# Основные теги

<!--

Тег <html> является контейнером, который заключает в себе все содержимое веб-страницы, включая теги <head> и <body>.

Открывающий и закрывающий теги <html> в документе необязательны, но хороший стиль диктует непременное их использование.

Тег <head> содержит в себе информацию о странице, но ни в коем случае не контент страницы. В нем располагаются метатеги,

ссылки на подключаемые модули.

Тег <title> содержит имя страницы, которое отображается пользователю на вкладке в браузере.

Тег <body> является контейнером для всего содержимого, которое будет отображено пользователю.

-->

## Комментарии

<!--Комментарии могут быть вставлены в любом месте, браузер не интерпретирует их пользователю-->

## Br

Для реализации переноса строк существует специальный тег br - brake row

<br />

Он не имеет содержимого, потому в конце видим символ /

<br />

С его помощью, к примеру, можно реализовать отображение стихотворения:

<br />

<br />

## Pre

Он является контейнером, который отображает содержимое, учитывая все пробельные символы, указанные внутри себя.

<br />

С его помощью, к примеру, можно реализовать отображение стихотворения:

<br />

<br />

<pre>

Поверхностью морей отражена,

Богатая Венеция почила,

Сырой туман дымился, и луна

Высокие твердыни осребрила.

Чуть виден бег далекого ветрила,

Студеная вечерняя волна

Едва шумит вод веслами гондолы

И повторяет звуки баркаролы.

</pre>

## Align

<p align="center">

## Hr

<!--

hr - тег, для реализации прямой линии, он одиночный, не имеет содержимого и потому не имеет закрывающего тега.

Автоматически растягивается на всю ширину родительского тега.

Атрибуты: align, color, size, width, noshade

-->

<p>Тэг hr определяет горизонтальную линию:</p>

<hr />

## Color

<p>This is a paragraph</p>

<hr color="Green" />

## Hr width size noshade

<!--

width - атрибут тега hr, который указывает ширину прямой.

Следует задавать в px, в связи с тем, что некоторые браузеры( к примеру устаревшие версии Internet Explorer) не воспринимают

значения без единиц измерения.

-->

<p>Тэг hr определяет горизонтальную линию:</p>

<hr width="600px" align="left" />

<p>Тэг hr определяет горизонтальную линию:</p>

<hr color="Green" size="1px" />

<!--

noshade - атрибут тега hr, который контролирует тень прямой.

-->

<p>Тэг hr определяет горизонтальную линию:</p>

<hr noshade="noshade" />

## Strong big small b

* <strong ></strong> - содержимое, помещенное внутрь этого тега, приобретает полужирное очертание. Возможно вложение этого тега в другие теги для работы с текстом.
* 2.<big></big> - размер содержимого, помещенное внутрь этого тега, увеличивается на единицу. Возможно вложение этого тега в другие теги для работы с текстом.
* 3.<small ></small> - размер содержимого, помещенное внутрь этого тега, увеличивается на единицу. Возможно вложение этого тега в другие теги для работы с текстом.
* 4.<b ></b> - содержимое, помещенное внутрь этого тега, приобретает жирное очертание. Возможно вложение этого тега в другие теги для работы с текстом.

I sub sup em

* <i></i> - содержимое, помещенное внутрь этого тега, приобретает курсивное очертание. Возможно вложение этого тега в другие теги для работы с текстом.
* 6. <em></em> - содержимое, помещенное внутрь этого тега, приобретает акцентированное состояние. Акцент выражается курсивом. Возможно вложение этого тега в другие теги для работы с текстом.
* 7. <sub ></sub> - содержимое, помещенное внутрь этого тега, переходит в нижний индекс предыдущего текстового содержимого. Возможно вложение этого тега в другие теги для работы с текстом.
* 8. <sup></sup> - содержимое, помещенное внутрь этого тега, переходит в верхний индекс предыдущего текстового содержимого. Возможно вложение этого тега в другие теги для работы с текстом.

Marquee

Тег MARQUEE - это контейнер, который все текстовое содержимое внутри себя реализует в бегущую строку

атрибут loop указывает количество итераций, то есть количество раз, сколько пробежит строка.

-->

<MARQUEE>текст, но заключенный в тег MARQUEE</MARQUEE>

<MARQUEE loop="3">текст, но заключенный в тег MARQUEE с атрибутом loop равным 3</MARQUEE>

## Style

<!--

Атрибут style - является первой формой использования CSS, он содержит в себе стили, которыми намного комфортнее работать, в том числе

и с текстом. Основные свойства для работы с текстом:

font-family: - шрифт, которым будет отображен текст

font-size: - размер шрифта

color: - цвет шрифта

В атрибуте style можно указать как один параметр, так и несколько, главное разделять их ";" и значения передавать после":"

-->

<h1 style="font-family:verdana;">A heading</h1>

<p style="font-family:courier new; color:red; font-size:20px;">A paragraph</p>

## A href

<a href = “ # ”></a>

-

это контейнер, который все текстовое содержимое

внутри себя делает ссылкой

.

У него есть следующие атрибуты:

href

,

name

,

target

,

title

,

rel

.

* 1.href - атрибут, который содержит в себе адрес страницы, на которую ведет ссылка # - ссылка на страницу, где раcположен данный элемент.
* 2.name - атрибут, который содержит в себе имя данной ссылки, он может быть практически у любого элемента разметки.
* 3.target - атрибут, который говорит о том, каким образом открывать ссылку: \_self - открыть в этом же окне, по умолчанию, \_blank - открыть на новой вкладке.
* 4.title – атрибут, который устанавливает тултип на ссылку(выскакивающее окошко с подсказкой).
* <a name="D2" href="http://edu.cbsystematics.com/">Ссылка, ведущая на внешний ресурс, указанный в атрибуте href</a>
* <br />
* <a name="coreI3" href="001\_Tag\_a.htm">Ссылка, ведущая на внутренний ресурс, указанный в атрибуте href</a>
* <br />
* <a name="changed\_window" href="#">Ссылка, ведущая на себя</a>

## mailto

С помощью ссылки можно реализовать посылку почты

mailto:( кому послать )someone@gmail.com? (указываем параметры) subject ( имя параметра ) = Hello%20again ( значение )

-->

<p>

Это ссылка:

<a href="mailto:someone@gmail.com?subject=Hello%20again">Отправить почту</a>

</p>

<p>

## Anchor referenc to text

<p>

<a href="#C4">See also Chapter 4.</a>

<a href="#C13">See also Chapter 13.</a>

</p>

<h2><a name="C4">Chapter 4</a></h2>

<p>This chapter explains something</p>

# Image

Тег<img>-предназначендлявставкиизображенийвразметку.

Тегнеимееттекстовогосодержимогоипотомуявляетсясамодостаточным.В связисэтимотсутствуетзакрывающийтег,ивконцеоткрывающегомывидим "/".

Элементимеетследующиеатрибуты:

* 1.Src–source,источник,откудаподгружаетсяизображение.
* 2.Alt–указываетальтернативныйтекст
* 3.Title–задаетвсплывающуюподсказку.
* 4.Width,height–контролируютразмерыизображения.
* 5.Align–указываетположениеизображение.

## Src tittle

<img src="Images/constr4.gif" alt="Deleted" title="ImageExample" />

## Local and network images

Изображение из локального ресурса:

<br />

<img src="Images/constr4.gif" alt="ImageExample" title="ImageExample" />

<br />

Изображение из ресурса в сети:

<br />

<img src="http://edu.cbsystematics.com/Images/arrow.png" alt="ImageExample" title="ImageExample" />

<br />

Анимация:

<br />

<img src="Images/hackanm.gif" alt="ImageExample" title="ImageExample" />

## Widt and Height

<!--

Атрибуты width и height контролируют размеры изображения.

Если один из них отсутствует, то изображение растянется пропорцыонально второму.

Если в результате изменения изображения пропорции будут нарушены, то изображение может быть отображено некоректно.

По умолчанию изображение принимает свои действительные размеры.

-->

<html>

<head>

<title>Атрибуты тега Img - width, height</title>

</head>

<body>

<p>

<img src="images/hackanm.gif" width="20" height="20" alt="ImageExample" title="ImageExample" />

</p>

<p>

<img src="images/hackanm.gif" width="45" height="45" alt="ImageExample" title="ImageExample" />

</p>

## Photo in text. Align middle

Может принимать следующие значения для вертикального выравнивания изображения:

bottom - указывает на выравние изображения по нижней направляющей остального контента ( по умолчанию )

middle - указывает на выравние изображения по центральной направляющей остального контента

top - указывает на выравние изображения по верхней направляющей остального контента

Не имеет значения где размещено изображение, в начале текста, в середине или в конце.

-->

<html>

<head>

<title>Атрибут тега Image - align(вертикальные значения)</title>

</head>

<body>

<p>

Картинка

<img src="images/hackanm.gif" align="bottom" width="48" height="48" alt="ImageExample" title="ImageExample" />

в тексте

</p>

<p>

Картинка

<img src="images/hackanm.gif" align="middle" width="48" height="48" alt="ImageExample" title="ImageExample" />

в тексте

</p>

## Image aligh right, left

Атрибут align указывает положение изображение относительно остального контента.

Может принимать следующие значения для горизонтального выравнивания изображения:

right - указывает на выравние изображения правому краю относительно остального контента

left - указывает на выравние изображения по краю левому относительно остального контента

-->

<html>

<head>

<title>Атрибут тега Image - align(горизонтальные значения)</title>

</head>

<body>

<p align = "justify">

<img src="images/hackanm.gif" align= "right" width="48" height="48" alt="ImageExample" title="ImageExample" />

## Image references

<body>

<a href="http://edu.cbsystematics.com/">

<img src="Images/logo.png" alt="Deleted" title="CyberBionic Systematics" border ="0" />

</a>

</body>

## Background-color

<body bgcolor="blue" style="background-color:yellow">

Здесь распологается основное содержимое страницы которое будет отображено пользователю в окне браузера.

</body>

## Background-image

<body style = "background-image:url(../001\_Images/Images/background.jpg)" >

Здесь распологается основное содержимое страницы, которое будет отображено пользователю в окне браузера.

</body>

## Background-repeat

<!--

Свойство background-repeat отвечает за вид повторения фона.

может принимать следующие значения: no-repeat, repeat-x, repeat-y

no-repeat - говорит о том что повторять изображение не стоит, и размещает его один раз согласно его размерам

repeat-x - делает повторение изображение только по оси х

repeat-y - делает повторение изображение только по оси y

-->

<html>

<head>

<title>Свойство background-repeat</title>

</head>

<body background="../001\_Images/Images/background.jpg" style="background-repeat:no-repeat">

Здесь распологается основное содержимое страницы которое будет отображено пользователю в окне браузера.

</body>

Background-size

<!--

Свойство background-size - отвечает за размещение изображения на заднем фоне

Может принимать следующие значения:

Cover - масштабирует изображение с сохранением пропорций таким образом, чтобы картинка целиком поместилась внутрь блока.

Contain - масштабирует изображение таким образом, чтобы оно полностью заполнило блок, при этом к двум его сторонам изображение

будет прилегать обязательно.

-->

<html>

<head>

<title>Свойство background-size</title>

</head>

<body style=" background-image:url(../001\_Images/Images/background.jpg); background-size:cover; background-repeat:no-repeat;">

<!-- <body style=" background-image:url(../001\_Images/Images/background.jpg); background-size:contain; background-repeat:no-repeat;">

-->

</body>

## Img map

<!--

Карта-изображение - это представление картинки, дополненой областями размещенной на ней с возможностью реализации

ссылок при нажатии на эти области.

Есть основных шага по созданию карт-изображений:

1. Вставить на страницу изображение

2. Привязать к нему описание областей указав атрибут usemap

3. Создать элемент <map></map> с соответствующим именем и id (в разных браузерах воспринимается тот или иной атрибут)

4. Описать области в тегах area

Тег area, имеет следующие атрибуты:

href - ссылка на документ, на которые следует перейти

shape - форма области

coords - координаты области

alt, title -текст который будет в сплывающей подсказке

-->

<html>

<head>

<title>Image Map - карта-изображение</title>

</head>

<body>

<img src="images/imageformap.jpg" usemap="#mymap"/>

<map name="mymap" id="mymap">

<area href="#" shape="rect" coords="26, 23, 197, 177" alt="first rect" title="first rect" />

<area href="#" shape="rect" coords="494, 199, 669, 372" alt="second rect" title="second rect" />

</map>

</body>

## Image shape map

<body>

<img src="images/imageforarea.jpg" usemap="#mymap"/>

<map name="mymap" id="mymap">

<area href="#" shape="rect" coords="22, 24, 198, 186" alt="rectangle" title="rectangle" />

<area href="#" shape="circle" coords="535, 109, 71" alt="circle" title="circle" />

<area href="#" shape="poly" coords="287, 237, 290, 329, 375, 359, 429, 280, 372, 208, 331, 220" alt="polygon" title="polygon" />

</map>

</body>

## Photo map

<body style=" background-color:Silver; text-align:center;">

<img src="images/Stanley\_Park.jpg" alt ="Stanley Park New York 2002" title ="Stanley Park New York 2002" width="1000px" usemap="#StanleyPark" />

<map id="StanleyPark" name="StanleyPark">

<area shape="rect" coords="550, 160, 670, 340" alt="Monument" title="Monument" href="#" />

<area shape="rect" coords="440, 150, 1000, 250" alt="City" title="City" href="#" />

<area shape="circle" coords="200, 222, 20" alt="Factories" title="Factories" href="#" />

<area shape="poly" coords="1, 256, 42, 253, 999, 390, 999, 470" alt="Cycling Road" title="Cycling Road" href="#"/>

<area shape="poly" coords="0, 40, 114, 42, 136, 128, 160, 163, 164, 234, 120, 247, 110, 257, 0, 257" alt="Trees" title="Trees" href="#"/>

<area shape="poly" coords="2, 259, 996, 466, 1000, 499, 0, 499" alt="Grass" title="Grass" href="#"/>

</map>

# Table

<h4>

Две строки и три колонки:</h4>

<table border="1">

<tr>

<td>

100

</td>

<td>

200

</td>

<td>

300

</td>

</tr>

<tr>

<td>

400

</td>

<td>

500

</td>

<td>

600

</td>

</tr>

</table>

## Th

th - вид ячеек, которые являются заглавными.

При этом их содержимое автоматически приобретает полужирное начертание и центрируется внутри ячейки.

Также семантически эти ячейки уже будут важнее простых.

-->

<html>

<head>

<title>Тег th</title>

</head>

<body>

<h4>Заголовки в таблице:</h4>

<table border="1">

<tr>

<th>Имя</th>

<th>Тел.</th>

<th>Телефон</th>

</tr>

<tr>

<td>Alex</td>

<td>555 77 854</td>

<td>555 77 855</td>

</tr>

</table>

## Caption

<!--

Тег Caption устанавливает название таблицы, которое отображается сверху над таблицей по центру.

-->

<html>

<head>

<title>Тег caption</title>

</head>

<body>

<h4>Заголовки в таблице:</h4>

<table border="1">

<caption>Контактные данные</caption>

<tr>

<th>Имя</th>

<th>Телефон</th>

<th>Телефон</th>

</tr>

<tr>

<td>Alex</td>

<td>555 77 854</td>

<td>555 77 855</td>

</tr>

</table>

## Border

<!--

Атрибут border управляет толщиной рамки таблицы как вокруг нее, так и внутри между ячейками.

-->

<html>

<head>

<title>Атрибут border</title>

</head>

<body>

<h4>Нормальная рамка:</h4>

<table border="1">

<tr>

<td>First</td>

<td>Row</td>

</tr>

<tr>

<td>Second</td>

<td>Row</td>

</tr>

</table>

## &nbsp

<!--

Тег td не допускает пустого содержимого.

Для того, чтобы ячейки отображались корректно, необходимо их заполнить чем-то.

Классическим подходом является вставка пробельного символа &nbsp.

-->

<html>

<head>

<title>Пустые ячейки</title>

</head>

<body>

<table border="1">

<tr>

<td>Some text</td>

<td>Some text</td>

</tr>

<tr>

<td>&nbsp</td>

<td>Some text</td>

</tr>

</table>

## Colspan

<!--

Атрибут colspan позволяет обьединять ячейки.

Для этого необходимо уменьшить количество ячеек в строке, и у одной установить соответствующий colspan.

-->

<html>

<head>

<title>Атрибут colspan</title>

</head>

<body>

<h4>Таблица с объединенными колонками:</h4>

<table border="1">

<tr>

<th>Name</th>

<th colspan="2">Telephone</th>

</tr>

<tr>

<td>Alex</td>

<td>555 77 854</td>

<td>555 77 855</td>

</tr>

</table>

</body>

## Rowspan

<!--

Атрибут rowspan позволяет обьединять ячейки.

Для этого необходимо уменьшить количество ячеек в строке, и у одной установить соответствующий colspan.

-->

<html>

<head>

<title>Атрибут rowspan</title>

</head>

<body>

<h4>Таблица с объединенными колонками:</h4>

<table border="1">

<tr>

<th>First Name:</th>

<td>Bill Gates</td>

</tr>

<tr>

<th rowspan="2">Telephone:</th>

<td>555 77 854</td>

</tr>

</table>

</body>

## Cellpadding

<!--

Атрибут cellpadding отвечает за отступы внутри ячеек таблицы.

-->

<html>

<head>

<title>Атрибут cellpadding</title>

</head>

<body>

<h4>

cellpadding = 0:</h4>

<table border="1" cellpadding="0">

<tr>

<td>

First

</td>

<td>

Row

</td>

</tr>

<tr>

<td>

Second

</td>

<td>

Row

</td>

</tr>

</table>

## Cellspacing

<!--

Атрибут cellspacing отвечает за отступы между ячейками таблицы, и отступы до границы таблицы.

-->

<html>

<head>

<title>Атрибут cellspacing</title>

</head>

<body>

<h4>

cellspacing = 0:</h4>

<table border="1" cellspacing="0">

<tr>

<td>

First

</td>

<td>

Row

</td>

</tr>

<tr>

<td>

Second

</td>

<td>

Row

</td>

</tr>

</table>

## Table color

<!--

Table - это контейнер, следовательно мы можем устанавливать у него фон, как определенный цвет, так и какое-либо изображение.

-->

<html>

<head>

<title>Фон тега Table</title>

</head>

<body>

<h4>

Цвет фона:</h4>

<table border="1" style="background-color: Red">

<tr>

<td>

First

</td>

<td>

Row

</td>

</tr>

<tr>

<td>

Second

</td>

<td>

Row

</td>

</tr>

</table>

## Table image

<tr>

<td background="images/bgdesert.jpg">

Second

</td>

<td>

Row

</td>

</tr>

## Table align

<tr>

<th align="left">Наименование</th>

<th align="right">Январь</th>

<th align="right">Февраль</th>

</tr>

# Lists

## Ol

<!--

Любой список должен находиться в контейнере.

Для упорядоченных списков это контейнер <ol></ol>.

Все элементы списка располагаются в тегах <li></li>.

Тип маркировки указывается в атрибуте type, который может принимать следующие значения:

A - большие латинские буквы

a - маленькие латинские буквы

I - Большие римские цифры

i - Маленькие римские цифры

По умолчанию используются арабские цифры.

-->

<html>

<head>

<title>Упорядоченные списки</title>

</head>

<body>

<h4>Numbered list:</h4>

<ol>

<li>Apples</li>

<li>Bananas</li>

<li>Lemons</li>

<li>Oranges</li>

</ol>

<h4>Letters list:</h4>

<ol type="A">

<li>Apples</li>

<li>Bananas</li>

<li>Lemons</li>

<li>Oranges</li>

</ol>

## Ul

<!--

Любой список должен находиться в контейнере.

Для неупорядоченых списков - это контейнер <ul></ul>.

Все элементы списка располагаются в тегах <li></li>.

Тип маркировки указывается в атрибуте type, который может принимать следующие значения:

disc - закрашенные кружочки

circle - пустые кружочки

square - закрашенные прямоугольники

По умолчанию используются закрашенные кружочки.

-->

<html>

<head>

<title>Неупорядоченные списки</title>

</head>

<body>

<h4>Disc bullets list:</h4>

<ul >

<li>Apples</li>

<li>Bananas</li>

<li>Lemons</li>

<li>Oranges</li>

</ul>

<h4>Circle bullets list:</h4>

<ul type="circle">

<li>Apples</li>

<li>Bananas</li>

<li>Lemons</li>

<li>Oranges</li>

</ul>

## Nested list

<!--

В содержимое элемента списка можно вкладывать еще списки, таким образом реализовать вложенные списки(подпункты).

При этом маркировка вложенного списка, если не задавать ее явно, изменится на следующую по списку.

-->

<html>

<head>

<title>Вложенные списки</title>

</head>

<body>

<h4>Вложенный список:</h4>

<ul>

<li>Coffee</li>

<li>Tea

<ol>

<li>Black tea</li>

<li>Green tea</li>

</ol>

</li>

<li>Milk</li>

</ul>

</body>

## Dl dt dd

<!--

Тег <dl></dl> - является контейнером, в котором хранится список определений.

Само определение заключается в тег <dt></dt>, а его описание в <dd></dd>.

-->

<html>

<head>

<title>Тег dl</title>

</head>

<body>

<h4>Список определений:</h4>

<dl>

<dt>Кофе</dt>

<dd>Черный горячий напиток</dd>

<dt>Молоко</dt>

<dd>Белый холодный напиток</dd>

</dl>

</body>

## Table color

<html><body>

<h2 class="example">Color Values</h2>

<table style="background-color: white;" width="100%" border="1" cellpadding="3" cellspacing="0">

<tr>

<th width="50%">Color</th>

<th width="25%">Color HEX</th>

<th width="25%">Color RGB</th>

</tr>

<tr>

<td bgcolor="#000000">&nbsp;</td>

<td>#000000</td>

<td>rgb(0,0,0)</td>

</tr>

# Divs

<body>

<div style=" background-color:Yellow">Этот текст находится в контейнере</div>

<div style=" background-color:Red">Этот текст находится в контейнере</div>

<div style=" background-color:Green">Этот текст находится в контейнере</div>

</body>

## Width and size

<body>

<div style="background-color:Yellow; width:200px; height:200px;">Этот текст находится в контейнере</div>

</body>

## Div placing

<!--

Если разместить несколько блоков подряд с явным указание ширины и высоты, то они все равно будут переноситься по строкам.

-->

<html>

<head>

<title>Тег div</title>

</head>

<body>

<div style=" background-color:Yellow; width:200px; height:200px;">Этот текст находится в контейнере</div>

<div style=" background-color:Red; width:200px; height:200px;">Этот текст находится в контейнере</div>

<div style=" background-color:Green; width:200px; height:200px;">Этот текст находится в контейнере</div>

</body>

## Float

<!--

Свойство float определяет по какому краю будет выравниваться элемент и как он будет обтекаем другими элементами.

Может принимать 3 значения:

left - элемент выравнивается по левому краю и будет обтекаем с правой стороны.

right - элемент выравнивается по правому краю и будет обтекаем с левой стороны.

none – обтекание элемента не задается.

-->

<body>

<div style=" background-color:Yellow; width:200px; height:200px; float: right">Этот текст находится в контейнере</div>

<div style=" background-color:Green; width:200px; height:200px; float: right">Этот текст находится в контейнере</div>

<div style=" background-color:Red; width:200px; height:200px; float: right">Этот текст находится в контейнере</div>

</body>

## Clear – prohibition of floating

<!--

Свойство clear – отменяет обтекание или запрещает обтекание элемента с указанной стороны.

-->

<html>

<head>

<title>Свойство Clear</title>

</head>

<body>

<div style="background-color: Yellow; width: 200px; height: 200px; float: left">Этот текст находится в контейнере</div>

<div style="background-color: Green; width: 200px; height: 200px; float: left">Этот текст находится в контейнере</div>

<div style="background-color: Red; width: 200px; height: 200px; float: left; clear:both ;">Этот текст находится в контейнере</div>

</body>

## Padding

<!--

Свойство Padding отвечает за отступы внутри элемента, от границ до его содержимого.

-->

<html>

<head>

<title>Свойство padding</title>

</head>

<body>

<div style=" width:200px; height:200px; background-color:Yellow; padding: 30px;">

содержимое будет отступать от границ на 30px

</div>

</body>

## Margin

<body style=" background-color:Blue">

<div style=" width:200px; height:200px; background-color:yellow; margin:0px">

Отспуп от соседнего блока составляет 0px

</div>

<div style=" width:200px; height:200px; background-color:Yellow; margin:30px">

Отспуп от соседнего блока составляет 30px

</div>

</body>

## Margin parametes

<!--

Свойство margin может принимать от одного до 4 параметров:

1 - этот отступ будет установлен для всех сторон одинаково

2 - первый будет устанавливать отступы по вертикали(сверху и снизу), второй по горизонтали(слева-справа)

3 - первый устанавливает отступ сверху, второй - слева и справа, третий - снизу

4 - первый устанавливает отступ сверху, второй - справа, третий - снизу, четвертый - слева

-->

<html>

<head>

<title>Тег div</title>

</head>

<body>

<div style=" background-color:Blue; width:400px; height:400px;">

<div style=" width:200px; height:200px; background-color:Yellow; margin:200px auto">

содержимое тега

</div>

</div>

<div style=" background-color:Blue; width:400px; height:400px;">

<div style=" width:200px; height:200px; background-color:Yellow; margin:10px 20px 30px">

содержимое тега

</div>

</div>

<div style=" background-color:Blue; width:400px; height:400px;">

<div style=" width:200px; height:200px; background-color:Yellow; margin:50px 20px 30px 40px">

содержимое тега

</div>

</div>

# Span

<!--

Тег <span> предназначен для форматирования текста документа. В отличие от блочных элементов, с помощью тега <span>

можно выделить часть информации внутри других тегов и установить для нее свой стиль.

-->

<html>

<head>

<title>Тег Span</title>

</head>

<body>

<span style="color:red; font-size: 40px; ">Э</span>тот параграф демонстрирует применение тегов

<span style="color:blue">span</span>, с их помощью установлен <span style="color:Green"> цвет </span>

различным частям документа, а так же задан <span style="color:red; font-family: Verdana; "> размер текста и его стиль.</span>

</body>

# CSS

## Style

<!--

Самым простым способом работы с CSS является атрибут style.

-->

<html>

<head>

<title>атрибут style</title>

</head>

<body>

<p style="font-family:Segoe UI; font-size:30pt; color:Green"> Первый способ задать CSS свойства</p>

</body>

## Style for tag

<!--

Тег style предназначен для создания стилей. Все описанные стили применяются к контенту тега на всех уровнях вложености.

-->

<html>

<head>

<title>Тег style</title>

<style type="text/css">

p

{

font-family:Segoe UI;

font-size:18pt;

color:Green;

}

</style>

</head>

<body>

<p style="">Простой параграф</p>

<p>Еще один простой параграф</p>

<div>

<p>Также еще один простой параграф, который находится в блочном элементе</p>

</div>

</body>

</html>

## Link on css file

<!--

Также можно создавать стили в отдельном файле с расширением .css. При этом тегу head необходимо создать ссылку на файл:

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

-->

<html>

<head>

<title>Тег style</title>

<link href="003\_CSS\_file.css" rel="stylesheet" type="text/css" />

</head>

## Class

.селектор-класс

{

свойство1: значение;

свойство2: значение;

}

Класс - особенный вид селектора, которым можно пометить как один, так и несколько элементов.

Чтобы указать селектор-класс, в CSS необходимо использовать конструкцию .имя класса

Чтобы привязать к этому классу элемент, в открывающем теге необходимо указать атрибут class = "имя класса"

-->

<html>

<head>

<title>Тег style</title>

<style type="text/css">

.class1

{

font-size: 20px;

color: red;

}

</style>

</head>

<body>

<!-- Селектор-класс доминирует над селектром тегом <p> -->

<p class="class1">Простой параграф</p>

<p>Еще один простой параграф</p>

<!-- Селектор-класс доминирует над селектром тегом <div> -->

<div class="class1" >

Также Еще один простой параграф, который находится в блочном элементе

</div>

</body>

</html>

## Id

Идентификатор - особенный вид селектора, который используется для обращения к элементу и изменения его стиля.

Идентификатор (ID селектор) определяетя уникальное имя элемента, поэтому по умолчанию предполагается,

что элемент с таким селектором на странице один.

Чтобы задать идентификатор, в CSS необходимо использовать конструкцию #имя идентификатора.

Чтобы привязать к этому идентификатору элемент, в открывающем теге необходимо указать атрибут id = "имя идентификатора".

-->

<html>

<head>

<title>Тег style</title>

<style type="text/css">

#Id1

{

font-family:Segoe UI;

font-size:18pt;

color:Green;

}

</style>

</head>

<body>

<p>Простой параграф</p>

<p id="Id1">Еще один простой параграф №1</p>

</body>

</html>

## Стили вложенных селекторов

1. Вложенные селекторы

2. Мультиклассы

3. Соседний селектор

Вложенные селекторы

Если необходимо определить стиль для элемента, вложенного в другой, то лучшим решением является конструкция вложенных селекторов.

Первым указывается селектор родительского элемента, а вторым селектор дочернего элемента.

родительский\_элемент дочерний\_элемент {}

-->

<html>

<head>

<title>Тег Table</title>

<style type="text/css">

div

{

color: Green;

font-size:18pt;

font-family:Arial;

background-color:green;

}

div div

{

color:Blue;

background-color:aqua;

}

div p

{

background-color:Red;

}

p {

background-color:yellow;

}

</style>

</head>

<body>

<p>Простой параграф</p>

<div>

<p> Содержимое параграфа</p>

<div>

Содержимое вложеного блока

</div>

</div>

</body>

</html>

## Мультиклассы

Мультиклассы

Если необходимо создать стиль для элемента, который привязан к нескольким селекторам,

то необходимо использовать следующую конструкцию.

селектор.селектор {}

Все селекторы, которым обязан соответствовать элемент, перечисляются без пробелов по очереди.

-->

<html>

<head>

<title>Тег Table</title>

<style type="text/css">

.class1

{

color:Green;

font-size:32px;

background-color:Black;

}

p.class2

{

color:Red;

font-size: 50px;

}

.class1.class3

{

color:Blue;

background-color:red;

}

p

{

font-weight:bold;

}

</style>

</head>

<body>

<p class="class1">Содержимое простого параграфа с установленным классом class1</p>

<span class="class1">Простой строковой элемент class1</span><br />

<p class="class1">Содержимое простого параграфа с установленным классом class2</p>

<span class="class2">Простой строковой элемент class2</span><br />

<p class="class1 class3">Содержимое простого параграфа с установленным классом class1 class2</p>

</body>

## Соседний селектор

Соседними называются те элементы разметки, которые следуют непосредственно друг за другом в коде документа.

Для задания стиля соседнему элементу используется символ «+», который устанавливается между двумя селекторами.

селектор1 + селектор2

-->

<html>

<head>

<title>Тег Table</title>

<style type="text/css">

p + span + div

{

background-color:Red;

}

</style>

</head>

<body>

<p>Содержимое параграфа №1</p>

<span>Содержимое тега span</span>

<div>Содержимое тега div </div>

<p>Содержимое параграфа №2</p>

</body>

</html>

## Font family

Устанавливает семейство шрифта, которое будет использоваться для оформления текста содержимого.

Список шрифтов может включать одно или несколько названий, разделенных запятой.

если в имени шрифта содержатся пробелы, например, Trebuchet MS, оно должно заключаться в одинарные или двойные кавычки.

Когда браузер встречает первый шрифт в списке, он проверяет его наличие на компьютере пользователя.

Если такого шрифта нет, берется следующее имя из списка и также анализируется на присутствие.

Поэтому несколько шрифтов увеличивает вероятность, что хотя бы один из них будет обнаружен на клиентском компьютере.

-->

<html>

<head>

<title>Работа со шрифтом</title>

<style type="text/css">

body

{

color: #0094ff;

font-family: Verdana, Arial;

}

p.one

{

font-family: 'Times New Roman';

color:darkcyan;

}

p.two

{

font-family: 'Monotype Corsiva';

color: darkgrey;

}

p.three

{

font-family: Arial;

color:darkorange;

}

</style>

</head>

<body>

<h1> CSS стили шрифта</h1>

<table border="1" cellpadding="5">

<tr>

<th>

Шрифт

</th>

<th>

Результат

</th>

</tr>

<tr>

<td>

<p class="one" > Time New Roman </p>

</td>

## Font family 2

<head>

<title>Тег Table</title>

<style type="text/css">

body

{

color: #000001;

background-color: #ffffff;

font-family: Verdana, sans-serif;

}

p.one

{

font-family: Arial, Verdana, sans-serif;

}

p.two

{

font-family: 'Times New Roman',Arial;

}

p.three

{

font-family: 'Segoe UI', 'Courier New';

}

</style>

## Font size

Для задания размеров различных элементов, в CSS используются абсолютные и относительные единицы измерения.

Абсолютные единицы не зависят от устройства вывода, а

относительные единицы определяют размер элемента относительно значения другого размера.

- ОТНОСИТЕЛЬНЫЕ ЕДИНИЦЫ:

1. Em - Высота шрифта текущего элемента

2. Ex - Высота символа x

3. Px - Пиксел

4. % - Процент от размера шрифта по умолчанию

Относительные единицы обычно используют для работы с текстом,

либо когда надо вычислить процентное соотношение между элементами.

- АБСОЛЮТНЫЕ ЕДИНИЦЫ:

1. in - Дюйм (1 дюйм равен 2,54 см)

2. cm - Сантиметр

3. mm - Миллиметр

4. Pc - Пика (1 пика равна 12 пунктам)

5. Pt - Пункт (1 пункт равен 1/72 дюйма)

Абсолютные единицы применяются реже, чем относительные и,

как правило, при работе с текстом.

-->

<html>

<head>

<title>Свойство font-size</title>

<style type="text/css">

body

{

color:#000000;

background-color:#ffffff;

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

font-size: 20px;

}

.page {width:880px; background-color:aqua;}

.column {padding:10px; width:200px; float:left;}

hr {clear:both;}

## Font weight

Свойство font-weight - Устанавливает насыщенность шрифта.

Значение устанавливается от 100 до 900 с шагом 100.

Сверхсветлое начертание, которое может отобразить браузер, имеет значение 100, а сверхжирное — 900.

Нормальное начертание шрифта (которое установлено по умолчанию) эквивалентно 400,

стандартный полужирный текст — значению 700.

Насыщенность шрифта задается с помощью ключевых слов:

bold — полужирное начертание, bolder — жирное начертание;

lighter — светлое начертание, normal — нормальное начертание.

Также допустимо использовать условные единицы от 100 до 900.

p.one

{

font-weight: normal;

}

p.two

{

font-weight: bold;

}

p.three

{

font-weight: normal;

}

p.three span

{

font-weight: bolder;

}

p.four

{

font-weight: bold;

}

p.four span

{

font-weight: lighter;

}

p.five

{

font-weight: 100;

}

Font style

font-style - свойство отвечает за стиль шрифта и может принимать следующие значения:

1. normal - нормальный шрифт

2. italic - наклонный шрифт

3. oblique - Браузеры текст со значением oblique всегда отображают как курсив (italic).

p.one {font-style:normal;}

p.two {font-style:italic;}

p.three {font-style:oblique;}

</style>

</head>

<body>

<h1> font-style</h1>

<p class="one">

This is a normal font</p>

<p class="two">

## Font variant

font-variant определяет, как нужно представлять строчные буквы, принимая значение small-caps переводит символы

в верхний регистр.

-->

<html>

<head>

<title>Свойство font-variant</title>

<style type="text/css">

body

{

color: #000000;

background-color: #ffffff;

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

font-size: 12px;

}

p

{

font-variant: normal;

}

span.smallcaps

{

font-variant: small-caps;

}

</style>

## Font color

span.color

{

color: #ff0000;

}

</style>

</head>

<body>

<h1>color</h1>

<p class="one">

Обычный текст <span class="color">и красный</span>

посередине.</p>

</body>

## Text align

text-align - свойство, которое отвечает за выравнивание текста, может принимать следующие значения:

1. center - выравнивает текст по центру

2. left - выравнивает текст влево

3. right - выравнивает текст вправо

4. justify - выравнивает текст по допустимой ширине

-->

<html>

<head>

<title>Свойство text-align</title>

<style type="text/css">

body {

color:#000000;

background-color:#ffffff;

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

font-size:12px;}

div {width:500px;margin:10px;}

.leftAlign {text-align:left;}

.rightAlign {text-align:right;}

.center {text-align:center;}

.justify {text-align:justify;}

</style>

## Text decoration

text-decoration - свойство, которое отвечает за декорирование текста, может принимать следующие значения:

1. underline - декорирует текст подчеркиванием

2. overline - декорирует текст надчеркиванием

3. line-through - декорирует текст перечеркиванием

4. blink - декорирует текст миганием, в связи с устарением, работает не во всех браузерах

-->

<html>

<head>

<title>Свойство text-decoration</title>

<style type="text/css">

body

{

color: #000000;

background-color: #ffffff;

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

font-size: 12px;

}

table

{

width: 500px;

}

p.underline

{

text-decoration: underline;

}

p.overline

{

text-decoration: overline;

}

p.line-through

{

text-decoration: line-through;

}

</style>

## Text indent

text-indent - свойство, которое отвечает за отступ с первой строки, принимает конкретную величину.

-->

<html>

<head>

<title>Свойство text-indent</title>

<style type="text/css">

body

{

color: #000000;

background-color: #ffffff;

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

font-size: 12px;

}

.indent

{

text-indent: 3em;

}

</style>

## Text transform

text-transform - свойство, которое отвечает за преобразование текста. Может принимать следующие значения:

1. none - текст не преобразуется

2. capitalize - текст преобразуется таким образом, что все слова будут начинаться с большой буквы

3. lowercase - текст будет преобразуется в нижний регистр

4. uppercase - текст будет преобразуется в верхний регистр

-->

<html>

<head>

<title>Свойство text-transform</title>

<style type="text/css">

body

{

color: #000000;

background-color: #ffffff;

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

font-size: 12px;

}

p.none

{

text-transform: none;

}

p.capitalize

{

text-transform: capitalize;

}

p.uppercase

{

text-transform: uppercase;

}

p.lowercase

{

text-transform: lowercase;

}

</style>

## Latter spacing

letter-spacing - свойство, отвечающее за отступы между буквами. Значение указывается в явной величине, при этом могут быть использованы

отрицательные значения.

-->

<html>

<head>

<title>Свойство letter-spacing</title>

<style type="text/css">

body

{

color: #000000;

background-color: #ffffff;

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

font-size: 12px;

}

p.one

{

letter-spacing: normal;

}

p.two

{

letter-spacing: 3px;

}

p.three

{

letter-spacing: 0.5em;

}

p.four

{

letter-spacing: -1px;

}

</style>

## Word spacing

word-spacing - свойство, отвечающее за отступы между словами. Значение указывается в явной величине, при этом могут быть использованы

отрицательные значения.

-->

<html>

<head>

<title>Тег Table</title>

<style type="text/css">

body

{

color: #000000;

background-color: #ffffff;

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

font-size: 12px;

}

.one

{

word-spacing: normal;

}

.two

{

word-spacing: 20px;

}

.three

{

word-spacing: -3px;

}

</style>

## Hover and visited

В CSS существуют специальные псевдо-свойства. Они описывают не сами элементы, а их состояние. Указываются после селектора

с префиксом ":".

На примере ссылок рассматриваются следущие псевдо-свойства:

1. hover - состояние, когда на элемент наведен курсор мыши

2. visited - состояние ссылки, которая уже была посещена

-->

<html>

<head>

<title>Тег Table</title>

<style type="text/css">

a {

color: Silver;

}

a:visited

{

color: Green;

}

a:hover

{

color: red;

}

</style>

</head>

<body>

<ul>

<li><a href="http://edu.cbsystematics.com/">Домашняя страница</a></li>

<li><a href="http://edu.cbsystematics.com/ru/courses/">Курсы</a></li>

<li><a href="http://edu.cbsystematics.com/ru/contacts/">Контакты</a></li>

</ul>

</body>

# Metatags

Метатеги - это теги языка HTML, содержащие различную служебную информацию. Метатеги размещаются в заголовке страницы (между тегами <head> и

</head> и имеют такой формат:

<meta name="имя" content="значение">

Условно метатеги можно разделить на три группы:

1. содержащие управляющие команды для браузера;

2. содержащие управляющие команды для поисковых систем;

3. содержащие информацию о странице и ее авторе.

-->

<html>

<head>

<title>Простой пример html страницы</title>

</head>

<body>

Здесь распологается основное содержимое страницы, которое будет отображено пользователю в окне браузера.

</body>

## http equiv

Таких метатегов немного, но на сегодняшний день это самые важные из метатегов. О том, что данный метатег предназначен для управления браузером,

говорит атрибут http-equiv.

Content-type - этот метатег указывает браузеру, в какой кодировке сделана страница.

Content-Language - метатег сообщает, на каком языке написана страница.

pragma - этот метатег запрещает браузеру кэшировать страницу (то есть сохранять ее в памяти).

refresh - Этот метатег через N секунд после загрузки в браузер текущей страницы, перезагрузит вместо нее страницу URL (естественно, вместо N

должна быть написана цифра - количество секунд, а вместо URL - правильный адрес страницы).

-->

<html>

<head>

<title>Mетатеги, содержащие управляющие команды для браузера</title>

<meta http-equiv="Content-type" content="text/html;charset=windows-1251" />

<meta http-equiv="Content-Language" content="ru" />

<meta http-equiv="pragma" content="no-cache" />

<meta http-equiv="refresh" content="3; url=001\_Matatags.htm" />

</head>

<body>

Здесь распологается основное содержимое страницы, которое будет отображено пользователю в окне браузера.

</body>

</html>

## Meta name

Generator - сообщает, с помощью какой программы был сгенерирован код страницы.

site-created - указывает дату создания страницы.

expires - этот метатег указывает, когда страница будет удалена.

Author - содержит имя автора страницы.

Copyright - указывает владельца авторских прав.

Reply-to - указывает способ связи с автором страницы.

Owner - указывает собственника страницы.

Address - содержит адрес автора страницы.

-->

<html>

<head>

<title>Mетатеги, содержащие информацию о странице и ее авторе</title>

<meta name="Generator" content="Visual Studio" />

<meta name="site-created" content="01.27.2012" />

<meta http-equiv="expires" content="Fri, 16 April 2014 23:59:59 GMT" />

<meta name="Author" content="Alex" />

<meta name="Copyright" content="Alex Corp" />

<meta name="Reply-to" content="alex@to.ru" />

<meta name="Owner" content="Alex" />

<meta name="Address" content="alex 324" />

</head>

<body>

Здесь распологается основное содержимое страницы, которое будет отображено пользователю в окне браузера.

</body>

</html>

## SEO

Не так давно эти метатеги были очень важны, правильно используя их, можно было достаточно легко вывести свой сайт на хорошие позиции в

поисковиках.

description - содержит в себе основное описание страницы, является наиболее важным из метатегов.

keywords - содержит ключевые слова, по которым будет индексироваться страница.

robots - метатег управляет индексированием страниц.

index / noindex — индексировать / не индексировать эту страницу;

follow / nofollow — идти / не идти по ссылкам с этой страницы;

all — эквивалентно index, follow

none — эквивалентно noindex, nofollow

revisit, revisit-after - дает команду поисковой системе индексировать сайт с нужной периодичностью.

-->

<html>

<head>

<title>Простой пример html страницы</title>

<meta name="description" content="HTML примеры" />

<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript, VBScript" />

<meta name="robots" content="index, follow" />

<meta name="revisit" content="3 days" />

<meta name="revisit-after" content="3 days" />

</head>

<body>

Здесь распологается основное содержимое страницы, которое будет отображено пользователю в окне браузера.

</body>

## Процентная верстка

Процентная верстка получается за счет того, что единицами измерения при верстке дизайна

в большинстве случаев выступают проценты, что позволяет растянуть макет на всю ширину окна браузера.

-->

<html>

<head>

<title>Процентная верстка</title>

<style type="text/css">

body {

margin: 0px;

padding: 0px;

}

p.white {

color: whitesmoke;

margin: 0px;

}

div {

height: 100px;

width: 100%;

}

</style>

</head>

<body>

<div style="height: auto;">

<div style="background-color: navy;">

<p class="white">TITLE</p>

</div>

<div>

<div style="width: 30%; background-color: Gray; float: left"></div>

<div style="width: 70%; background-color: whitesmoke; float: left"></div>

</div>

<div style="background-color: darkgreen">

<p class="white">FOOTER</p>

</div>

</div>

</body>

## Статическая верстака

Ширина при статической верстке определяется из расчета того, что по статистическим данным большинство пользователей

используют мониторы с разрешением 1024х768 пикселей. Следовательно, для того, чтобы избежать появления горизонтальной полосы прокрутки

и искажения дизайна, нужно в среднем брать ширину макета примерно 800-990 пикселей.

Недостаток при таком методе верстки – это неэффективное использование свободной площади.

-->

<html>

<head>

<title>Статическая верстка</title>

<style type="text/css">

body {

margin: 0px;

padding: 0px;

}

p.white {

color: whitesmoke;

margin: 0px;

}

div {

height: 100px;

width: 800px;

margin: auto;

}

</style>

</head>

<body>

<div style="height: auto;">

<div style="background-color: navy">

<p class="white">TITLE</p>

</div>

<div>

<div style="width: 300px; background-color: Gray; float: left"></div>

<div style="width: 500px; background-color: whitesmoke; float: left"></div>

</div>

<div style="background-color: darkgreen">

<p class="white">FOOTER</p>

</div>

</div>

## Смешаная верстка

<title>Смешанная верстка</title>

<style type="text/css">

body {

margin: 0px;

padding: 0px;

}

p.white {

color: whitesmoke;

margin: 0px;

}

div {

height: 100px;

margin: auto;

/\*max-width\*/

max-width: 800px;

}

</style>

</head>

<body>

<div style="background-color: navy">

<p class="white">TITLE</p>

</div>

<div>

<div style="width: 30%; background-color: Gray; float: left"></div>

<div style="width: 70%; background-color: whitesmoke; float: left"></div>

</div>

<div style="background-color: darkgreen">

<p class="white">FOOTER</p>

</div>

</body>

</html>

## Position absolute

<style type="text/css">

div

{

border:1px solid black;

width:150px;

height:150px;

margin:15px;

position:absolute;

}

#MisterGreen

{

z-index:3;

top:30px;

left:30px;

background-color:Green;

}

#MisterRed

{

z-index:2;

top:20px;

left:20px;

background-color:Red;

}

#MisterYellow

{

z-index:1;

background-color:Yellow;

}

</style>

</head>

<body>

<div id="MisterGreen">

</div>

<div id="MisterRed">

</div>

<div id="MisterYellow">

</div>

## Position relative

<style type="text/css">

div

{

text-align:center;

width:150px;

height:150px;

position:relative;

}

#MisterGreen

{

background-color:Green;

border:1px solid black;

}

#MisterRed

{

top:30px;

left:30px;

background-color:Red;

border:1px solid black;

}

#MisterYellow

{

position: relative;

left:20px;

background-color:Yellow;

border:1px solid black;

z-index: 5;

}

.MisterWhite

{

margin:15px;

border:1px dashed black;

}

.MisterBlue

{

width: 150px;

height: 150px;

background: blue;

}

</style>

</head>

<body>

<div class="MisterWhite">

<div id="MisterGreen"> <br /><br />In Its Place </div>

</div>

<div class="MisterBlue">

<div id="MisterYellow" style=" position: relative;"> <br /><br />left:20px; </div>

</div>

<div class="MisterWhite">

<div id="MisterRed"> <br /><br />top:30px;<br /> left:30px; </div>

</div>

## Position fixed

<style type="text/css">

body

{

color:#969696;

font-family:Segoe UI;

height:1500px;

width:3000px;

}

h1

{

color:Gray;

}

#FixedDiv

{

width:200px;

height:200px;

text-align:center;

font-size:14pt;

color:White;

border:1px solid black;

background-color:Blue;

top:200px;

left:600px;

position:fixed;

}

</style>

## Border radius

<html>

<!--border-radius – позволяет закруглять границы (border) HTML-элементов-->

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Закругленные углы</title>

<style type="text/css">

body

{

Font-size: 14pt;

padding-left: 300px;

}

#border\_corner

{

background-color: #ddccb5;

border: 2px solid #897048;

padding: 10px;

margin-top: 30px;

width: 200px;

height: 125px;

border-radius: 200px/10px;

/\*-moz-border-radius: 400px;

-webkit-border-radius: 100px 20px;\*/

}

## Opacity

<head>

<title>Opacity Example</title>

<style type="text/css">

div

{

width:350px;

height:350px;

}

#Globe-PNG

{

float:left;

background-image:url(img001/Globe-picture.jpg);

}

#Globe-PNG-Opacity

{

float:left;

background-image:url(img001/Globe-picture.png);

}

#Globe

{

filter: alpha(opacity=60);

opacity: 0.6;

background-image:url(img001/Globe-picture.jpg);

}

#Globe:hover

{

filter: alpha(opacity=100);

opacity: 1.0;

}

</style>

</head>

<body>

<div id="Globe-PNG"></div>

<div id="Globe-PNG-Opacity"></div>

<div><hr width="800px" /></div>

<div id="Globe"></div>

## FavIcon

<head>

<title>Layout</title>

<link href="img001/favicon1.ico" rel="icon" type="image/x-icon" />

<link href="img001/favicon1.ico" rel="shortcut icon" type="image/x-icon" />

<link rel="icon" type="image/png" href="img001/yinyang.png" />

</head>

<body>

<head>

<title>Layout</title>

<!-- Разные способы подключения FavIcon с учетом кроссбраузерности. -->

<link href="img001/favicon1.ico" rel="icon" type="image/x-icon" />

<link href="img001/favicon1.ico" rel="shortcut icon" type="image/x-icon" />

<link rel="icon" type="image/png" href="img001/yinyang.png" />

</head>

# Forms

## Input login password

input - Тег для создания элементов управления.

атрибуты:

type - тип зоздаваемого элемента управления

name - имя, которое будет отправлтся на сервер

value - значение, которое будет отправлятся на сервер

-->

<html>

<head>

<title>Тег input</title>

</head>

<body>

<!--text - поле ввода-->

Login <input type="text" name="login" value="Введите логин" />

<!--password - поле ввода для пароля -->

Password <input type="password" name="psw" value="" />

</body>

## Checkbox

<body>

I like Bike: <input type="checkbox" name="input\_1" value="bike" />

<br />

I like Airplane: <input type="checkbox" name="input\_2" value="airplane" />

<br />

<!--checked="checked" - сделать элемент выбранным-->

I like Car: <input type="checkbox" name="input\_3" value="car" checked="checked" />

<br />

</body>

## Radio button

<body>

I will travel to London by Bike:<input type="radio" name="transport" value="bike" />

<br />

I will travel to London by Airplane:<input type="radio" name="transport" value="airplane" />

<br />

I will travel to London by Car:<input type="radio" name="transport" value="car" />

</body>

## Form

Тег form - элемент разметки для отправки на сервер значений введенных пользователем.

атрибуты элемента form:

1. action - адрес ресурса, который должен обработать данные введенные в форму

2. method - HTTP глагол (метод), которым будет отправлен запрос

3. target - указывает, как будут отображены результаты, в текущем окне либо новой вкладке

-->

<html>

<head>

<title>Тег form</title>

</head>

<body>

<form action="SomePage.aspx" method="get" target="\_blank">

Login

<input type="text" name="login" value="" />

<br />

Password

<input type="password" name="password" value="" />

<!--Кнопка для отправки данных на сервер-->

<input type="submit" value="Enter" />

</form>

## Input reset

input type="reset" - элемент формы, который реализует кнопку очистки полей формы.

При этом поля не просто очищаются от внесенных данных,

но и воcстанавливают значения по умолчанию.

-->

<html>

<head>

<title>Клавиша Reset</title>

</head>

<body>

<form>

<br />

Enter your first name: <input type="text" name="first\_name" value="Alexander" />

<br />

Enter your second name: <input type="text" name="sec\_name" />

<br />

Enter your password: <input type="password" name="password" />

<br />

<input type="submit" name="submit" value="Enter" />

<br />

<input type="reset" name="reset" value="Reset" />

</form>

## Onclick

input type="button" определяет кнопку, через атрибут onclick можно определить действие которое произойдет при клике по кнопке.

-->

<html>

<head>

<title>Создание кнопок</title>

</head>

<body>

<form>

<br />

Enter your first name:

<input type="text" name="first\_name" value="Ivan" />

<br />

Enter your second name:

<input type="text" name="sec\_name" value="Ivanov" />

<br />

Enter your password:

<input type="password" name="password" />

<br />

<input type="submit" name="submit" value="Enter" />

<br />

<input type="reset" name="reset" value="Reset" />

<br />

<input type="button" name="button" value="Say hello" onclick="alert('hello world')" />

</form>

## Select. dropdown list

Тег select - тег, контейнер, содержащий в себе описание выпадающего списка. Каждый элемент списка описан внутри с помощью отдельного тега option.

Содержимое тега option будет отображено пользователям в качестве элементов выпадающего списка, но на сервер будут отправлены значения

атрибутов value и name выбранного элемента.

selected="selected" - атрибут отвечает за элемент выбраный по умолчанию

-->

<html>

<head>

<title>Тег select</title>

</head>

<body>

<form>

Your favorite film:

<select name="select">

<option value="film001">2 smoking guns</option>

<option value="film002" >Bad Boys</option>

<option value="film003">Revenge</option>

<option value="film004" selected="selected" >Crow</option>

<option value="film005">Dead Man Walkin</option>

<option value="film006">Wrong thing</option>

</select>

## Select multy

Тег select - тег, контейнер, содержащий в себе описание выпадающего списка. Каждый элемент списка описан внутри с помощью отдельного тега option.

Содержимое тега option будет отображено пользователям в качестве элементов выпадающего списка, но на сервер будут отправлены значения

атрибутов value и name выбранного элемента.

Атрибут multiple="multiple" говорит о том, что список может позволять многозначный выбор при зажатии клавиши Shift или Ctrl

-->

<html>

<head>

<title>Многозначный список</title>

</head>

<body>

<form>

Your favorite film:

<select multiple="multiple" name="select">

<option value="film001">2 smokin guns</option>

<option value="film002">Bad Boys</option>

<option value="film003">Revenge</option>

<option value="film004">Crow</option>

<option value="film005">Dead Man Walkin</option>

<option value="film006">Wrong thing</option>

</select>

## Select group

Тег select - тег, контейнер, содержащий в себе описание выпадающего списка. Каждый элемент списка описан внутри

с помощью отдельного тега option.

Содержимое тега option будет отображено пользователям в качестве элементов выпадающего списка, но на сервер будут

отправлены именно значения атрибутов value и name.

Тег <optgroup> - предназначен для группировки элементов списка, при этом у него есть особенный атрибут label, в котором

указывается название группы.

-->

<html>

<head>

<title>Группировка эементов выпадающего списка</title>

</head>

<body>

<form>

Your favorite film:

<select name="select">

<optgroup label="Action">

<option value="film001">2 smokin guns</option>

<option value="film002">Bad Boys</option>

<option value="film003">Revenge</option>

</optgroup>

<optgroup label="Fantastic">

<option value="film004">Crow</option>

<option value="film005">Dead Man Walkin</option>

</optgroup>

<optgroup label="Other">

<option value="film006">Wrong thing</option>

</optgroup>

</select>

## Input file

input type="file" - элемент формы, предназначенный для создания поля выбора загрузки файла. В разных барузерах отображается по разному.

-->

<html>

<head>

<title>Поле для загрузки файла</title>

</head>

<body>

<form>

<br />

Enter your login: <input type="text" name="login" value="here..." />

<br />

Enter your password: <input type="password" name="password" />

<br />

<input type="file" name="input\_file" value="choose here..." />

<br />

<input type="submit" name="submit" value="Enter" />

</form>

## Input hidden

input type="hidden" - элемент формы, который не отображается пользователю, при этом передается на сервер при отправке.

Используется для передачи параметров, которые не должны быть видны либо изменяться пользователем.

-->

<html>

<head>

<title>Невидимое поле</title>

</head>

<body>

<form>

<br />

Enter your first name: <input type="text" name="first\_name" value="Ivan" />

<br />

Enter your second name: <input type="text" name="sec\_name" value="Ivanov" />

<br />

Enter your password: <input type="password" name="password" />

<input type="hidden" name="user\_id" value="1234" />

<input type="hidden" name="price" value="100" />

<input type="hidden" name="payment\_type" value="MasterCard" />

<br />

<input type="submit" name="submit" value="Enter" />

</form>

## Textarea

<textarea> - тег, предназначенный для создания многострочного поля для ввода. Имеет специальные атрибуты rows cols, которые, соответственно,

указывают его высоту и ширину.

-->

<html>

<head>

<title>Тег textarea</title>

</head>

<body>

<form>

<br />

Enter your name:

<input type="text" name="login" value="" />

<br />

Enter your message:

<br />

<textarea cols="40" rows="10" name="textarea" style ="resize:none"></textarea>

<br />

<input type="submit" name="submit" value="Enter" />

</form>

## Fieldset and legend

<fieldset> предназначен для группировки элементов формы. Такая группировка облегчает работу с формами, содержащими большое число данных.

Тег <legend> применяется для создания заголовка группы элементов формы, которая определяется с помощью тега <fieldset>.

-->

<html>

<head>

<title>Группировка элементов формы</title>

</head>

<body>

<form>

<fieldset style="border: 2px solid red; width: 500px;">

<legend>Authentication</legend>

<br />

Login:

<input type="text" name="login" />

<br />

Password:

<input type="password" name="password" />

<br />

<input type="submit" name="submit" value="Enter" />

</fieldset>

</form>

## Label

Тег label устанавливает связь между определенной метекой и элементом формы.

Существует два способа связывания объекта и метки. Петрвый заключается в использовании идентификатора id внутри элемента формы

и указания его имени в качестве атрибута for тега lable. При втором способе элемент формы помещается внутырь контейнера.

Атрибут accesskey содержит клавишу которую необходимо зажать вместе с клавишей alt для того чтобы

связанный элемент управления получил фокус.

-->

<style type="text/css">

label

{

width: 200px;

float: left;

}

</style>

<html>

<head>

<title>Label</title>

</head>

<body>

<form>

<br />

<label for="first\_name" accesskey="k">Enter your first name: </label>

<input type="text" id="first\_name" name="first\_name" value="Ivan" />

<br />

<label for="sec\_name" accesskey="s">Enter your second name:</label>

<input type="text" id="sec\_name" name="sec\_name" value="Ivanov" />

<br />

<label for="psw" accesskey="p">Enter your password:</label>

<input type="password" id="psw" name="psw" />

<br />

<input type="submit" name="submit" value="Enter" />

<br />

<input type="reset" name="reset" value="Reset" />

</form>

## Title

Атрибуты title отвечает за значение, которое пользователю браузер будет выдавать как подсказку при наведении на элемент,

с текстом, написанном в этих атрибутах.

-->

<html>

<head>

<title>атрибут title</title>

</head>

<body>

<form>

<br />

Enter your first name:

<input type="text" name="first\_name" value="Ivan" title="First Name" />

<br />

Enter your second name:

<input type="text" name="sec\_name" value="Ivanov" title="Second Name" />

<br />

Enter your password:

<input type="password" name="password" title="password" />

<br />

<input type="submit" name="submit" value="Enter" />

<br />

<input type="reset" name="reset" value="Reset" />

</form>

## Disabled

Атрибут disabled - указывает на то, что элемент формы должен отображаться пользователю не активным.

-->

<html>

<head>

<title>атрибут style</title>

</head>

<body>

<form>

<br />

Enter your first name:

<input type="text" name="first\_name" value="Ivanov" />

<br />

Enter your second name:

<input type="text" name="sec\_name" value="Ivan" disabled="disabled" />

<br />

Enter your password:

<input type="password" name="password" />

<br />

<input type="submit" name="submit" value="Enter" />

<br />

<input type="reset" name="reset" value="Reset" />

</form>

## Readonly

Атрибут readonly позволяет сделать элемент не восприимчивым к изменениям пользователя.

-->

<html>

<head>

<title>атрибут style</title>

</head>

<body>

<form>

<br />

Enter your first name:

<input type="text" name="first\_name" value="Ivan" />

<br />

Enter your second name(readonly):

<input type="text" name="sec\_name" value="Ivanov" readonly="readonly" />

<br />

Enter your second name(disabled):

<input type="text" name="sec\_name" value="Ivanovich" disabled="disabled" />

<br />

Enter your password:

<input type="password" name="password" />

<br />

<input type="submit" name="submit" value="Enter" />

<br />

<input type="reset" name="reset" value="Reset" />

</form>

Input size

Атрибут size используется для указания размеров текстового поля для ввода.

-->

<html>

<head>

<title>атрибут style</title>

</head>

<body>

<form>

<br />

Enter your first name:

<input type="text" name="first\_name" value="Ivan" size="20" />

<br />

Enter your second name:

<input type="text" name="sec\_name" value="Ivanov" size="70" />

<br />

Enter your password:

<input type="password" name="password" size="70" />

<br />

<input type="submit" name="submit" value="Enter" />

<br />

<input type="reset" name="reset" value="Reset" />

</form>

## Input maxlength

Атрибут maxlength используется для того, чтобы задать максимально допустимое значение для текстого поля.

maxlength="5" невозможно ввести более пяти символов.

-->

<html>

<head>

<title>атрибут style</title>

</head>

<body>

<form>

<br />

Enter your first name: <input type="text" name="first\_name" value="Ivan" size="30" maxlength="15" />

<br />

Enter your second name: <input type="text" name="sec\_name" value="Ivanov" size="30" maxlength="5" />

<br />

Enter your password: <input type="password" name="password" size="30" maxlength="10" />

<br />

<input type="submit" name="submit" value="Enter" />

<br />

<input type="reset" name="reset" value="Reset" />

</form>