# Технология CSS

**Каскадные таблицы стилей** или **CSS** (Cascading Style Sheets) являются следствием развития HTML и дают возможность перейти на следующий уровень представления информации. Таблицы стилей позволяют разделить смысловое содержимое странички и его оформление.

Все оформление рекомендуется вынести во внешний стилевой файл. Основная же страничка будет содержать только информацию и ссылки на необходимые стили.

При показе странички конкретному устройству должна быть задействована соответствующая случаю таблица стилей. Для сотового телефона и монитора компьютера они, разумеется, должны быть разными. В первом случае мы используем минимальное оформление, которое позволит представить информацию наиболее оптимально и компактно. Во втором же случае в нашем распоряжении имеется все богатство шрифтового и цветового оформления.

Таблицу стилей нужно написать всего один раз при создании сайта для каждого из устройств, на котором планируется вывод информации. К тому же таблица стилей может быть единой для целого сайта. И, следовательно, не нужно будет повторять одни и те же описания стилей на каждой из страниц.

# Подключение таблиц стилей

Для осуществления этой задачи мы можем воспользоваться одним из 3-х предлагаемых методов/

## Inline-описание (описание, встроенное в тег)

При помощи дополнительного атрибута style мы можем определить нужные нам стилевые параметры в любом теге. Это самый легкий способ, и действует он в пределах лишь одного тега. Этот способ не слишком отличается, к примеру, от прямого описания внешнего вида при помощи тега <font>.

<p style="color:red; text-align:center;">

Этот текст переопределен стилем

</p>

## Описание в секции заголовка

Его действие распространяется на всю страничку. Определение стилей происходит при помощи классов, которые представляют собой списки с определением всех необходимых параметров оформления.

При использовании этого метода описание стилей необходимо разместить в секции заголовка:

<head> ....

<style type="text/css">

<!--

.header {

text-align :center;

font-size : 27pt;}

.red {color : red; }

-->

</style>

</head>

Теперь эти стили можно применять в любом месте html-кода. Для этого используется следующая конструкция:

<p class=header>Этот текст написан стилем header<p>

<p class=red>Этот текст написан красным цветом<p>

Кроме определения новых классов имеется возможность переопределять стандартные теги. Например, тег <p>:

<style type="text/css">

<!-- p { text-align : center; font-size :12pt;}

--> </style>

Теперь весь текст, заключенный в теги <p></p>, будет выглядеть так, как определено данным стилем. Это очень удобно и позволяет легко адаптировать уже существующие странички к использованию стилей. Кроме того, это несколько уменьшает объем файла за счет отсутствия лишних атрибутов class.

## Вынесение описания стилей во внешний файл

Диапазон его воздействия простирается на все файлы, в которые включено описание.

Сначала создается CSS-файл с описанием всех нужных нам классов (напрмер, mystyle.css):

.header { text-align : center; font-size : 27pt;}

.red { color :red; }

p { text-align : center; font-size : 12pt;}

А потом ссылка на него внедряется в документ при помощи тега <link>:

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

href="css/mystyle.css" title="MyStyleSheet"> .... </head>

Это самый удобный способ, и для основной таблицы стилей рекомендуется пользоваться именно им.

# Синтаксис CSS

Описание каждого класса делается при помощи конструкции, подобной этой:

.small { font-size: 9pt; }

Сначала указывается имя класса – оно может быть произвольным, но желательно все-таки давать осмысленное название. Далее, в фигурных скобках перечисляются все необходимые параметры для данного класса. Параметры отделяются друг от друга точкой с запятой. Вот еще один пример, в котором используется более длинное описание:

Заметьте, что имя класса начинается с точки и таким образом опредяет универсальный класс, т.е. такой, который может быть применен к любому тегу. И делается это при помощи следующей конструкции:

<p class=small>Накладываем стиль на этот текст</p>

Существуют универсальные классы и, так называемые, теговые классы:

p.small { font-size: 9pt; }

Класс, определенный таким образом, сработает только в том теге, для которого он предназначен, а для всех остальных будет проигнорирован.

Мы можем определять параметры не только для одного тега, но и сразу для нескольких. Для этого в определении стиля достаточно перечислить их через запятую:

p, td { font-size: 9pt; color:green;}

Такой прием называется группировкой, и в данном случае мы определили и для <p>, и для <td> одинаковый размер и цвет текста.

В случае переопределения существующих тегов, в описании стиля можно указывать не все параметры, а лишь те из них, которые мы хотим изменить. Все остальные параметры примут значения по умолчанию, которые для разных тегов различны.

## Псевдоклассы

В CSS есть такое понятие как псевдокласс. В отличие от обычного класса, действие псевдокласса распространяется не на весь текст, к которому применен данный стиль, а лишь на его часть и возможно лишь в определенном состоянии. Чтобы было понятнее, давайте разберем эффект, при котором ссылки подчеркиваются лишь при наведении на них курсора. Эффект достаточно распространенный, и его можно наблюдать в том числе и на этом сайте. Вот фрагмент таблицы стилей, который позволяет достигать вышеописанного эффекта:

a { text-decoration: none; }

a:hover { text-decoration: underline;}

Верхняя строчка - это переопределение стандартного тега <a>, которое запрещает подчеркивать ссылки, а вот нижняя – это определение стиля для псевдокласса hover, который описывает стиль ссылки в момент, когда курсор находится над ней.

А вот и другой пример псевдокласса – определение буквицы в начале абзаца:

p:first-letter { font-size: 200\%; font-weight: bold; }

Заметьте, что и в том, и в другом случае действие стиля распространяется либо на определенное состояние (пользователь собирается щелкнуть по ссылке), либо на фрагмент текста (изменяется только первая буква абзаца). В этом и заключается смысл псевдоклассов.

## Единицы измерения

В свойствах, которым требуется указание размеров, можно использовать несколько способов для их задания:

* относительный размер в процентах (%)
* относительный размер при помощи словесного описания (larger, smaller, xx-small, x-small, small, medium, large, x-large, xx-large)
* абсолютный размер в типографских единицах - размер может задаваться в пунктах (pt), пиках (pc), пикселях (px), средней шириной буквы "m" (em), средней шириной буквы "x" (eх)
* абсолютный размер в стандартных единицах длины - размер может задаваться в сантиметрах (cm), миллиметрах (mm), дюймах (in), абсолютных пикселях (px)

## Задание цвета

Цвет для тех свойств, где это нужно, может быть определен одним из трех способов:

* при помощи названия цвета: yellow, red, green, grey;
* шестнадцатеричным заданием цвета в формате #RRGGBB: #ff0000, #883490, #ffffff;
* десятичным заданием составляющих цвета в формате rgb: rgb(255,0,0), rgb(100,23,78).

## Комментарии

Как и в любом достаточно сложном языке, при создании таблицы стилей можно пользоваться комментариями. Их формат аналогичен классическому C:

/\* Этот текст является комментарием \*/

Для небольших сайтов эта возможность Вам вряд ли пригодится, а вот при создании сложных, многоуровневых таблиц стилей комментарии могут пригодиться. Кстати, здесь будет уместно привести золотое правило - чем понятнее названа переменная (в данном случае имя класса), тем меньше комментариев необходимо.

## Каскадность стилей

Каскадность заключается в том, что стили могут переопределяться. Приведенный выше список способов внедрения стилей соответствует порядку переопределения. Нижерасположенный способ может переопределять вышерасположенный.

Например, мы определили во внешнем стилевом файле, что текст в теге <p> должен быть написан при помощи шрифта высотой 10 пунктов. Но если в заголовке странички мы дополнительно укажем, что тот же текст в теге <p> должен быть написан шрифтом в 12 пунктов, то текст будет выведен именно таким кеглем - т.е. стиль в заголовке странички переопределил стиль во внешнем файле.

# Основные свойства CSS

## Шрифт

font-family

font-style

font-variant

font-weight

font-size

font

### Семейство шрифта [font-family]

Свойство font-family указывает приоритетный список шрифтов, используемых для отображения данного элемента или web-страницы. Если первый шрифт списка не установлен на компьютере, с которого выполняется доступ к сайту, ищется следующий шрифт списка, пока не будет найден подходящий.

Для категоризации шрифтов используются два типа имён:

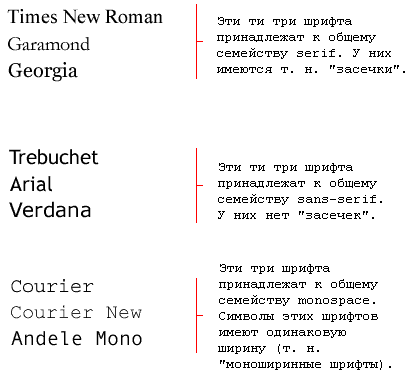
**Family-name**

Пример family-name (часто называемое просто "шрифт") это, например, "Arial", "Times New Roman" или "Tahoma".

**Generic family**

Его можно проще описать как группу family-names, имеющих характерные общие черты. Пример – sans-serif, набор шрифтов без "засечек ".

Разницу можно также проиллюстрировать так:



При указании шрифтов для вашего web-сайта вы, естественно, начинаете с предпочтительного шрифта, а затем перечисляете альтернативные. Рекомендуем в конце списка указывать родовое имя. Тогда страница, как минимум, будет отображена шрифтом того же семейства, если отсутствуют все специфицированные конкретные шрифты.

Список шрифтов может выглядеть так:

h1 {font-family: arial, verdana, sans-serif;}

h2 {font-family: "Times New Roman", serif;}

Заголовки <h1> будут отображаться шрифтом "Arial". Если он не установлен на пользовательской машине, будет использоваться "Verdana". Если недоступны оба шрифта, для показа заголовков будет использован шрифт семейства **sans-serif**.

Обратите внимание, что имя шрифта "Times New Roman" содержит пробелы, поэтому указано в двойных кавычках.

### Стиль шрифта [font-style]

Свойство font-style определяет **normal**, **italic** или **oblique**. В примере все заголовки <h2> будут показаны курсивом *italic*.

h1 {font-family: arial, verdana, sans-serif;}

h2 {font-family: "Times New Roman", serif; **font-style: italic;**}

### Вес шрифта [font-weight]

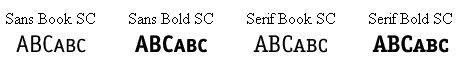
Свойство font-weight описывает, насколько толстым, или "тяжёлым", должен отображаться шрифт. Шрифт может быть **normal** или **bold**. Некоторые браузеры поддерживают даже числовые значения 100-900 (в сотнях) для описания веса шрифта.

p {font-family: arial, verdana, sans-serif;}

td {font-family: arial, verdana, sans-serif; **font-weight: bold;**}

### Вариант шрифта [font-variant]

Свойство font-variant используется для выбора между вариантами **normal** и **small-caps**. Шрифт **small-caps** использует малые заглавные буквы (upper case) вместо букв нижнего регистра. Примеры:



Если font-variant имеет значение **small-caps**, а шрифт small-caps недоступен, браузер, скорее всего, отобразит текст буквами верхнего регистра.

h1 {font-variant: small-caps;}

h2 {font-variant: normal;}

### Размер шрифта [font-size]

Используются различные единицы измерения (например, пикселы и проценты) для описания размера шрифта. В данном учебнике мы будем использовать самые распространённые и удобные единицы измерения. Вот примеры:

h1 {font-size: **30px**;}

h2 {font-size: **12pt**;}

h3 {font-size: **120%**;}

p {font-size: **1em;**}

### Сокращённая запись [font]

Используя сокращенную запись font, можно указывать все свойства шрифта в одном стилевом правиле.

Например, вот четыре строки описания свойств шрифта для <p>:

p {

font-style: italic;

font-weight: bold;

font-size: 30px;

font-family: arial, sans-serif;

}

Используя сокращённую запись, код можно упростить:

p {

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

}

Порядок свойств font:

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

## Абзац

text-indent

text-align

text-decoration

letter-spacing

text-transform

### Отступы [text-indent]

Свойство text-indent позволяет выделить параграф с помощью установки отступа для его первой строки. В примере **30px** применяется ко всем параграфам <p>:

p {

text-indent: 30px;

}

### Выравнивание текста [text-align]

CSS-свойство text-align соответствует атрибуту, используемому в старых версиях HTML. Текст может быть выровнен **left**, **right**, **centred** или **justify**.

В примере текст заголовочных ячеек таблицы <th> выравнивается вправо, а в ячейках данных <td> - по центру. Кроме того, нормальные параграфы - justify:

th {

text-align: **right;**

}

td {

text-align: **center;**

}

p {

text-align: **justify;**

}

### Декоративные эффекты [text-decoration]

Свойство text-decoration позволяет добавлять различные "декоративные эффекты". Например, можно подчеркнуть текст, провести линию по или над текстом и т.д. В примере <h1> подчёркнуты, <h2> - имеют черту над текстом, а <h3> - перечёркнуты.

h1 {

text-decoration: **underline**;

}

h2 {

text-decoration: **overline**;

}

h3 {

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

}

### Интервал между буквами [letter-spacing]

Значение - нужная величина. Например:

h1 {

letter-spacing: **6px;**

}

p {

letter-spacing: **3px;**

}

### Регистр символв [text-transform]

Можно выбрать **capitalize**, **uppercase** или **lowercase**, в зависимости от того, как выглядит текст в оригинальном HTML-коде.

Например, слово "headline" можно показать "HEADLINE" или "Headline". Имеются четыре возможных значения text-transform:

**capitalize**

Капитализирует каждое слово. Например: "john doe" станет "John Doe".

**uppercase**

Конвертирует все символы в верхний регистр. Например: "john doe" станет "JOHN DOE".

**lowercase**

Конвертирует все символы в нижний регистр. Например: "JOHN DOE" станет "john doe".

**none**

Трансформации нет - текст отображается так же, как в HTML-коде.

Для примера мы используем список имён. Все имена выделены с помощью <li> (list-item). Давайте капитализируем все имена и отобразим все заголовки верхним регистром.

h1 {

text-transform: **uppercase;**

}

li {

text-transform: **capitalize;**

}

## Цвет и фон

color

background-color

background-image

background-repeat

background-attachment

background-position

background

### Цвет переднего плана [color]

Свойство color описывает цвет переднего плана элемента.

Например, представьте, что мы хотим сделать все заголовки документа тёмно-красными. Все заголовки обозначаются HTML-элементом <h1>. В нижеприведённом коде цвет элемента <h1> устанавливается красным.

h1 {

**color: #ff0000;**

}

Цвета можно указывать как шестнадцатеричные значения, как в примере (#ff0000), либо вы можете использовать названия цветов ("red") или rgb-значения (rgb(255,0,0)).

### Цвет фона [background-color]

В элементе <body> размещается всё содержимое HTML-документа. Таким образом, для изменения цвета фона всей страницы свойство background-color нужно применить к элементу <body>.

Вы можете также применять это свойство к другим элементам, в том числе - к заголовкам и тексту. В следующем примере различные цвета фона применяются к элементам <body> и <h1>.

body {

**background-color: #FFCC66;**

}

h1 {

color: #990000;

**background-color: #FC9804;**

}

### Фоновые изображения [background-image]

Для вставки рисунка в качестве фонового изображения web-страницы просто примените свойство background-image в тэге <body> и укажите местоположение рисунка.

body {

background-color: #FFCC66;

**background-image: url("butterfly.gif");**

}

h1 {

color: #990000;

background-color: #FC9804;

}

NB: Обратите внимание, что мы специфицируем место, где находится файл как **url("butterfly.gif")**. Это означает, что он находится в той же папке, что и таблица стилей. Разумеется, можно ссылаться и на файлы изображений в других папках, используя,например, **url("../images/butterfly.gif"),** или на файлы в Internet, указывая полный адрес файла : **url("http://www.html.net/butterfly.gif")**.

### Повторение (мультипликация) фонового изображения [background-repeat]

Управляет заполнением фонового рисунка на пространстве страницы. Возможные значения:

|  |  |
| --- | --- |
| **Значение** | **Описание** |
| background-repeat: repeat-x | Рисунок повторяется по горизонтали |
| background-repeat: repeat-y | Рисунок повторяется по вертикали |
| background-repeat: repeat | Рисунок повторяется по горизонтали и вертикали |
| background-repeat: no-repeat | Рисунок не повторяется |

Например, для отмены повторения/мультипликации фонового рисунка мы должны записать такой код:

body {

background-color: #FFCC66;

background-image: url("butterfly.gif");

**background-repeat: no-repeat;**

}

h1 {

color: #990000;

background-color: #FC9804;

}

### Блокировка фонового изображения [background-attachment]

Свойство background-attachment определяет, фиксируется ли фоновый рисунок, или прокручивается вместе с содержимым страницы.

|  |  |
| --- | --- |
| **Значение** | **Описание** |
| Background-attachment: scroll | Изображение прокручивается вместе со страницей – разблокировано. |
| Background-attachment: fixed | Изображение блокировано. При прокручивании страницы изображение не будет двигаться. |

Например, следующий код фиксирует изображение.

body {

background-color: #FFCC66;

background-image: url("butterfly.gif");

background-repeat: no-repeat;

**background-attachment: fixed;**

}

h1 {

color: #990000;

background-color: #FC9804;

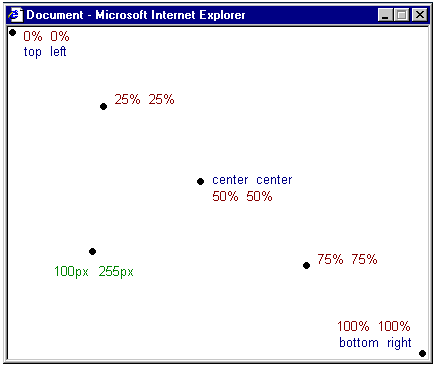
}

### Расположение фонового рисунка [background-position]

По умолчанию фоновый рисунок позиционируется в левом верхнем углу экрана. Свойство background-position позволяет изменять это значение по умолчанию, и фоновый рисунок может располагаться в любом месте экрана.

Есть много способов установить значение background-position. Тем не менее, все они представляют собой набор координат. Например, значение '100px 200px' располагает фоновый рисунок на 100px слева и на 200px сверху в окне браузера.

Координаты можно указывать в процентах ширины экрана, в фиксированных единицах (пикселы, сантиметры, и т. п.), либо вы можете использовать слова top, bottom, center, left и right.



В таблице дано несколько примеров.

|  |  |
| --- | --- |
| **Значение** | **Описание** |
| background-position: 2cm 2cm | Рисунок расположен на 2 cm слева и на 2 cm сверху |
| background-position: 50% 25% | Рисунок расположен по центру и на четверть экрана сверху |
| background-position: top right | Рисунок расположен в правом верхнем углу страницы |

В примере кода фоновое изображение располагается в правом нижнем углу экрана:

body {

background-color: #FFCC66;

background-image: url("butterfly.gif");

background-repeat: no-repeat;

background-attachment: fixed;

**background-position: right bottom;**

}

h1 {

color: #990000;

background-color: #FC9804;

}

### Сокращённая запись [background]

Свойство background входит в состав всех свойств, перечисленных в этом уроке.

С помощью background вы можете сжимать несколько свойств и записывать ваши стили в сокращённом виде, что облегчает чтение таблиц.

Например:

background-color: #FFCC66;

background-image: url("butterfly.gif");

background-repeat: no-repeat;

background-attachment: fixed;

background-position: right bottom;

Используя background, того же результата можно достичь одной строкой кода:

background: #FFCC66 url("butterfly.gif") no-repeat fixed right bottom;

Порядок свойств этого элемента таков:

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

Если свойство отсутствует, оно автоматически получает значение по умолчанию. Например, если background-attachment и background-position нет в данном примере:

background: #FFCC66 url("butterfly.gif") no-repeat;

Этим двум неспецифицированным свойствам будут присвоены значения по умолчанию - scroll и top left.

## Блочная структура

В концепции CSS любую часть документа можно представить в виде **блока** (box):

**Содержимое блока (box content)**

Поля (margin)

Отступы (padding)

Рамка (border)

top

bottom

right

left

Блоки (**элементы**) могут вкладываться один в другой. Например, блок в тэгах <body> </body> содержит внутри себя все отсальные блоки.

### Поля [margin]

У элемента есть четыре стороны: right, left, top и bottom. Поля margin это расстояние от каждой стороны с до соседних элементов (или краёв документа).

В качестве первого примера мы разберёмся, как определить поля самого документа, т. е. элемента <body>. На иллюстрации показано, какие поля нам нужны.

|  |  |
| --- | --- |
| Examples of margins | CSS-код:  body {  margin-top: 100px;  margin-right: 40px;  margin-bottom: 10px;  margin-left: 70px;  }  Или:  body {  margin: 100px 40px 10px 70px;  } |

### Отступы (внутренние поля) [padding]

Определяет внутреннее расстояние между рамкой и содержимым элемента. Использование padding можно показать на простом примере, где все заголовки имеют цветной фон. Определяя заполнение для заголовков, вы устанавливаете величину поля вокруг текста каждого заголовка:

h1 {

background: yellow;

**padding: 20px 20px 20px 80px;**

}

h2 {

background: orange;

**padding-left:120px;**

}

width

height

### Ширина [width]

Свойством width вы можете определять ширину элемента. В примере показан блок для ввода текста:

div.box {

width: 200px;

border: 1px solid black;

background: orange;

}

### Высота [height]

Обратите внимание, как, в предыдущем примере, высота блока устанавливается содержимым этого бокса. Высоту элемента можно определить свойством height. В качестве примера попытаемся создать блок с высотой 500px:

div.box {

height: 500px;

width: 200px;

border: 1px solid black;

background: orange;

}

### Поплавок [float]

Определяет, по какой стороне будет выравниваться элемент, при этом остальные элементы будут обтекать его с других сторон. Когда значение свойства float равно none, элемент выводится на странице как обычно, при этом допускается, что одна строка обтекающего текста может быть на той же линии, что и сам элемент.

Возможные значения:

**left**

Выравнивает элемент по левому краю, а все остальные элементы, вроде текста, обтекают его по правой стороне.

**right**

Выравнивает элемент по правому краю, а все остальные элементы обтекают его по левой стороне.

**none**

Обтекание элемента не задается.

**inherit**

Наследует значение родителя.

img.leftside {

margin: 10px 10px 10px 10px;

border: 1px solid red;

float: left;

}

## Рамки

border-width

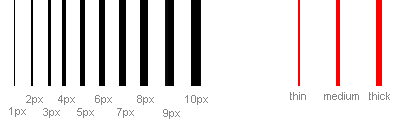
border-color

border-style

border

### Толщина рамки [border-width]

Толщина рамки определяется свойством border-width, которое может иметь значения thin, medium и thick, или числовое значение в пикселах.



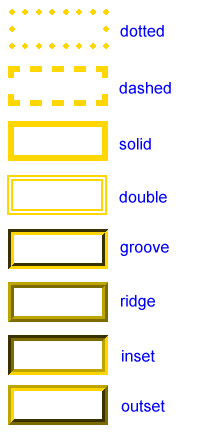
### Цвет рамки [border-color]

Свойство border-color определяет цвет рамки. Значения - нормальные значения цвета, например: "#123456", "rgb(123,123,123)" или "yellow" .

### Стиль рамки [border-style]

Существуют различные типы рамок. Ниже показаны восемь типов рамки и их интерпретация в Internet Explorer 5.5. Все примеры показаны цветом "gold" и толщиной "thick".

Значения none или hidden могут использоваться, если вы не хотите отображать рамку.



### Примеры определения рамок

Три рассмотренных выше свойства можно объединить в каждом элементе и, соответственно, устанавливать разные рамки. Для иллюстрации взглянем на документ, где определены разные рамки для <h1>, <h2>, <ul> и <p>:

h1 {

border-width: thick;

border-style: dotted;

border-color: gold;

}

h2 {

border-width: 20px;

border-style: outset;

border-color: red;

}

p {

border-width: 1px;

border-style: dashed;

border-color: blue;

}

ul {

border-width: thin;

border-style: solid;

border-color: orange;

}

Можно также установить специальные свойства для верхнего, нижнего, правого и левого края рамки:

h1 {

border-top-width: thick;

border-top-style: solid;

border-top-color: red;

border-bottom-width: thick;

border-bottom-style: solid;

border-bottom-color: blue;

border-right-width: thick;

border-right-style: solid;

border-right-color: green;

border-left-width: thick;

border-left-style: solid;

border-left-color: orange;

}

### Сокращённая запись [border]

Пример:

p {

border-width: 1px;

border-style: solid;

border-color: blue;

}

можно объединить в:

p {

border: 1px solid blue;

}

## Тэги <span> и <div>

В самом HTML эти тэги никак не влияют на отображение документов. Они служат лишь для того, чтобы указать CSS-стиль для какой-то его части.

Тэг <span> используется для оформления текста *внутри* абзаца (или другого блока). Тэг <div> используется для назначения стиля одному или несколькимблокам *целиком*.

<style>

body {

font-family: Arial, sans-serif;

}

.citation {

font-family: Times, serif;

font-style: italic;

margin: 10px 5px;

padding: 5px;

background-color: #FFFFF0;

}

.keyword {

color: red;

font-size: 150%;

}

</style>

...

<div class="citation">

<p>Разумные люди приспосабливаются к окружающему миру. Неразумные люди приспосабливают мир к себе. Вот почему <span class="keyword">прогресс</span> определяется действиями неразумных людей.</p>

<p align=right>Бернард Шоу</p>

</div>

# Пример

Файл главной страницы **index.html**:

<html>

<head>

<title>Моя первая страничка с CSS</title>

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

</head>

<body>

<!-- МЕНЮ -->

<ul class="navbar">

<li><a href="index.html">Главная</a>

<li><a href="city.html">Мой город</a>

<li><a href="about.html">Обо мне</a>

<li><a href="contacts.html">Контакты</a>

<li><a href="links.html">Ссылки</a>

</ul>

<!-- ГЛАВНАЯ -->

<h1> Моя первая страничка с CSS </h1>

<div class="block1">

<p>Здесь нужно добавить еще что-то, но я еще не знаю, что именно.</p>

</div>

<p>Добро пожаловать на мою первую страничку!</p>

<p>На ней нет картинок, но зато есть стили. И ссылки, даже если они никуда не ведут;)</p>

<!-- КОПИРАЙТ -->

<address>14 октября 2013 &copy; Я</address>

</body>

</html>

**Файл стилей mystyle.css:**

/\* ТЕЛО ДОКУМЕНТА \*/

body {

padding-left: 11em;

font-family: Georgia, "Times New Roman",

Times, serif;

color: purple;

background-color: #d8da3d }

/\* МЕНЮ в виде списка \*/

ul.navbar {

list-style-type: none;

padding: 0;

margin: 0;

position: absolute;

top: 2em;

left: 1em;

width: 9em }

ul.navbar li {

background: white;

margin: 0.5em 0;

padding: 0.3em;

border-right: 1em solid black }

ul.navbar a {

text-decoration: none }

/\* ЗАГОЛОВОК 1 \*/

h1 {

font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif; }

/\* ССЫЛКИ \*/

a:link {

color: blue }

a:visited {

color: purple }

/\* КОПИРАЙТ внизу страницы \*/

address {

margin-top: 1em;

padding-top: 1em;

border-top: thin dotted;

font-size: 90%; }

/\* Текст в РАМОЧКЕ \*/

.block1 {

font-family: Georgia, "Times New Roman",

Times, serif;

width: 200px;

background: #ccc;

padding: 5px;

padding-right: 20px;

border: solid 1px black;

float: right;}

# Задание

Создайте собственный стиль оформления для приведенного примера. Добавьте недостающие странички. Задайте следующие стили:

1. Основной текст.
2. Заголовки 1-3 уровней.
3. Меню, используя тэг <div>.
4. Гиперссылки.
5. Рисунок.
6. Произвольный стиль для тэгов <div> и/или <span>, например: эпиграф, определение, цитата, заметка.

# Рекомендуемая литература

1. Справочник CSS. – URL: <http://htmlbook.ru/css>
2. Учебник CSS. – URL: <http://ru.html.net/tutorials/css/>
3. CSS Validation Service. – URL: <http://jigsaw.w3.org/css-validator/validator.html.ru>