



CSS

Что такое CSS?

CSS расшифровывается **C**ascading **S**tyle **S**heets (Каскадные Документы Стилей).

С помощью CSS Вы можете оформлять HTML документы.

CSS было представлено вместе с HTML 4.0 и создавалось для разрешения проблем с оформлением, возникающих в предыдущих версиях HTML.

Пример

```
#css table { border-collapse:collapse; font-family:'arial'; }  
#css th { background-color:#DC63C2; padding:7px;  
font-size:1.1em; border:1px black solid; color:white; } #css  
td { padding:7px; border:1px solid; }
```

Использование стилей разрешает множество проблем

Во время создания HTML в 1991 году не предполагалось, что он будет содержать тэги, позволяющие оформлять документы.

Изначально HTML был создан только для группировки текста:

- ▶ `<h1>` Заголовок 1 `</h1>`
`<p>` Какой-то текст `</p>`
- ▶ `<h1>` Заголовок 2 `</h1>`
- ▶ `<p>` Какой-то текст `</p>`

Когда в HTML 3.2 были добавлены такие тэги как `` (данный тэг использовался для изменения шрифта элементов) оформление и разметка смешались в единое целое и оформление элементов стало занимать огромное количество времени.

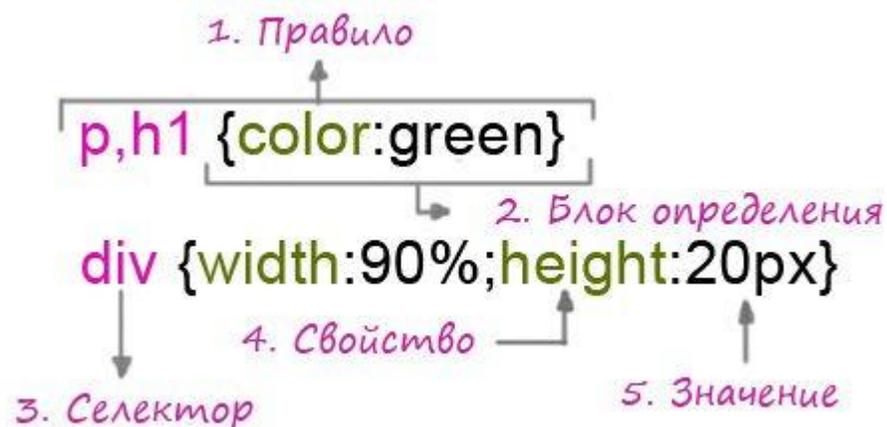
Для того, чтобы разрешить эту проблему в конце 1996 года W3C (Консорциум Всемирной Паутины - занимается разработкой веб-стандартов) представил CSS.

CSS позволяет хранить информацию об оформлении HTML документа в отдельном внешнем файле с расширением `.css`. Редактируя лишь один этот файл стало возможным изменение оформления целого веб-сайта.

На данный момент CSS является стандартом оформления HTML документов и поддерживается всеми современными браузерами.

CSS синтаксис

- ▶ Таблицы стилей состоят из набора правил(1). Каждое правило состоит из одного или нескольких селекторов(3) и блока определения(2), выделяющегося фигурными скобками.
- ▶ Блок определения может содержать одно или несколько свойств(4) отделенных точкой с запятой (;) (после последнего свойства точка с запятой необязательна). Каждое свойство должно иметь значение(5)отделенное двоеточием (:).



CSS комментарии

Таблицы стилей могут содержать комментарии. Комментарии используются для создания пояснений.

Комментарии полностью игнорируются браузером при разборе таблиц стилей. В CSS комментарии начинаются с "/*", и заканчиваются "*/", например:

```
/* Правило перекрасит абзацы HTML документа в зеленый цвет */ p {color:green;}
```

Обратите внимание: разбираться в коде с пояснениями намного проще, так что старайтесь использовать пояснения как можно чаще.

Селекторы CSS

- ▶ С помощью селекторов Вы можете выбирать элементы на странице, которые хотите оформить.
- ▶ В CSS существуют следующие виды селекторов:

Селекторы тэгов

- ▶ Вы можете выбирать элементы на странице для оформления по названию тэга.

```
p { color:green; } h2 { color:red; }
```

Селектор id

- ▶ Данный вид селекторов позволяет производить более точную выборку и используется, когда необходимо выбрать только один определенный элемент на странице, с предварительно заданным идентификатором.
- ▶ Идентификатор для элемента задается с помощью атрибута id (<p id="идентификатор">текст</p>).
- ▶ Для того, чтобы затем оформить данный элемент необходимо обратиться к идентификатору в таблицах стилей добавив перед ним символ "#" (#идентификатор {color:red}).

```
/* Оформим элемент с id="test1" */ #test1 { color:green;  
font-family:verdana; font-size:1.2em; }
```

Селекторы CSS

Селектор class

- ▶ Данный вид селекторов позволяет выбирать для оформления не единственный элемент, а группу элементов.
- ▶ С помощью атрибута class можно задать, что элемент относится к группе (`<p class="имя_группы">текст</p>`).
- ▶ Для того, чтобы затем оформить эту группу необходимо в таблицах стилей обратиться к имени группы добавив перед ней символ "." (`.имя_группы {color:red}`).

```
/* Свойства будут применены ко всем элементам с  
class="test1" */ .test1 { color:green; font-family:verdana;  
font-size:1.2em; }
```

Обратите внимание: имя группы и идентификатор может состоять только из латинских букв и не может начинаться с цифр.

Внешнее объявление стилей

- ▶ Внешнее объявление стилей используется в случаях, когда оформление задается для группы связанных HTML документов (например для целого веб-сайта).
- ▶ В этом случае все оформление выносится в один внешний файл, на который должны ссылаться все документы веб-сайта.
- ▶ Внешнее объявление стилей очень удобно так как позволяет редактируя лишь один файл изменять оформление целого веб-сайта.
- ▶ Для того, чтобы подключить внешний файл стилей необходимо в секции head каждой страницы веб-сайта указать ссылку на него с помощью элемента `<link>`:

Пример:

```
<head> <link rel="stylesheet" type="text/css"
href="адрес_внешнего_файла_стилей" /> </head>
```

- ▶ **Обратите внимание:** атрибут `type` тэга `style` стал необязательным в HTML5.
- ▶ Внешний файл стилей является обычным текстовым файлом с расширением `.css`.
- ▶ Пример содержимого внешнего файла стилей:
- ▶ `h1 {color:red;} p {margin-right:38px;} div {float:left;}`

Внутреннее объявление стилей

- ▶ Внутреннее объявление стилей используются в случаях, когда стиль нужно задать только для одного отдельного HTML документа.
- ▶ В этом случае оформление определяется в секции head с помощью тэга `style`:

Пример:

```
<head> <style type='text/css'> h1 {color:red;} p  
{margin-right:38px;} div {float:left;} </style> </head>
```

Строковое объявление стилей

- ▶ Строковое объявление стилей используется в случаях, когда необходимо оформить только один определенный элемент в HTML документе.
- ▶ Для того, чтобы оформить элементы этим способом Вы должны воспользоваться атрибутом `style` соответствующего элемента.
- ▶ Атрибут `style` может содержать любые CSS свойства.

`<p style="font-size:1.3em">` Абзац оформленный с помощью CSS.`</p>`

Приоритет стилей

- ▶ Стили подключенные разным способом имеют разный приоритет:
 - ▶ Внешние стили: 1
 - ▶ Внутренние стили: 2
 - ▶ Строковые стили: 3
- ▶ Если один HTML документ имеет несколько стилей подключенных разным способом и в них заданы разные свойства оформления для одного и того же элемента, то в итоге элемент будет оформлен согласно содержимому стиля с более высоким приоритетом.

/ Внутренние стили: */ h1 {color:red}*

/ Строковые стили: */ h1 {color:green}*

/ В результате заголовок будет зеленого цвета так как строковые стили имеют более высокий приоритет */*

Цвет текста

- ▶ С помощью CSS свойства `color` Вы можете изменять цвет текста HTML элементов.
- ▶ Цвет может быть задан следующими способами:
- ▶ С помощью названия цвета (например `'red'` задаст красный цвет);
- ▶ С помощью RGB значения (например `'rgb(255,255,255)'` задаст белый цвет);
- ▶ С помощью шестнадцатеричного значения (например `'#00ff00'` задаст зеленый цвет).
- ▶ **Пояснения:**
- ▶ Первый способ в основном используется для задания основных цветов, названия которых хорошо известны. Например `red` определит красный, `blue` - синий, `white` - белый.
- ▶ Второй способ может использоваться для задания любых цветов и оттенков.
- ▶ **Синтаксис:**
`rgb(красный,зеленый,голубой)`
- ▶ `красный` число от 0 до 255 указывающее как много красного будет в итоговом оттенке.
- ▶ `зеленый` число от 0 до 255 указывающее как много зеленого будет в итоговом оттенке.
- ▶ `голубой` число от 0 до 255 указывающее как много голубого будет в итоговом оттенке.
- ▶ Например `rgb(255,0,0)` задаст красный цвет, а `rgb(0,255,0)` зеленый. Смешивая красный с зеленым `rgb(255,255,0)` мы получим желтый.

Цвет текста

- ▶ Третий способ по функциональности эквивалентен второму, но более компактен. На практике в основном используют именно этот способ.
- ▶ Синтаксис:

#красныйзеленыйголубой

- ▶ **красный** шестнадцатеричное число от 0 до ff указывающее как много красного будет в итоговом оттенке.
- ▶ **зеленый** шестнадцатеричное число от 0 до ff указывающее как много зеленого будет в итоговом оттенке.
- ▶ **голубой** шестнадцатеричное число от 0 до ff указывающее как много голубого будет в итоговом оттенке.
- ▶ Например *#ff0000* задаст красный цвет, а *#00ff00* зеленый. Смешивая красный с зеленым *#ffff00* мы получим желтый.

Цвет текста

- ▶ Теперь попробуем перекрасить абзацы в зеленый цвет всеми перечисленными выше способами:

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

```
p {color:rgb(0,255,0);}
```

```
p {color:#00ff00;}
```

- ▶ **Обратите внимание:** удобно выбирать необходимые оттенки цвета можно с помощью [подборщика цветов](#)



| Обозначение | Расшифровка |
|-------------|---|
| R | Красная составляющая цвета. |
| G | Зеленая составляющая цвета. |
| B | Синяя составляющая цвета. |
| # | Выбранный цвет в шестнадцатеричном формате. |
| H | Оттенок цвета. |
| S | Насыщенность цвета. |
| B | Яркость цвета. |

Выравнивание текста

- ▶ С помощью CSS свойства `text-align` Вы можете выровнять текст элемента по горизонтали.
- ▶ Текст может быть выровнен:
 - ▶ По центру (значение `center`);
 - ▶ По левому краю (`left`);
 - ▶ По правому краю (`right`);
 - ▶ По ширине (`justify`).
- ▶ **Обратите внимание:** по ширине (`justify`) текст выравнивается путем растягивания всех строчек до одинаковой длины. Этот метод выравнивания часто используется в газетах и журналах.

```
p.ta1 {text-align:center;}
```

```
p.ta2 {text-align:left;}
```

```
p.ta3 {text-align:right;}
```

```
p.ta4 {text-align:justify;}
```

Свойство `text-decoration`

- ▶ С помощью CSS свойства `text-decoration` Вы можете сделать текст HTML элемента:
 - ▶ Подчеркнутым (значение `underline`)
 - ▶ Перечеркнутым (`line-through`)
 - ▶ Отобразить над текстом элемента линию (`overline`)

```
p.td1 {text-decoration:underline;}  
p.td2 {text-decoration:line-through;}  
p.td3 {text-decoration:overline;}
```

- ▶ Свойство `text-decoration` со значением `none` "очищает" текст от всех вышеперечисленных эффектов. Это может использоваться для создания не подчеркнутых ссылок.

```
a:link {text-decoration:none;}  
a:visited {text-decoration:none;}
```

- ▶ **Обратите внимание:** подчеркивать обычный текст не рекомендуется, так как пользователи могут перепутать его со ссылкой.

Отступ между словами и буквами в тексте

- ▶ С помощью CSS свойства **letter-spacing** Вы можете увеличивать или уменьшать отступ между буквами в тексте HTML элементов.

```
p.ls1 {letter-spacing:10px;}
```

- ▶ С помощью свойства **word-spacing** Вы можете увеличивать или уменьшать отступ между словами в тексте HTML элементов.

```
p.ws1 {word-spacing:15px;}
```

Остальные CSS свойства оформления текста

| Свойство | Описание | Значения |
|-----------------------|--|--|
| direction | Устанавливает направление текста. | ltr rtl |
| line-height | Устанавливает высоту строки. | normal пиксели % |
| text-indent | Устанавливает величину отступа первого символа текста. | пиксели % |
| text-transform | Устанавливает регистр букв текста элемента. | none capitalize uppercase lowercase |
| vertical-align | Устанавливает вертикальное выравнивание элемента. | sub super top middle bottom пиксели % |
| white-space | Устанавливает как должны обрабатываться пробелы внутри элемента. | normal pre nowrap |

Выполнить задание

http://www.wisdomweb.ru/editor/wweditor.php?fname=css_txtex1

Установка шрифта

- ▶ CSS свойство `font-family` позволяет устанавливать шрифт для текста HTML элементов.
- ▶ **Обратите внимание:** если название шрифта состоит из нескольких слов, то оно обязательно должно заключаться в кавычки.

```
p {font-family:Arial;}
```

Безопасные шрифты

- ▶ Безопасными шрифтами называют шрифты, вероятность поддержки которых на любом компьютере с любой установленной ОС близка к 100%.

- ▶ Список безопасных шрифтов:

- ▶ Arial

`<p style='font-family:Arial;'>Демонстрация шрифта Arial.</p>`

- ▶ Arial Black

`<p style='font-family:"Arial Black";'>Демонстрация шрифта Arial Black.</p>`

- ▶ Courier New

`<p style='font-family:"Comic Sans MS";'>Демонстрация шрифта Comic Sans MS.</p>`

- ▶ Comic Sans MS

`<p style='font-family:"Courier New";'>Демонстрация шрифта Courier New.</p>`

- ▶ Georgia

`<p style='font-family:Georgia;'>Демонстрация шрифта Georgia.</p>`

- ▶ Impact

`<p style='font-family:Impact;'> Демонстрация шрифта Impact.</p>`

- ▶ Times New Roman

`<p style='font-family:"Times New Roman";'>Демонстрация шрифта Times New Roman.</p>`

- ▶ Trebuchet MS

`<p style='font-family:"Trebuchet MS";'>Демонстрация шрифта Trebuchet MS.</p>`

- ▶ Verdana

`<p style='font-family:Verdana;'>Демонстрация шрифта Verdana.</p>`

Обратите внимание: в CSS3 было добавлено новое свойство `@font-face`, позволяющее использовать на веб-страницах небезопасные шрифты.

Обратите внимание: если Вы хотите использовать небезопасные шрифты, но не хотите использовать новые свойства используйте резервные шрифты, о которых рассказано далее.

Резервные шрифты

- ▶ При задании шрифта для элемента Вы можете указать **резервный шрифт**, который будет использован в случае, если основной шрифт не установлен на компьютере пользователя.
- ▶ Элементы могут также иметь несколько резервных шрифтов, которые будут поочередно перебираться пока не найдется установленный шрифт.

/ Если на компьютере пользователя нет шрифта Verdana, то будет использован Times New Roman, если на компьютере нет Times New Roman, то будет использован Arial */*

```
p {font-family:Verdana,"Times New Roman",Arial;}
```

- ▶ В качестве замыкающего резервного шрифта часто указывают семейство, к которому принадлежит основной шрифт, чтобы в крайнем случае браузер автоматически выбрал один из шрифтов данного семейства установленных на компьютере пользователя.
- ▶ В CSS шрифты делятся на следующие "семейства":
- ▶ **Serif** шрифты данного семейства имеют небольшие засечки на концах символов. Примеры шрифтов данного семейства: Times New Roman, Georgia.
- ▶ **Sans-Serif** шрифты данного семейства не имеют засечек. Примеры: Arial, Arial Black, Trebuchet MS, Verdana.
- ▶ **Monospace** все символы шрифтов данного семейства занимают одинаковую ширину. Примеры: Courier New.
- ▶ **Fantasy** семейство "причудливых" шрифтов использующихся в основном для создания красочных заголовков. Примеры: Impact
- ▶ **Cursive** семейство шрифтов рукописного начертания. Примеры: Comic Sans MS

/ Если на компьютере пользователя нет шрифта Verdana, то будет использован Arial, если на компьютере нет Arial, то будет использоваться один из шрифтов семейства Sans-Serif имеющихся на компьютере */*

```
p {font-family:Verdana,Arial,sans-serif;}
```

Размер шрифта

- ▶ CSS свойство **font-size** устанавливает размер шрифта HTML элементов. Размер шрифта можно задать двумя способами:
- ▶ **Абсолютный способ**
- ▶ Этот способ по сравнению со следующим удобнее, но его использование может привести к несоответствию при отображении одной и той же страницы в различных браузерах.
- ▶ Данный способ позволяет задавать размер шрифта в абсолютных единицах таких как: пиксели (px) или проценты (%).

```
p.fz1 {font-size:20px;} p.fz2 {font-size:30px;} p.fz3 {font-size:13px;}
```

- ▶ **Относительный способ**
- ▶ Данный способ помогает избежать проблем с несоответствием при отображении страницы в разных браузерах, так как все размеры устанавливаются относительно.
- ▶ Для задания размера шрифта данным способом используются единицы **em**. **1em** эквивалентен размеру шрифта в браузере по умолчанию и равен **16px**.
- ▶ **Обратите внимание:** W3C рекомендует для задания шрифта использовать именно этот способ.

```
p.fz1 {font-size:1.2em;} p.fz2 {font-size:1.5em;} p.fz3 {font-size:1.15em;}
```

Обратите внимание: не используйте свойство **font-size** для того, чтобы оформлять абзацы как заголовки. Всегда используйте для определения абзацев тэги **p**, а для заголовков **h1-h6**

Стиль шрифта

- ▶ Свойство `font-style` позволяет сделать шрифт HTML элемента курсивным.
- ▶ Свойство `font-weight` позволяет изменять толщину шрифта.

```
p.italic {font-style:italic;} p.fz1 {font-weight:bold;}
```

Выполнить задание

- ▶ http://www.wisdomweb.ru/editor/wweditor.php?fname=css_fontex1

Оформление фона в CSS

- ▶ В CSS существует группа свойств для оформления фона HTML элементов:
 - ▶ background-attachment
 - ▶ background-color
 - ▶ background-image
 - ▶ background-position
 - ▶ background-repeat

Цвет фона

- ▶ CSS свойство `background-color` позволяет установить цвет фона для выбранного элемента.
- ▶ Пример ниже делает цвет фона страницы зеленым:

```
body { background-color:green; }
```

- ▶ Цвет может быть задан следующими способами:
- ▶ С помощью имени (например `green` задаст зеленый цвет);
- ▶ С помощью RGB значения (`'rgb(255,0,0)'` задаст красный цвет);
- ▶ С помощью шестнадцатеричного значения (`"#0000ff"` задаст синий цвет).

Изображение в качестве фона

- ▶ С помощью CSS свойства **background-image** Вы можете вставить произвольное изображение в качестве фона.
- ▶ По умолчанию, изображение будет повторяться пока не заполнит все содержимое элемента.

```
body { background-image:url('http://www.wisdomweb.ru/editor/spider2.gif'); }
```

- ▶ **Обратите внимание:** всегда тщательно подбирайте цвета. Фоновое изображение не должно сливаться с текстом.

Изображение в качестве фона

- ▶ Повторяющееся изображение в качестве фона это не всегда, то что нужно.
- ▶ CSS свойства **background-repeat** позволяет определить как должно повторяться фоновое изображение при вставке:
- ▶ **background-repeat:repeat-x** - изображение будет повторяться только по горизонтали;
- ▶ **background-repeat:repeat-y** - изображение будет повторяться только по вертикали;
- ▶ **background-repeat:no-repeat** - изображение не будет повторяться.

/ Фоновое изображение будет повторяться только по горизонтали */*

```
body {  
background-image:url('http://www.wisdomweb.ru/editor/spider2.gif');  
background-repeat:repeat-x;  
}
```

Изображение в качестве фона

- ▶ CSS свойство `background-position` позволяет задавать местоположение фонового изображения.
- ▶ В качестве первого значения данного свойства должна задаваться величина смещения изображения по горизонтали, а в качестве второго величина смещения по вертикали.
- ▶ Величина смещения может быть указана как с помощью пикселей (px), процентов (%) и сантиметров (cm) (`background-position:50px 30px;`), так и с помощью predetermined ключевых слов (`background-position:right top;`).

```
body { background-image:url('http://www.wisdomweb.ru/editor/spider2.gif');  
background-repeat:no-repeat;  
background-position:40px 60px; }
```

Изображение в качестве фона

- ▶ CSS свойство **background-attachment** позволяет прикреплять фоновые изображения к определенным местам.
- ▶ Прикрепленное изображение будет оставаться на данном месте даже при прокрутке страницы.

/ Фоновая картинка будет оставаться на одном месте даже при прокрутке страницы */*

```
body { background-image:url('http://www.wisdomweb.ru/editor/spider2.gif');  
background-repeat:no-repeat;  
background-attachment:fixed; background-position:right top; }
```

Краткая форма записи

- ▶ Для того, чтобы сократить итоговый размер кода в CSS была предусмотрена краткая форма записи некоторых свойств.
- ▶ Для краткой записи оформления фона элементов в CSS предусмотрено свойство **background**.

/ Фоновое изображение будет размещено в правом верхнем углу (в предыдущем примере нам пришлось использовать 4 различных CSS свойства для того, чтобы сделать тоже самое) */*

```
body { background:url('http://www.wisdomweb.ru/editor/spider2.gif') no-repeat fixed right top; }
```

- ▶ При использовании стенографической формы записи соблюдайте следующий порядок следования свойств:
- ▶ background-color
- ▶ background-image
- ▶ background-repeat
- ▶ background-attachment
- ▶ background-position
- ▶ **Обратите внимание:** Вы можете пропускать неиспользуемые значения.

Список всех свойств оформления фон

| Свойство | Описание | Значение |
|------------------------------|--|--|
| background | Устанавливает все возможные свойства фона за одно определение | background-color background-image background-repeat background-attachment background-position |
| background-attachment | Указывает будет ли фоновая картинка привязана к одному месту, или будет прокручиваться вместе с текстом. | scroll fixed |
| background-color | Устанавливает фоновый цвет элемента. | rgb(x,x,x) #xxx название_цвета |
| background-image | Устанавливает фоновую картинку для элемента | url(URL) none |
| background-repeat | Указывает как фоновая картинка будет повторяться | repeat repeat-x repeat-y no-repeat |
| background-position | Указывает координаты расположения фоновой картинки. | left top left center left bottom center top center center center bottom right top right center right bottom x% y% xpos ypos |

Оформление ссылок

- ▶ Ссылки могут находиться в четырех различных состояниях и каждое из этих состояний может быть отдельно оформлено с помощью специальных псевдо-классов.
- ▶ Псевдо-класс должен добавляться к селектору элемента, отделяясь от него двоеточием (:).
- ▶ Оформление для псевдо-классов должно задаваться в порядке перечисленном ниже.
- ▶ **a:link** - определяет оформление обычной не посещенной ссылки.
- ▶ **a:visited** - определяет оформление посещенной пользователем ссылки.
- ▶ **a:hover** - определяет оформление ссылки, на которую наведен курсор мыши.
- ▶ **a:active** - определяет оформление ссылки, на которую щелкнули мышкой.

```
a:link { text-decoration:none; color:green; }  
a:visited { text-decoration:none; color:green; }  
a:hover { text-decoration:underline; color:red; font-size:1.1em; }  
a:active { text-decoration:none; color:red; font-size:1.1em; }
```

Оформление списков

- ▶ CSS свойство **list-style-type** позволяет оформлять списки.
- ▶ Всего существует два вида списков:
- ▶ **Неупорядоченные** - маркер таких списков имеет вид закрашенного круга.
- ▶ **Упорядоченные** - элементы таких списков нумеруются арабскими цифрами.
- ▶ Ниже представлены таблицы с возможными значениями для оформления списков:
- ▶ **Значения для неупорядоченных списков:**

| Значение | Описание |
|---------------|--|
| none | Нет маркера. |
| disc | <i>Значение по умолчанию.</i> Маркер имеет вид закрашенного черного круга. |
| circle | Маркер принимает вид не закрашенного круга. |
| square | Маркер принимает вид закрашенного квадрата. |

Оформление списков

Значения для упорядоченных списков:

| Значение | Описание |
|-----------------------------|--|
| armenian | Элементы списка нумеруются Армянскими числами. |
| decimal | <i>Значение по умолчанию.</i> Элементы списка нумеруются Арабскими числами. |
| decimal-leading-zero | Элементы списка нумеруются Арабскими числами начинающимися с нуля (01, 02, 03 и т.д.). |
| georgian | Элементы нумеруются традиционной Грузинской нумерацией. |
| lower-greek | Элементы нумеруются традиционной Греческой нумерацией (alpha, beta, gamma). |
| lower-latin | Элементы нумеруются маленькими буквами латиницы (a, b, c, d). |
| lower-roman | Элементы нумеруются маленькими римскими цифрами (i, ii, iii, iv). |
| upper-latin | Элементы нумеруются большими буквами латиницы (A, B, C, D). |
| upper-roman | Элементы нумеруются большими римскими цифрами (I, II, III, IV). |

Использование маркера-картинки

- ▶ Свойство `list-style-image` позволяет заменить маркер списка на произвольное изображение.

```
ul.lis1 { list-style-image:url('marimg.gif'); }
```

Оформление таблиц

- ▶ Пример оформления таблицы

```
table, th, td
{
border-style:solid; border-width:1px; border-collapse:collapse; padding:2px;
}
th
{
height:28px; background-color:#f892dc; color:white; border-color:black;
}
.ts1
{
background-color:#ffeffb;
}
```

Ширина и Высота таблицы

- ▶ С помощью CSS свойства **width** Вы можете устанавливать ширину таблицы.
- ▶ В основном ширина устанавливается в пикселях или %, но можно также использовать **cm** и **em**.

```
.tab1 { width:100%; }  
.tab2 { width:70%; }  
.tab3 { width:300px; }
```

- ▶ CSS свойство **height** позволяет установить высоту таблицы. Высота в основном указывается в пикселях, но можно также использовать **cm** и **em**.

```
.tab1 { height:200px; }  
.tab2 { height:7cm; }
```

Оформление границ

- ▶ Для оформления табличных границ в CSS используется свойство **border**.

```
table, th, td {  
  border-style:solid; border-color:green; border-width:1px;  
}
```

- ▶ **Обратите внимание:** свойство **border** не является уникальным свойством таблиц оно может использоваться с любыми элементами. Данное свойство будет подробно рассмотрено далее в данном учебнике.

Свойство `border-collapse`

- ▶ Таблица в примере выше имеет двойную границу, потому что и сама таблица и ее ячейки (элементы `th` и `td`) имеют собственные границы.
- ▶ Свойство `border-collapse` позволяет соединить границы таблицы и ячеек. Соединенные границы обычно смотрятся более аккуратно и красиво.

```
table, th, td
{
border-style:solid; border-color:green; border-width:1px; border-collapse:collapse;
}
```

Выравнивание табличного текста

- ▶ С помощью свойства **text-align** Вы можете выравнивать текст табличных ячеек по горизонтали. Текст может быть выровнен:

- ▶ По левому краю (значение left)
- ▶ По правому краю (right)
- ▶ По центру (center)

```
.tab1 { text-align:right; }  
.tab2 { text-align:left; }  
.tab3 { text-align:center; }
```

- ▶ С помощью свойства **vertical-align** Вы можете выравнивать текст табличных ячеек по вертикали. Текст может быть выровнен:

- ▶ По верхней границе (top)
- ▶ По центру (middle)
- ▶ По нижней границе (bottom)

```
.top { vertical-align:top; }  
.mid { vertical-align:middle; }  
.bot { vertical-align:bottom; }
```

Свойство padding

- ▶ С помощью свойства padding Вы можете устанавливать величину отступа между границей ячейки и ее содержимым.

```
#tab1 td { padding:10px; }  
#tab2 td { padding:5px; }  
#tab3 td { padding:0px; }
```

Сделайте сами

- ▶ http://www.wisdomweb.ru/editor/wweditor.php?fname=css_tableex1

Стиль границ

- ▶ CSS свойство **border-style** позволяет установить стиль для границ HTML элемента.
- ▶ **Значения для установки стилей границ:**
 - ▶ **solid** границы будут нарисованы сплошной линией;
 - ▶ **dashed** границы будут нарисованы пунктирной линией;
 - ▶ **dotted** границы будут нарисованы точками;
 - ▶ **double** границы будут нарисованы двойной сплошной линией.

```
.bor1 { border-style:solid; }  
.bor2 { border-style:dashed; }  
.bor3 { border-style:dotted; }
```

- ▶ **Обратите внимание:** по умолчанию граница элементов всегда невидима (значение none).

Цвет границы

- ▶ С помощью CSS свойства **border-color** Вы можете задать цвет границы HTML элемента.
- ▶ Цвет может быть задан следующими способами:
 - ▶ С помощью имени (например 'red' задаст красный цвет);
 - ▶ С помощью RGB значения (например 'rgb(255,255,255)' задаст белый цвет);
 - ▶ С помощью шестнадцатеричного значения (например '#00ff00' задаст зеленый цвет).

```
.bor1 { border-style:solid; border-color:red; }  
.bor2 { border-style:solid; border-color:green; }  
.bor3 { border-style:solid; border-color:blue; }
```

Толщина границы

- ▶ С помощью CSS свойства **border-width** Вы можете задать толщину границы HTML элемента.
- ▶ Толщина может быть указана либо в пикселях, либо с помощью predefined значений: **thin**, **medium**, **thick** (тонкая, средняя, толстая).

```
.bor1 { border-style:solid; border-width:4px; }  
.bor2 { border-style:solid; border-width:2px; }  
.bor3 { border-style:solid; border-width:thin; }
```

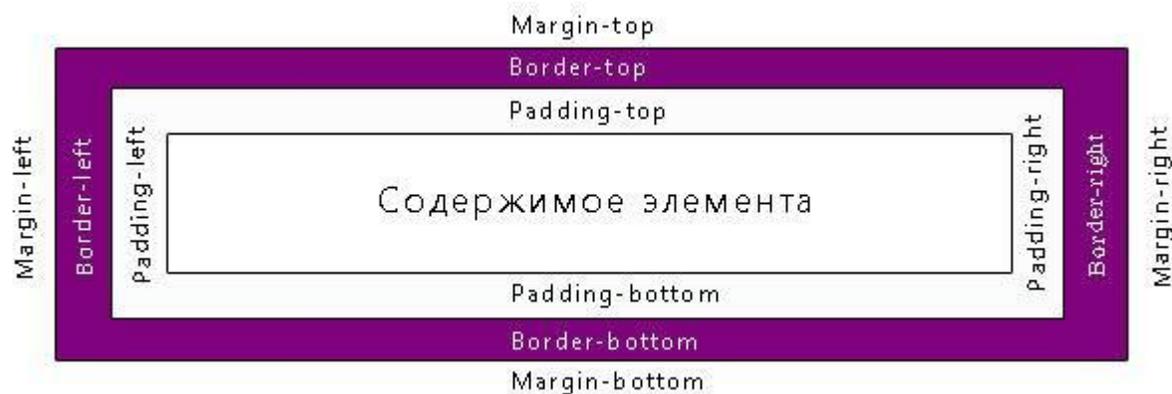
Отступы

- ▶ **CSS Padding**
- ▶ В CSS существует два вида отступов: **внутренние** и **внешние**.
- ▶ Внутренний отступ - это расстояние между содержимым элемента и его границей. Внешний отступ - это расстояние отступаемое с внешней стороны границы элемента.
- ▶ Величина отступов может определяться в пикселях (px), сантиметрах (cm), процентах (%), em.
- ▶ CSS свойство **padding** позволяет установить величину внутреннего отступа.

```
.pad1 { border-style:solid; padding:20px; }  
.pad2 { border-style:solid; padding:10px; }  
.pad3 { border-style:solid; padding:5px; }
```

Отступы

- ▶ Величина отступа может быть независимо определена для каждой стороны элемента:



/ Отступ от содержимого элемента до его верхней границы равен 30, до левой 20, до нижней 10 и до правой 40 пикс. */*

```
.pad1 { padding-top:30px; padding-left:20px; padding-bottom:10px; padding-right:40px; }
```

Скрытие элементов в CSS

- ▶ В CSS скрыть элементы можно двумя способами:
 - ▶ С помощью свойства `visibility:hidden;`
 - ▶ С помощью `display:none.`
- ▶ Элемент скрытый первым способом будет невидим, но будет по прежнему занимать место на странице.

/ Абзац стал невидим, но все еще занимает место на странице */*

```
p.dis1 { visibility:hidden; }
```

- ▶ Второй способ позволяет полностью скрыть элемент, чтобы он больше не занимал места на странице.

/ Абзац стал невидим и не занимает места */*

```
p.dis1 { display:none; }
```

Размещение элементов

- ▶ С помощью свойств позиционирования Вы можете размещать элементы где пожелаете.

```
#pos1 { position:absolute; top:10px; left:200px; }  
#pos2 { position:absolute; top:1px; left:0px; }  
#pos3 { position:absolute; top:100px; right:70px; }
```

- ▶ Местоположение элементов задается с помощью следующих CSS свойств:
- ▶ **top** - устанавливает величину смещения текущего элемента от верхнего края родительского элемента;
- ▶ **bottom** - устанавливает величину смещения текущего элемента от нижнего края родительского элемента;
- ▶ **left** - устанавливает величину смещения текущего элемента от левого края родительского элемента;
- ▶ **right** - устанавливает величину смещения текущего элемента от правого края родительского элемента.
- ▶ Описанные выше свойства не вступят в силу пока Вы не зададите **способ размещения**. Также способ размещения определяет поведение данных свойств.
- ▶ В CSS существуют 4 различных способа размещения элементов:

Статическое размещение

- ▶ Статичные элементы всегда отображаются там, где они были объявлены. CSS свойства `top`, `bottom`, `left` и `right` не работают со статичными элементами.
- ▶ Все элементы по умолчанию размещаются данным способом, но Вы также можете явно объявить элемент статичным с помощью `position:static`.

```
#pos1 { position:static; top:40px; left:17px; }
```

Фиксированное размещение

- ▶ Фиксировано размещенные элементы не изменяют своего местоположения даже при прокрутке окна браузера. К фиксировано размещенным элементам могут применяться CSS свойства `top`, `bottom`, `left`, `right`.
- ▶ Элемент может быть объявлен фиксировано размещенным с помощью `position:fixed`.

```
#pos1 { position:fixed; right:40px; top:17px; }
```

Относительное размещение

- ▶ Относительно размещенные элементы размещаются относительно их обычной позиции.
- ▶ Элемент может быть объявлен относительно размещенным с помощью `position:relative`.
- ▶ **Обратите внимание:** элемент будет по-прежнему занимать свою изначальную позицию.

```
#pos1 { position:relative; top:40px; left:170px; }
```

Абсолютное размещение

- ▶ Абсолютно размещенные элементы располагаются относительно первого родительского элемента, способ размещения которого отличен от статического. Если такие элементы не были найдены, то элемент будет расположен относительно корневого элемента (html).
- ▶ Иногда для того, чтобы добиться желаемого эффекта родительский элемент специально определяется как относительно размещенный с нулевым смещением.
- ▶ Вы можете объявить элемент абсолютно размещенным с помощью **position: absolute**.

```
#pos1 { position: absolute; top: 10px; left: 200px; }  
#pos2 { position: absolute; top: 1px; left: 0px; }  
#pos3 { position: absolute; top: 100px; right: 70px; }
```

Наложение элементов

- ▶ При применении свойств позиционирования элементы могут накладываться друг на друга. Свойство `z-index` позволяет установить какой элемент в случае наложения будет сверху, а какой снизу.
- ▶ Элементы с большим значением свойства `z-index` располагаются выше других.
- ▶ **Обратите внимание:** свойство `z-index` может принимать отрицательные значения.

```
#pos1 { z-index:10; }  
#pos2 { z-index:5; }  
#pos3 { z-index:-1; }
```

Связанные с размещением CSS свойства

| Свойства | Описание | Значения |
|-----------------|---|---|
| clip | Обрезает элемент размещенный абсолютно. | rect auto inherit |
| cursor | Задаёт вид, который будет принимать курсор при наведении на элемент. | auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help |
| overflow | Устанавливает как должно быть отображено содержимое вышедшее за границы элемента. | auto hidden scroll visible inherit |

Сделайте сами

- ▶ http://www.wisdomweb.ru/editor/wweditor.php?fname=css_posex1

Выравнивание

- ▶ **Центрирование с помощью margin**
- ▶ Блочные элементы могут быть выровнены по центру установкой margin с левой и правой стороны значения `auto`.
- ▶ **Обратите внимание:** если ширина блочного элемента будет равна 100%, то он не будет выровнен (т.к. доступного margin нет).

```
.ali1 { margin-left:auto; margin-right:auto; width:50%; }
```

Выравнивание

- ▶ **Выравнивание с помощью свойств позиционирования**
- ▶ Вы можете выравнивать элементы с помощью свойств позиционирования.

```
.pos { position:absolute; width:400px; left:0px; background-color:green; }
```

Выравнивание

▶ Свойство float

- ▶ Элементы в CSS также могут быть выравнены с помощью свойства `float`.
- ▶ Элемент выравненный с помощью `float` будет прижат к левой или правой границе родительского элемента (в зависимости от заданного значения) и заставит следующие за ним элементы "обтекать" его с противоположной стороны.
- ▶ Свойство `float` часто используют с картинками (как в примере ниже), но оно также бывает полезно при обычном выравнивании.

```
.fl1 { float:right; }
```

▶ Очищение элементов от float

- ▶ HTML элементы следующие за элементом с заданным `float` будут его "обтекать", чтобы избежать этого используйте свойство `clear`.
- ▶ Значения переданные данному свойству (`left` или `right`) указывают с какой стороны находится элемент с `float`. Если Вы не уверены или элементы с `float` находятся с двух сторон воспользуйтесь свойством `both`.

```
.fl2 { clear:both; }
```

Сделайте сами

- ▶ http://www.wisdomweb.ru/editor/wweditor.php?fname=css_alix1