**Якорі**

Якір – це вид посилання, котрий знаходиться на одній сторінці, але в різних місцях. Частише за все він застосовується для того, щоб зробити навігацію по сторінці зручнішою.

Зараз я наведу приклад , а ви подивитесь , що відбудеться:

Структура нашої сторінки , а точніше тіла ,на даному етапі буде наступна:

1. Список
2. Параграфи з тескстом (який ми тут не будемо писати в повному обсязі, щоб зекономити місце, але в вас повинного його буди багато)

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8"/>

<title> якорі</title>

</head>

<body >

<ol> Зміст

<li> Перший пункт</li>

<li>Другий пункт</</li>

<li><a href=”#a3”>Третій пункт</a></</li>

</ol>

<p>Якийсь текст по першому пункту</p>

<p>Текст по другому пункту</p>

<p><a id=”a3”>Текст по третьому пункту</a></p>

</body>

</html>

Оновивши сторінку, ми побачимо, що у нашому списку 3й пункт в нас підсвітився. Чому? Тому що ми зробили посилання <a href=”#a3”>.

Якщо ж ми клікнемо на наше посилання, ми одразу перейдемо до п.3 нашого списку. Чому? Тому що, ми зробили якір до тексту, який прив’язаний до нашого пункту <a id=”a3”>

Таким чином ми можемо зробити посилання від будь-якої точки сторінки до будь-якої іншої точки сторінки.

Доречі, пусті посилання робляться наступним чином <a href=”#”> - це посилання буде підсвічуватись, але не буде нікуди вести.

*Не можна надавати якорю id яке складається лише з цифр. Не можна розділяти id пробілами.*

Тепер розглянемо ситуацію, коли ми бажаємо зробити посилання на іншу сторінку.

Беремо той самий код і трішки змінюємо його:

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8"/>

<title> Посилання</title>

</head>

<body >

<ol> Зміст

<li> Перший пункт</li>

<li>Другий пункт</</li>

<li><a href=”http://www.lgs.lviv.ua”>Третій пункт</a></</li>

</ol>

<p>Якийсь текст по першому пункту</p>

<p>Текст по другому пункту</p>

<p > Текст третього пункту</p>

</body>

</html>

Тепер при кліку на третій пункт нашого змісту, ми перейдемо на сторінку <http://www.lgs.lviv.ua>

Тобто, ми зробили посилання на зовнішню сторінку. У випадку з якорем, ми робили посилання на внутрішню сторінку (ту на якій знаходимось)

***Звертання до вкладених елементів***

Розглянемо наступну структуру:

<body>

<div class="outside">

<div class="inside">

<p></p>

</div>

</div>

<p></p>

<p></p>

</body>

Якщо нам потрібно звернутись до будь-якого з елементів нашої структури, ми можем використовувати .ХХХ (звернення через класс), але наш параграф не буде мати класу або індинтифікатору, а нам необхідно звернутись саме до тегу <p> який лежить у блоку і змінити його колір?

Якщо у цсс-файлі ми звернемось через

p{

color:red;

}

То в нас всі параграфи приймуть встановленні значення.

Що робити?

Для цьго існує наступний синтаксис:

div.outside>div.inside>p

Що це означає?

Дослівно: я звертаюсь до тегу <p>, котрий лежить у блоку з класом inside, котрий в свою чергу лежить у блоку з класом outside.

В такому контексті ми можем звертатись до будь-якого елементу як би глибоко він не лежав.

Додаткові джерела по селекторах:

<http://htmlbook.ru/css/selector>

***Псевдокласс :hover***

Псевдоклас ***:hover*** під’єднаний до окремого елементу при наведенні курсору змінює його властивості.

ВАЖЛИВО:

Синтаксис псевдокласу наступний і тільки такий:

***Елемент:hover –*** без пробілів!!!!

p.boy:hover{

color:red;

}

**Додаткове джерело по псевдокласах**: <http://exabyte.ws/css/visited/>

**Висота та ширина.**

height – висота.

width – шрина

Ці властивості можуть приймати значення у пікселях, відсотках, та значення auto, або inherit

І вони задають розміри для блочних елементів сторінки.

**Рамки, border & outline.**

Властивість border дозволяє одночасно встановити товщину, стиль і колір межі навколо елемента. Значення можуть йти в будь-якому порядку, розділяючись пробілом, браузер сам визначить, яке з них відповідає потрібному властивості.

Також можна все записувати і окремими властивостями як:

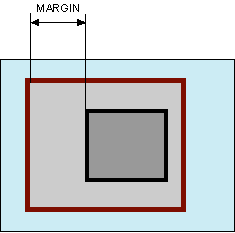
 border-top, border-bottom, border-left, border-right.

border-width – визначає товщину

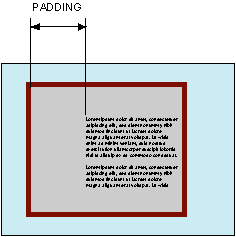
 border-style – визначає стиль

outline- одночасно встановлює колір, стиль і товщину зовнішнього кордону на всіх чотирьох сторонах елементу. І не впливає на його розмір і положення. На відмінну від border властивість outline не може задаватись на одній із чотирьох сторін елементу.

margin- встановлює величину відступу від кожного краю елемента. Відступом є простір від кордону поточного елемента до внутрішньої межі його батьківського елементу.



padding- встановлює значення полів навколо вмісту (контенту) елементу. Полем називається відстань від внутрішнього краю рамки елемента до уявного прямокутника, що обмежує його вміст.



**div контейнери**

Один із найважливіших тегів.

Елемент <div> є блоковим елементом і призначений для виділення фрагмента документа з метою зміни виду вмісту. Як правило, вид блоку описується за допомогою стилів.

Як і при використанні інших блочних елементів, вміст тегу <div> завжди починається з нового рядка.

Не забувайте про те, що цей тег не буде видно, якщо не задати йому значення height та width, background або значення у відсотковому вигляді та не вкласти в цей контейнер наприклад зображення котре буде мати якісь значення по ширині та висоті, або не задати йому контент.

float- визначає,по якій стороні буде вирівнюватися елемент, при цьому інші елементи будуть обтікати його з інших сторін. Коли значення властивості float: none, елемент виводиться на сторінці як зазвичай, при цьому допускається, що один рядок обтікаючого тексту може бути на тій самій лінії, що і сам елемент.  
Ця властивість вибиває елемент з потоку створення елементів на сторінці, і інші елементи перестають його відчувати, і може статися так, що ті елементи, що створені уже після елементу з властивістю float будуть зображені наче під ним.  
В такому разі допомагає метод який називають clearfix.

clear – ця властивість допомагає, підчистити випадання елементів з потоку, щоб знову вернути його на місці, при тому не втративши позиціонувань елементів.

**Додаткові джерела**

<http://htmlbook.ru/css/height>

<http://htmlbook.ru/css/width>

<http://htmlbook.ru/css/float>

<http://htmlbook.ru/css/margin>

<http://htmlbook.ru/css/padding>

<http://htmlbook.ru/css/border>

<http://htmlbook.ru/css/outline>

[**http://htmlbook.ru/blog/shirina-bloka**](http://htmlbook.ru/blog/shirina-bloka) **- обов’язково почитати!**

[**http://htmlbook.ru/samlayout/blochnaya-verstka/pozitsionirovanie-elementov**](http://htmlbook.ru/samlayout/blochnaya-verstka/pozitsionirovanie-elementov) **- позиціонування**