: "\201E" "\201C"; /* Вид лапок для німецької мови */
```

```
}
```

```
q:lang(en) {
```

```
quotes: "\201C" "\201D"; /* Вид лапок для англійської мови */
```

```
}
```

```
q:lang(fr), q:lang(ru) { /* Вид лапок для української та французької мови */
```

```
quotes: "\00AB" "\00BB";
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p> Цитата французькою мовою: <q lang="fr">Ce que femme veut,
```

```
Dieu le veut</q>.</p>
```

```
<p> Цитата німецькою: <q lang="de">Der Mensch, versuche die Gotter  
nicht</q>.</p>
```

```
<p> Цитата англійською: <q lang="en">To be or not to be</q>.</p>
```

```
</body>
```

```
</html>
```

Для відображення типових лапок у прикладі використовується стильова властивість **quotes**, а саме перемикання мови і відповідного виду лапок відбувається через атрибут **lang**, що додається до тегу **<q>**.

Цитата французькою мовою: «Ce que femme veut, Dieu le veut».

Цитата німецькою: „Der Mensch, versuche die Gotter nicht“.

Цитата англійською: “To be or not to be”.

Різні лапки для різних мов