**Теоретичні відомості до лабораторної роботи № 2**

**Тема:***Стилі відображення гіпертексту CSS.*

**Мета:***здійснити ознайомлення студентів з поняттям «каскадні таблиці стилів» та навчити працювати з стилями відображення гіпертексту CSS.*

**Теоретичні відомості**

[**https://css.in.ua/html/tags**](https://css.in.ua/html/tags)

[**https://html5book.ru/css-css3/**](https://html5book.ru/css-css3/)

Мова HTML є мовою розмітки гіпертексту з дуже обмеженими можливостями. Вона була створена для того, щоб визначати структуру документа для розміщення його в Веб-середовищі, а не для роботи зі стилями.

Поступово в HTML були додані теги та технології, які дозволяли створювати та контролювати структуру та вигляд документа, з’явилися таблиці, фрейми, інструменти що вирівнюють текст, з’являється мова JavaScript.

У зв’язку з тим, що HTML не мав можливості управління зовнішнім виглядом Веб-сторінок, щоб вирішити ці проблеми, консорціумом W3C була розроблена технологія Cascading Style Sheets або CSS (Каскадні таблиці стилів) - технологія опису зовнішнього вигляду документа, написаного мовою розмітки. В CSS надається можливість призначити усім об'єктам стиль, опис якого може зберігатися в самому HTML-файлі, або в окремому файлі.

Один і той самий [HTML](http://uk.wikipedia.org/wiki/HTML) або [XML](http://uk.wikipedia.org/wiki/XML) документ може бути відображенний по-різному в залежності від використаного **CSS**. Стилі для відображення сторінки можуть бути:

* Стилі автора (інформація надана автором сторінки):
  + зовнішні таблиці стилів ([англ.](http://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D1%96%D0%B9%D1%81%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *stylesheet*), частіше за все окремий файл або файли **.css**
  + внутрішні таблиці стилів, включені як частина документу або блоку
  + стилі для окремого елементу
* Стилі користувача
  + локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад [Opera](http://uk.wikipedia.org/wiki/Opera))
* Стилі переглядача (броузера)
  + стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, те, в якій послідовності і для яких елементів застосовуються стилі. Таким чином використовується принцип *каскадності*, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена більш загальними стилями.

**Переваги**

* Інформація про стиль для цілого [сайту](http://uk.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82) 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: наприклад великий розмір шрифту для користувачів з послабленим зором, стилі для виводу сторінки на [принтер](http://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BD%D1%82%D0%B5%D1%80), стиль для мобільних пристроїв;
* Сторінки зменшуються в об'ємі та стають більш структурованими, за рахунок того що інформація про стилі відділена від тексту та має певні правила застосування і сторінка побудована з їх урахуванням;
* Прискорення завантаження сторінок і зменшення обсягів інформації, що передається, навантаження на сервер та канал передачі. Досягається за рахунок того, що сучасні браузери здатні кешувати (запам'ятовувати) інформацію про стилі і використовувати для всіх сторінок, а не завантажувати для кожної.

**ПРАВИЛА В CSS.** CSS надає можливість створювати правила, які легко змінювати, редагувати і застосовувати до усіх визначених нами елементів. Кожне правило, складається з двох частин. У лівій частині міститься *селектор (selector),* а у правій - *блок визначення (declaration block).* Блок визначення складається з набору *властивостей* *(property)* та їх *значень (value)*.

*Селектор –* елемент, який визначає правило.

*Властивість* описує елемент, що вводиться.

*Значення* визначають природу *властивостей.*

| *Селектор* | *блок визначення* |
| --- | --- |
| селектор | {властивість: значення;} |

**ДОДАВАННЯ ТАБЛИЦЬ СТИЛІВ В HTML-ДОКУМЕНТ.** Існують **три способи** додавання правил CSS в HTML-документи:

* ***СПОСІБ ПЕРШИЙ: ДОДАВАННЯ CSS В HTML-ТЕГ.***

У цьому способі CSS додається в HTML-документ за допомогою HTML-атрибуту style у середині будь-якого HTML-тегу, що знаходиться у контейнері <body>.

<html>

<head>

<title>Приклад №1</title>

<meta charset="UTF-8">

</head>

**<body style="background-color: #FF0000;">**

<p>Колір фону цієї сторінки - червоний</p>

</body>

</html>

Цей спосіб використовується у тому разі коли окремому елементу потрібно надати декілька стилів не використовуючи вбудовані або зовнішні стилі.

Застосування цього способу несе за собою певні недоліки:

- збільшується об’єм файлу, що приводить до збільшення часу завантаження веб-сторінки;

- ускладнює редагування документів.

* ***СПОСІБ ДРУГИЙ: ВСТАНОВЛЕННЯ СТИЛЮ ДЛЯ ТЕГІВ В HTML-ДОКУМЕНТІ***

CSS додається в HTML-документ за допомогою HTML-тегу <style> в середині контейнеру <head>. В ньому описуються всі стилі, що будуть використані.

<html>

<head>

<title>Приклад №2</title>

<meta charset="UTF-8">

**<style type="text/css">**

**body {background-color: #FF0000;}**

**</style>**

</head>

<body>

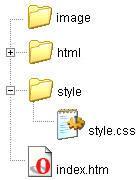
<p>Колір фону цієї сторінки - червоний</p>

</body>

</html>

* ***СПОСІБ ТРЕТІЙ: ПОСИЛАННЯ НА ТАБЛИЦЮ СТИЛІВ***

Зовнішня таблиця стилів являє собою звичайний текстовий файл з розширенням css.



Для того щоб зробити посилання на зовнішній файл із HTML-документа (index.htm) на файл таблиці стилів (style.css) треба у контейнері<head>вставити наступну стрічку:

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

Це посилання указує браузеру, що він повинен використовувати правила відображення HTML-файлу з CSS-файлу.

<html>

<head>

<title>Приклад №3</title>

<meta charset="UTF-8">

**<link rel="stylesheet" type="text/css" href="../style/style.css">**

</head>

<body>

<p>Колір фону цієї сторінки - голубий</p>

</body>

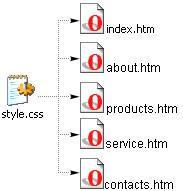
</html>

У файл **style.css** записуємо наступне:

body

{background-color: # 79e2f5;}

Найважливішим тут є те, що один CSS-файл можна використовувати для управління відображення багатьох HTML-документів.



**ІДЕНТИФІКАЦІЯ І ГРУПУВАННЯ ЕЛЕМЕНТІВ**

Атрибут class вказує, що елемент є членом певного класу. Модифікуємо список у файлі **index.html**, таким чином щоб одержати три списки посилань. Потрібно, щоб кольори посилань цих списків відрізнялися.

У файлі **index.html:**

<p>Список №1:</p>

<ul>

<li><a href="html/example1.html">Приклад 1</a></li>

<li><a href="html/example2.html">Приклад 2</a></li>

<li ><a href="html/example3.html">Приклад 3</a></li>

</ul>

<p>Список №2:</p>

<ul>

<li><a href="html/example4.html">Приклад 4</a></li>

<li><a href="html/example5.html">Приклад 5</a></li>

<li><a href="html/example6.html">Приклад 6</a></li>

</ul>

<p>Список №3:</p>

<ul>

<li><a href="html/example7.html">Приклад 7</a></li>

<li><a href="html/example8.html">Приклад 8</a></li>

<li><a href="html/example9.html">Приклад 9</a></li>

</ul>

<a href="html/example10.html">Приклад 10</a>

Щоб досягти нашої мети розділимо посилання на три категорії за допомогою привласнення першим двом спискам класу кожному посиланню атрибутом class, а третій список залишаємо без класу.

У файлі **index.html** додамо class="classname" в тег до якого ми будемо застосовувати клас.

<p>Список №1:</p>

<ul>

<li><a href="html/example1.html" class="list1">Приклад 1</a></li>

<li><a href="html/example2.html" class="list1">Приклад 2</a></li>

<li ><a href="html/example3.html" class="list1">Приклад 3</a></li>

</ul>

<p>Список №2:</p>

<ul>

<li><a href="html/example4.html" class="list2">Приклад 4</a></li>

<li><a href="html/example5.html" class="list2">Приклад 5</a></li>

<li><a href="html/example6.html" class="list2">Приклад 6</a></li>

</ul>

<p>Список №3:</p>

<ul>

<li><a href="html/example7.html">Приклад 7</a></li>

<li><a href="html/example8.html">Приклад 8</a></li>

<li><a href="html/example9.html">Приклад 9</a></li>

</ul>

<a href="html/example3.html" id="exam10">Приклад 10</a>

У файлі style.css опишемо властивості класів для посилань. Синтаксис опису класу має наступний вигляд:

| селектор. | ім’я класу | {властивість: значення;} |
| --- | --- | --- |

a {

color: blue;

}

a.list1 {

color: #FFBB00;

}

a.list2 {

color: #800000;

}

Окрім групування елементів вам може знадобитися ідентифікувати один унікальний елемент. Це можна реалізувати за допомогою атрибуту id.

Особливість id в тому, що в документі не може бути більш за один елемент з даним конкретним id. Кожен id має бути унікальним. У інших випадках треба використовувати атрибут class. У файлі **index.html** додамо id="idname" для посилання на Приклад 10:

<a href="html/example3.html" id="exam10">Приклад 10</a>

Оформлення ідентифікатора у css-файлі буде мати наступний вигляд:

#exam10

{

color: #56c500;

}**КОЛІР ТА ФОН.** Властивість color описує колір елементу. Наприклад, ми бажаємо встановити зелений колір тексту абзаца, для цього у файлі style.css створимо таке правило:

p.ex4{

color: #00ff00;

}

<html>

<head>

<title>Приклад №4</title>

<meta charset="UTF-8">

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

</head>

<body>

<p class=”ex4”> Колір тексту у цьому абзаці зелений</p>

</body>

</html>

Властивість background-color описує колір фону елемента. Для того, щоб замінити фон всієї веб-сторінки властивість background-color слід застосувати до тегу <body>.

Також властивість background-color можна застосовувати до інших елементів веб-сторінки.

У файл style.css додаємо правило для Прикладу 5, що описують фон абзацу.

p.ex5{

color: #00ff00;

background-color:#ff0000;

}

<html>

<head>

<title>Приклад №5</title>

<meta charset="UTF-8">

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

</head>

<body class="body5">

<p class="ex5"> Колір тексту у цьому абзаці зелений на червоному фоні.</p>

</body>

</html>

За допомогою властивості background- image, зображення можна помістити на задній план елементу або зробити фоном веб-сторінки.

У цьому прикладі ми в якості зображення будемо використовувати зображення шару.



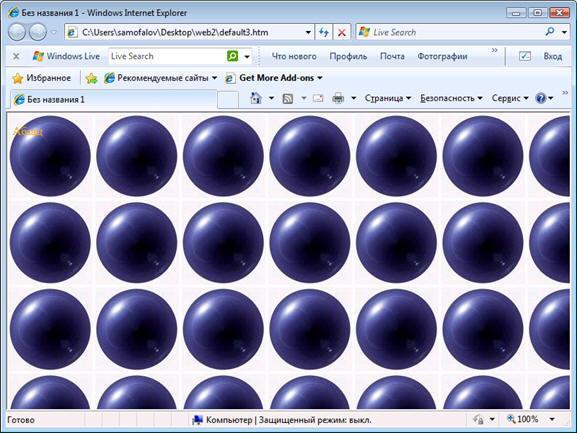
Щоб зробити дане зображення фоном веб-сторінки достатньо в файлі style.css для **Прикладу 5** застосувати параметри тега <body>

body.body5{

background-image: url("../images/ball.jpg");

}

**Властивість BACKGROUND-REPEAT.** Результатом виконання попереднього прикладу було повторення зображення шару, як по вертикалі так і по горизонталі, щоб керувати цим повторенням застосовується властивість background-repeat.



| **Таблиця значень background-repeat** | |
| --- | --- |
| **Значення** | **Властивість** |
| background-repeat: repeat-x | Зображення повторюється тільки по горизонталі. |
| background-repeat: repeat-y | Зображення повторюється тільки по вертикалі |
| background-repeat: repeat | Зображення повторюється по горизонталі і вертикалі. |
| background-repeat: no-repeat | Зображення не повторюється. |
|  |  |

**Модифікуємо Приклад 5:**

body. body5{

background-image: url("../images/ball.jpg");

background-repeat: repeat-x;

}

**Властивість BACKGROUND-ATTACHMENT** надає можливість фіксування фонового зображення або прокручувати його разом з текстом.

| **Таблиця значень background-attachment** | |
| --- | --- |
| **Значення** | **Властивість** |
| background-attachment: scroll | Зображення не фіксується |
| background-attachment: fixed | Зображення зафіксовано. |
|  |  |

**Модифікуємо Приклад 5:**

body. body5{

background-image: url("../images/ball.jpg");

background-repeat: repeat-x;

background-attachment: fixed;

}

За замовчуванням фонове зображення розміщається у верхньому лівому куті екрану. За допомогою властивості background-position зображення можна розміщати у будь-якому місці екрану.

Значення властивості background-position являють собою вісь координат.

**Модифікуємо Приклад 5:**

body. body5{

background-image: url("../images/ball.jpg");

background-repeat: no-repeat;

background-position: 5cm 5cm

}

Після виконання цього прикладу ми побачимо, що зображення розташовано на 5 см зліва та на 5 см зверху.

В якості значень властивості background-position можна використовувати значення у процентах, ключові слова (right, left, top, bottom).

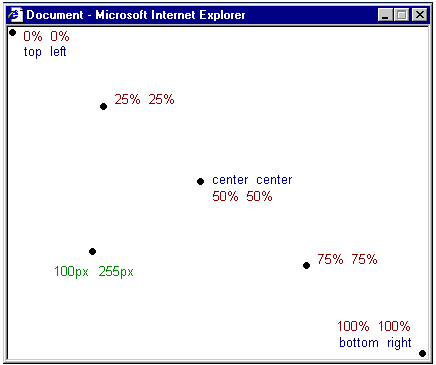
body.body5{

background-image: url("../images/ball.jpg");

background-repeat: no-repeat;

background-position: 100% 25%

}



Властивості background, що були розглянуті, можна записувати у скороченому вигляді.

Розглянемо фрагмент код:

background-color:#ff0000;

background-image: url("../images/ball.jpg");

background-repeat: no-repeat;

background-attachment: fixed;

background-position: 5cm 5cm

Для того, щоб властивість background можна було записати однією стрічкою, необхідно розташувати його властивості у такому порядку:

[background-color] | [background-image] | [background-repeat] |

[background-attachment] | [background-position].

background: #ff0000 url("../images/ball.jpg") no-repeat fixed 5cm 5cm;

Якщо властивість відсутня, то вона автоматично отримує значення за замовчуванням. Наприклад, якщо властивості background-attachment та background-position відсутні, то вони отримують свої значення за замовчуванням.

**Управління шрифтом.**

В старих версіях HTML шрифт оформлявся за допомогою тегу<font>, але цей тег треба було додавати кожного разу, коли було необхідно встановити шрифт, від цього збільшувався розмір веб-сторінки, незручно було змінювати властивість шрифту.

Щоб задати сімейство шрифтів використовується властивість font-family. В цій властивості завжди вказується ряд шрифтів, розділених комою, наприкінці списку вказується сімейство шрифтів. При застосуванні шрифтів до веб-сторінки завжди задається основний шрифт, а потім альтернативний. У разі відсутності на комп’ютері користувача заданих шрифтів, то веб-сторінка, як мінімум буде відображена шрифтом, що входить до цього сімейства.

Існують п’ять сімейств шрифтів:

| **Шрифт** | **Сімейство шрифтів** |
| --- | --- |
| Times New Roman  Georgia | serif(з засічками) |
| Arial  Candara | sans-serif (без засічок) |
| Courier  Courier New | monospace (моноширинний) |
| Bradley Hand ITC  Edwardian Script ITC | cursive (рукописний) |
| ∧∑®™⎡⎨©⌠ | fantasy (декоративний) |

Шрифти сімейства serif найкраще підходять для основного тексту сторінки. Засічки допомагають направляти увагу читача уздовж рядка

Шрифти сімейства sans-serif використовуються для оформлення заголовків, панелей посилань та посилань. Шрифти без засічок звертають на себе більше уваги, але погано підходять для довгого читання.

У моношириних шрифтах усі символи мають однакову ширину. Слід зазначити, що у моноширинних шрифтах збільшується між символьний інтервал, як зліва так і з права символу (цей інтервал є невід’ємною частиною символу). Моноширинні шрифти допускаються тільки для створення якихось особливих ефектів оформлення — наприклад, у даній роботі моношириним шрифтом набрано фрагменти коду HTMLта CSS. У моноширинному шрифті всякий символ має одну і ту ж ширину.

У файлі style.css створимо для оформлення абзацу таке правило:

**p.ex6{font-family: Arial, Candara, Century-Gothic, sans-serif}**

**<html>**

**<head>**

**<title>Приклад №6</title>**

**<meta charset="UTF-8">**

**<link rel="stylesheet" type="text/css" href="../style/style.css" />**

**</head>**

**<body>**

**<p class=”ex6”> Шрифт цього абзацу відноситься до сімейства sans-serif</p>**

**</body>**

**</html>**

У цьому прикладі основним шрифтом абзацу є шрифт Arial у разі його відсутності буде загружено наступний, у разі відсутності усіх шрифтів абзац буде оформлено шрифтом сімейства sans-serif.

***Властивість font-style*** визначає стиль шрифту з обраного сімейства може мати наступні значення:

* normal– звичайний шрифт;
* italic– курсивний шрифт (більш декоративний шрифт з нахилом в право);
* oblique– нахилений шрифт (звичайний шрифт нахилений в право).

У файлі style.css створимо таке правило:

**h1.ex7{**

**font-family: Arial, Candara, Century-Gothic, sans-serif;**

**font-style: normal;**

**}**

**h2.ex7{**

**font-family: Times New Roman, Georgia, serif;**

**font-style: italic;**

**}**

**p.ex7{**

**font-family: Times New Roman, Georgia, serif;**

**font-style: oblique;**

**}**

**<html>**

**<head>**

**<title>Приклад №7</title>**

**<meta charset="UTF-8">**

**<link rel="stylesheet" type="text/css" href="../style/style.css" />**

**</head>**

**<body>**

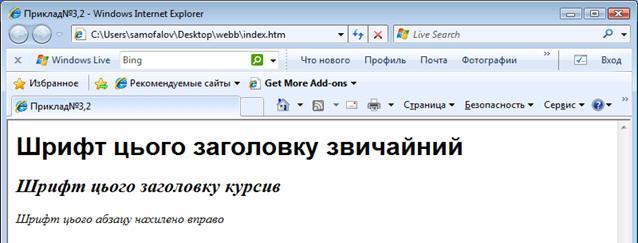
**<h1 class="ex7">Шрифт цього заголовку звичайний</h1>**

**<h2 class="ex7">Шрифт цього заголовку курсив</h2>**

**<p class="ex7"> Шрифт цього абзацу нахилено вправо</p>**

**</body>**

**</html>**



***Властивість font-variant*** використовується для вибору між варіантами normal і small-caps.

Параметр normal задає звичайні літери.

Параметр small-caps задає малі заголовні букви (upper case) замість букв нижнього регістру.

У файлі style.css створимо таке правило:

**p.ex8{**

**font-family: Arial, Candara, Century-Gothic, sans-serif;**

**font-style: normal;**

**font-variant: small-caps;**

**}**

**<html>**

**<head>**

**<title>Приклад №8</title>**

**<meta charset="UTF-8">**

**<link rel="stylesheet" type="text/css" href="../style/style.css" />**

**</head>**

**<body>**

**<p class="ex8">Абзац оформлено малими прописними літерами</p>**

**</body>**

**</html>**

***Властивість font-weight*** описує, наскільки товстим, або "важким", повинен відображуватися шрифт. Шрифт може бути normal або bold. Деякі браузери підтримують навіть числові значення 100-900 (у сотнях) для опису ваги шрифту.

Іноді при значенні "500" або "600" браузери відображають "проміжну" насиченість між normal і bold.

У файлі style.css створимо таке правило:

**p.ex9{**

**font-family: Arial, Candara, Century-Gothic, sans-serif;**

**font-style: normal;**

**font-variant: normal;**

**font-weight: 600**

**}**

**<html>**

**<head>**

**<title>Приклад №9</title>**

**<meta charset="UTF-8">**

**<link rel="stylesheet" type="text/css" href="../style/style.css" />**

**</head>**

**<body>**

**<p class="ex9">Товщина шрифту абзацу дорівнює 600</p>**

**</body>**

**</html>**

Властивість font-size задає розмір шрифту для відображення вмісту елементу. Щоб задати розмір шрифту використовуються різні одиниці виміру (наприклад, пікселі або відсотки).

**h1.ex10{**

**font-family: Arial, Candara, Century-Gothic, sans-serif;**

**font-style: normal;**

**font-size: 30px;**

**}**

**h2.ex10{**

**font-family: Times New Roman, Georgia, serif;**

**font-style: italic;**

**font-size: 100pt;**

**}**

**h3.ex10{**

**font-family: Times New Roman, Georgia, serif;**

**font-style: italic;**

**font-size: 120%;**

**}**

**p.ex10{**

**font-style: italic;**

**font-weight: bold;**

**font-size: 30px;**

**font-family: arial, sans-serif;**

**}**

**<html>**

**<head>**

**<title>Приклад №10</title>**

**<meta charset="UTF-8">**

**<link rel="stylesheet" type="text/css" href="../style/style.css" />**

**</head>**

**<body>**

**<h1 class="ex10">Шрифт нормальний, розмір 30рх</h1>**

**<h2 class="ex10"> Шрифт курсив, розмір 100рt </h2>**

**<h3 class="ex10"> Шрифт курсив, розмір 120% </h3>**

**<p class="ex10"> Шрифт напівжирний, курсив, 30рх</p>**

**</body>**

**</html>**

Властивості font, що були розглянуті, можна записувати у скороченому вигляді. Розглянемо фрагмент коду оформлення абзацу:

p{

font-style: italic;

font-weight: bold;

font-size: 30px;

font-family: arial, sans-serif;

}

Використовуючи скорочений запис, код можна спростити:

p{

font: italic bold 30px arial, sans-serif;

}

Порядок розташування властивостей такий:

font-style | font-variant | font-weight | font-size | font-family.

**УПРАВЛІННЯ ТЕКСТОМ**

***Властивість text-indent*** задає відступ першого рядка при відображенні блокових елементів. Відступ може бути, як позитивним так і негативним і задається одним з наступних способів:

| **Значення** | **Дія** |
| --- | --- |
| <розмір> | визначає, що відступ першого рядка фіксований |
| <відсоток> | визначає, що відступ обчислюється щодо ширини блоку, що вміщується. |
|  |  |

Додаємо для **Прикладу 10** властивість *text-indent* (задані раніше властивості залишаємо без змін):

**p.ex10{**

**text-indent: 10%;**

**}**

Модифікуємо для **Приклад 10** властивість text-indent:

**p.ex10{**

**text-indent: 3pt;}**

Властивість text-align задає вирівнювання тексту при відображенні блокових елементів. Воно може набувати наступних значень:

| **Значення** | **Дія** |
| --- | --- |
| left | Вирівнювання по лівому краю. |
| right | Вирівнювання по правому краю. |
| center | Вирівнювання по центру. |
| justify | Вирівнювання за шириною. |
|  |  |

Додаємо для **Прикладу 10** властивість *text-align* (задані раніше властивості залишаємо без змін):

**p.ex10**{

**text-align: justify;**

**}**

Властивість text-decoration прикрашає текст при відображенні елементів. Воно може набувати наступних значень:

| **Значення** | **Відображення** |
| --- | --- |
| none | Звичайний текст. |
| underline | Підкреслений текст. |
| overline | Надкреслений текст. |
| line-through | Перекреслений текст. |
| blink | Блимаючий текст. |
|  |  |

Додаємо для **Прикладу 10** властивість *text-decoration* (задані раніше властивості залишаємо без змін):

**p.ex10{**

**text-indent: 3pt;**

**text-align: justify;**

**text-decoration: line-through;**

**}**

Властивість letter-spacing задає інтервал між буквами при відображенні тексту. Його значення задається одним з наступних способів:

| **Значення** | **Відображення** |
| --- | --- |
| normal | Звичайний текст. |
| <розмір> | Задає інтервал в додаток до стандартного інтервалу. Це значення може бути негативним. |

Додаємо для **Прикладу 10** властивість *letter-spacing* (задані раніше властивості залишаємо без змін):

**p.ex10{letter-spacing: 2em;}**

Властивість text-transform задає перетворення тексту при відображенні елементів. Воно може набувати наступних значень:

| **Значення** | **Дія** | **Приклад** |
| --- | --- | --- |
| none | Без перетворення. | Це приклад. |
| capitalize | Робити першу літеру кожного слова прописною. | Це Приклад. |
| uppercase | Виводити текст прописними літерами. | ЦЕ ПРИКЛАД. |
| lowercase | Виводити текст рядковими літерами. | це приклад. |

До **Прикладу 10** додати властивість text-transform з чотирма варіантами значень, результатом виконання яких є наведені приклади.

**РІЗНОВИДИ СЕЛЕКТОРІВ**

1. **ГРУПУВАННЯ СЕЛЕКТОРІВ.** При розробці стилю сайту виникає ситуація, коли використовуються параметри, що повторюються. Для того, щоб ці параметри не повторювалися використовується групування селекторів за певними властивостями.

| **без групування** | **з групуванням** |
| --- | --- |
| H1 {  font-family: Arial, Helvetica, sans-serif;  font-size: 160%;  color: #000aff;  }  H2 {  font-family: Arial, Helvetica, sans-serif;  font-size: 135%;  color: #aaaaaa;  } | H1, H2 {  font-family: Arial, Helvetica, sans-serif;  }  H1 {  font-size: 160%;  color: #000aff;  }  H2 {  font-size: 135%;  color: #aaaaaa;  } |
| h4{  font-weight:bold;  text-align:center;  }  #a h4{  font-style: italic;  }  #b h4{  font-style: italic;  } | h4{  font-weight:bold;  text-align:center;  }  #a h4, #b h4{  font-style: italic;  } |

1. *\**

\* {margin: 0; padding: 0; }

Цей CSS-селектор виділяє кожен елемент на сторінці. Багато розробники використовують його для того, щоб скинути у всіх елементів значення margin і padding. На перший погляд це зручно, але все-таки в робочому коді так краще не робити. Цей CSS-селектор занадто сильно вантажить броузер.

\* Також можна використовувати для виділення дочірніх елементів.

#container \* {border: 1px solid black; }

В даному випадку виділяються всі дочірні елементи #container. Знову ж, намагайтеся не зловживати ним.

сумісність

1. *Х Y*

li a {text-decoration: none; }

CSS-селектор дочірніх елементів зустрічається найчастіше. Якщо вам треба виділити елементи певного типу з множини елементів, використовуєте цей селектор. Наприклад, вам треба виділити всі посилання, які знаходяться в елементі li. У цьому випадку використовуйте цей селектор.

1. *Х + Y*

ul + p {color: red; }

Виділяє наступний елемент. Він буде вибирати *тільки* перший елемент типу Y, який йде відразу після елемента Х. У прикладі текст першого абзацу після кожного ul буде червоного кольору.

1. *Х>**Y*

div # container> ul {border: 1px solid black; }

Різниця між стандартними Х Y і X> Y полягає в тому, що даний CSS-селектор буде вибирати тільки безпосередні дочірні елементи. Наприклад, розглянемо наступний код.

<Div id = "container">

<Ul>

<Li> Приклад

<Ul>

<Li> Дочірній елемент </ li>

</ Ul>

</ Li>

<Li> Приклад </ li>

<Li> Приклад </ li>

<Li> Приклад </ li>

</ Ul>

</ Div>

CSS-селектор #container> ul вибере тільки ul-и, які є безпосередніми дочірніми елементами div з id = container. Він не вибере, наприклад, ul-и, які є дочірніми елементами перших li.

Тому можна отримати виграш в продуктивності використовуючи даний CSS-селектор. Справді, це особливо рекомендується при роботі з jQuery або іншими бібліотеками, які обирають елементи на основі правил CSS -селекторов.

1. *Х ~ Y*

ul ~ p {color: red; }

Цей CSS-селектор дуже схожий на X + Y, однак, є менш суворим. При використанні ul + p буде вибрати тільки перший елемент, що йде за Х. В даному випадку будуть вибрані всі елементи p, що йдуть за ul.

1. *X [title]*

a [title] {color: green; }

В CSS-селекторах також можна використовувати атрибути. Наприклад в даному прикладі ми виділили всі посилання, які мають атрибут title. Решта посилання залишаться не порушеними.

1. X [href = "foo"]

a [href = "http: // everstudent .ua"] {color: # ffde00; }

Всі посилання, які посилаються на everstudent.ua будуть золотими. Всі інші посилання залишаться без змін.

 Зверніть увагу на лапки. Не забудьте так само робити в jQuery і інших JavaScript бібліотеках, в яких елементи вибираються по CSS-селекторів.

1. *X [HREF \* = "everstudent.ua"]*

a [href \* = "everstudent"] {color: # 1f6053; }

Зірка позначає, що шукане значення має з'являтися *будь*-*де* в атрибуті. Таким чином, CSS-селектор охоплює *everstudent.ua, http://everstudent.ua/portfolio* і т.д.

Але що робити, якщо посилання веде на якийсь сторонній і не пов'язаний ресурс, в адресі якого присутній everstudent? Тоді потрібно використовувати "^" або "&", для посилання на початок і кінець рядка відповідно.

1. X [href ^ = "http"]

a [href ^ = "http"] {

background: url (path / to / external / icon.png) no-repeat;

padding-left: 10px;

}

Ви ніколи не замислювалися, як деякі веб-сайти можуть відображати маленький значок поряд із зовнішніми посиланнями? Я впевнений, що ви бачили їх раніше, вони добре запам'ятовуються.

"^" - найбільш часто використовується у регулярних виразах символ. Він використовується для позначення початку рядка. Якщо ми хочемо охопити всі теги, у яких href починається з http, нам треба використовувати CSS-селектор, наведений вище.

 Зверніть увагу, що ми не шукаємо "http: //". Це не правильно, оскільки не враховуються адреси, що починаються з https: // 

А якщо ми хочемо задати стиль тільки для посилань, що ведуть на фотографію? Потрібно шукати *кінець* рядка.

1. *X [href $ = ". jpg"]*

a [href $ = ". jpg"] {color: red; }

Знову ж таки, ми використовуємо символ регулярного виразу "$" для позначення кінця рядка. В даному прикладі ми шукаємо посилання, які посилаються на jpg-файли, або url-и, в кінці у яких міститься ".jpg".

1. *X: checked*

input [type = radio]: checked {border: 1px solid black; }

Цей псевдоклас виділяє тільки елементи перемикач або прапорець. Причому ті, які відзначені / обрані.

1. *X: after*

Псевдокласи: before та: after дуже круті. Створюється враження, що кожен день з'являються нові способи їх ефективного використання. Вони просто генерують контент навколо обраного елемента.

.clearfix: after {

content: "";

display: block;

clear: both;

visibility: hidden;

font-size: 0;

height: 0;

}

.clearfix {

\* display: inline-block;

\_height: 1%;

}

Цей *хак* використовує after щоб додати пробіл після елемента і заборонити його обтікання.

1. *X: hover*

div: hover {background: # e3e3e3; }

Цей CSS-селектор застосовує стиль до елементу, коли на нього наводиться курсор миші.

Цей CSS-селектор часто використовують для того, щоб поставити border-bottom на посилання, коли на них наводять мишкою.

a: hover {border-bottom: 1px solid black; }

 border-bottom: 1px solid black; виглядає краще, ніж text-decoration: underline;

*Більше інформації за посиланням: http://everstudent.ru/blog/htmlcss/30-css-seletors-to-memorize/*

**ПОСИЛАННЯ**

**Псевдокласи.** Властивості гіперпосилання засобами CSS можна визначати по-різному, залежно від того, відвідали вже посилання, чи активне воно, чи знаходиться покажчик миші над посиланням. Це дозволяє додати цікаві ефекти на ваш веб-сайт. Щоб оформити гіперпосилання засобами CSS, треба використовувати так звані псевдокласи.

Псевдоклас дозволяє враховувати різні стани або події при визначенні властивостей html-тега. У гіперпосилання є декілька станів та подій.

Псевдоклас : link використовується для посилань на сторінки, які користувач ще не відвідував.

a: link{

color: #0000ff;

font-weight: normal;

}

Псевдоклас: active використовується для активних посилань.

a: active {

color: #00bfff;

background-color:ffd700;

{

У цьому прикладі у посилання буде змінено колір шрифту та колір фону.

Псевдоклас: visited використовується для посилань на сторінки, які відвідав користувач.

a: visited {

color: #0000ff;

font-weight: 400;

}

Псевдоклас: hover використовується для посилань, над котрими находиться вказівник миші.

a: hover{

font-weight: normal;

text-trasform: uppercase;

letter-spacing: 10px;

}

У цьому прикладі шрифт посилання буде нормальної товщини, текст буде відображатися прописними літерами, відстань між літерами буде складати 10px.

**Видалення підкреслювання посилань.** Видалити підкреслювання посилань дуже просто. Для видалення підкреслювання достатньо встановити властивість text-decoration зі значенням none.

a: link{

color: #0000ff;

font-weight: normal;

text-decoration: none;

}

**Одиниці вимірювання**

px - прив'язані до абсолютних пікселі і тому не потрібні mm, cm, pt і pc. Використовується для максимально конкретного і точного завдання розмірів.

* 1mm (мм) = 3.8px
* 1cm (см) = 38px
* 1pt (типографнийй пункт) = 4/3 px
* 1pc (типографська піка) = 16px

Не відповідають реальним см., мм. чи пунктам. 1см=2.54 дюйми, 1 дюйм=72 pt.

em - задає розмір щодо шрифту батька, можна відносно конкретних символів: "x" (ex) і "0" (ch), використовується там, де потрібно спростити масштабування компоненти.

rem - задає розмір щодо шрифту <html>, використовується для зручності глобального масштабування: елементи які планується масштабувати, задаються в rem, а JS змінює шрифт у <html>.

% - Відносно такої ж властивості батька (як правило, але не завжди), використовується для ширин, висот і так далі, без нього нікуди, але треба знати, щодо чого він вираховуються відсотки.  
 vw, vh, vmin, vmax - щодо розміру екрана.