

Каскадные таблицы стилей

Каскадные таблицы стилей (или CSS – Cascading Style Sheets) можно встроить в документ тремя способами:

- Используя тег `<link>` с указанием URL внешней таблицы стилей.

```
<link rel="stylesheet" type="text/css" href="http://your.url.com/anyfile.css">
```

- Используя тег `<style>` внутри заголовка документа, т.е. внутри тега `<head>`.

```
<style type="text/css">
    ....
    таблица стилей
    ....
</style>
```

- Используя атрибут `style` в параметрах тега. Например, если необходимо получить красный параграф:

```
<p style="color: red;">Красный параграф.</p>
```

Таблица стилей это запись атрибутов тегов в довольно простом формате:

```
имя_тега {параметр_тега1: значение; ... }
имя_тега.класс {параметр_тега1: значение; ... }
.класс {параметр_тега1: значение; ... }
```

Имя тега, для которого будут действовать заданные настройки указывается в поле **имя_тега**. В фигурных скобках указываются параметры тега с их значениями, о которых будет сказано ниже.

Воспользовавшись описанием в первом формате, вы обеспечиваете всем тегам с этим именем одинаковое отображение. Если же вас не устраивает, что все теги с одинаковым именем в документе отображаются одинаково, то вы можете создать класс, или даже лучше сказать подкласс тегов с таким именем, но имеющих другое оформление. Для задания имени этого нового подкласса строка таблицы стилей записывается по второму формату. Но теперь в теге с именем **имя_тега** для использования нового класса нужно будет использовать атрибут **class**. Можно рассмотреть следующий пример:

```
<style type="text/css">
    p {color: red;}
</style>
```

После создания такой таблицы стилей все параграфы в документе будут отображаться красным цветом. Естественно это может показаться не нужным для всего документа. Тогда будем использовать второй формат:

```
<style type="text/css">
    p.red {color: red;}
</style>
```

При таком задании таблицы все параграфы документа будут отображаться по умолчанию, или с ранее заданными настройками. Для того чтобы окрасить параграф в красный цвет нам придется использовать тег с атрибутом **class**:

```
<p class="red">Red paragraph.</p>
```

В данном случае класс **red** будет распространяться только на теги параграфов **<p>**. Если же необходимо использовать этот класс в любых других тегах, то используется запись по третьему формату, т.е. без указания имени тега. При этом использование атрибута **class** у любого из тегов будет ему соответствующий класс для отображения из таблицы стилей.

В такой технологии раскрашивания есть еще очень большая и хорошая возможность по заданию стиля только определенным тегам. Заключается она в том, что если перед именем тега, для которого составляется стиль, написать какой-то список других тегов, то будет действовать принцип вложенности. Т.е. придуманный стиль будет действовать только на теги с заданным именем и еще вложенным внутри тех тегов, имена которых записали ранее. Пример:

```
p b { color: green}
```

При такой записи тег **** будет окрашиваться в зеленый цвет, если он находится внутри параграфа:

```
<p>Обычный текст, а это <b>зеленый текст</b>.</p> <b>А это все же обычный текст...</b>
```

CSS: Параметры шрифтов

Первым и наиболее важным параметром для любого шрифта является его размер. Размер шрифта в CSS задается параметром **font-size**. Этот параметр может задаваться несколькими способами. Первый способ это задание абсолютного размера букв шрифта, он указывается в пискелах, при этом к числу надо приписывать слово **px**, хотя если ничего нет, то браузер воспринимает обычные числа как количество пикселей. Например:

```
p { font-size: 15px; }  
b { font-size: 15; }
```

В данном примере оба элемента будут отображаться размером в 15 пикселей.

Броузер имеет внутри таблицу стандартных размеров шрифтов, т.е. тех размеров, которые задаются атрибутом **size** в теге ****. Для установки этих семи стандартных значений можно пользоваться встроенными названиями: **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**. Значение **xx-small** соответствует размеру 1, а **xx-large** самому большому 7.

Вторым способом задания размера является относительный размер. Для увеличения размера используется значение **larger**, а для уменьшения **smaller**. Эти выражения имеют такой же смысл, как и стандартные способы задания размера шрифта **size=+1** или **size=-1**. Естественно, что в данном случае будет только семь значений размера шрифта. Если необходимо получить иной более точный размер, то его можно задать в процентной форме или с добавлением к числовому выражению суффикса **em** или **ex**:

```
p { font-size: xx-large; }  
p { font-size: 1.5em; }  
p { font-size: 150%; }  
p { font-size: 3ex; }
```

Суффиксы **em** и **ex** имеют разные результаты. Суффикс **em** действует от текущего размера шрифта, а размер заданный через **ex** считается относительным от единичного размера шрифта, т.е. от **size=1**.

Следующим характерным для шрифта параметром является его жирность или, по другому, вес. Задается вес параметром **font-weight**. Для него существует 9 стандартных значений жирности от 100 до 900, т.е. 100, 200, 300... Некоторым из этих значений присвоены свои имена. Например, шрифт с нормальным весом 400 может обозначаться словом **normal**, и соответственно жирный весом в 700 через **bold**. Также здесь, как и везде, характерен способ задания относительной жирности. Чтобы повысить жирность шрифта используется слово **bolder**, а для понижения **lighter**.

Например, для того, чтобы сделать вывод шрифта в параграфах полужирным начертанием:

```
p {font-weight: bold;}
```

Начертание (гарнитура) шрифта задается параметром **font-family**. Значением этого параметра может служить точное название выбранного шрифта, например, **Times New Roman** или **Arial**. Но на разных платформах может не существовать таких шрифтов, поэтому есть возможность задавать не просто названия шрифтов, но и целые семейства подобных по стилю начертания шрифтов. Для обозначения этих семейств ввели несколько стандартных значений: **serif**, **sans-serif**, **cursive**, **fantasy**, **monospace**.

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

Шрифт может изображаться двумя способами: нормально и наклонно, т.е. курсивом. Поэтому для этого существует параметр **font-style**, который может иметь два значения **normal** и **italic** или **oblique**. Последние два значения для браузера считаются идентичными.

Также существуют шрифты, в которых прописные буквы и строчные изображаются идентично, но разным размером или небольшими отличиями в пропорциях. Для того чтобы браузер заменил все строчные буквы на заглавные и изменил их размер необходимо использовать параметр **font-variant** со значением **small-caps**, по умолчанию этот параметр равен **normal**.

Также существует способ задания всех этих параметров шрифта разом. Это обеспечивает параметр **font** и в нем задаются все выше перечисленные параметры шрифта через пробел в любой последовательности:

```
p { font: bold italic x-large Arial, serif }  
p { font: normal small-caps 120% fantasy }
```

CSS: Параметры текста

Язык **html** в принципе позволяет достаточно гибко управлять параметрами текста, но все же существуют некоторые его свойства, которые способны изменить только таблицы стилей **CSS**.

Первым параметром, который обычно очень важен для оформления, является выравнивание. Этим свойством управляет **text-align** и может принимать следующие значения:

left – выравнивание относительно левого края

right – относительно правого края

center – относительно центра, т.е. центрирование

justify – выравнивание по всей ширине документа (или области для текста)

В принципе можно для этих целей использовать уже всеми принятый атрибут тегов **align**, но если в документе встречается много структур с определенным выравниванием, то экономичнее описать его сразу в таблице стилей.

К этому же числу относится и параметр **vertical-align**, который задает выравнивание по вертикали. Его значения могут быть следующими:

baseline – выравнивается относительно «базовой линии» родительского элемента.

middle – выравнивается относительно середины родительского элемента, получается прибавлением к «базовой линии» половины полной высоты элемента.

sub, supper – эти значения аналогичны тегам **<sub>** и **<sup>**, которые делают текст в индексном или степенном пространстве строки, т.е. ниже или выше средней линии.

text-top, top – оба значения выравнивают текст по верху. При этом **top** выравнивает текст относительно всех элементов, а **text-top** относительно самого высокого символа в тексте.

text-bottom, bottom – выравнивают аналогично по низу элемента или текста.

Также можно задать расстояние между двумя «базовыми линиями» двух строчек. Задается он параметром **line-height**, который может характеризоваться своим точным размером и процентным отношением. При использовании этого параметра следует использовать также параметр задания высоты текста, это позволит предотвратить наезжание двух строк текста, если размер шрифта вдруг станет больше.

```
div { line-height: 1.2; font-size: 10pt }
```

Следующим свойством текста большинство тегов никогда не обладало. Имеется в виду отступ в первой строке, или «красная строка». Например, раньше такого эффекта можно было добиться только использованием **<dd>** без указания тега списка. Теперь это позволяет устанавливать параметр **text-indent**. Отступ указывается с помощью указания его длины или в процентном отношении. Значение этого параметра может быть также отрицательным. Еще следует отметить, что такой отступ действует только в начале тега, т.е. он не действует при разрывах в середине, например при использовании тега **
**.

```
p { text-indent: 3cm }
```

В свойствах шрифта был параметр, который преобразовывал все символы текста в заглавные. Его возможности достаточно ограниченные. У текста же существует параметр **text-transform**, который позволяет делать с текстом больше преобразований:

capitalize – делает в каждом слове первую букву заглавной.

uppercase – весь текст делает заглавными буквами.

lowercase – весь текст маленькими

none – оставить, как есть

В продолжение темы оформления текста следует сказать о параметре **text-decoration**, который позволяет перечеркивать или подчеркивать текст разными способами. Для этого существуют стандартные значения **underline**, **overline** или **line-through**. Для нормального текста рекомендуется использовать значение **none**. Этот параметр, как правило, используется для оформления ссылок, так как почти всегда на странице ссылки являются единственным подчеркнутым элементом.

Есть еще два параметра, которые игнорируются некоторыми браузерами. Эти параметры задают расстояние между символами и между словами в тексте. Называются соответственно **letter-spacing** и **word-spacing**. Как и всегда задаются с помощью размера промежутка в пикселах или относительно текущего размера. По умолчанию имеют значение **normal**.

Единицы измерения

Что бы вы ни пытались сделать с помощью CSS, без единиц измерения вам не обойтись. В языке HTML существует два способа задать линейные размеры элемента: пиксели и проценты. Например, ширину таблицы можно задать так:

```
<table width="750">
```

что будет соответствовать 750-ти пикселям. А можно вот так:

```
<table width="100%">
```

что будет соответствовать ста процентам от ширины элемента-предка данного элемента. Если эта таблица является непосредственным предком элемента *BODY*, то она будет занимать все окно браузера.

В CSS единиц длины гораздо больше. Однако сразу надо сказать, что использовать вы будете все те же пиксели, все те же проценты, и время от времени величину *em*.

Вообще происхождение единиц в CSS позволяет разделить их на три группы. Первая группа – это величины, пришедшие из **реального мира**, которые используются повсеместно для измерения длин реальных предметов. К ним относятся:

- *in* – дюймы;
- *cm* – сантиметры;
- *mm* – миллиметры.

Данная группа величин привычна всем, в этом ее достоинство.

Ко второй группе можно отнести величины, которые пришли в CSS из **типографии**. То есть они используются для установки размеров шрифта, межстрочных интервалов и прочих типографских величин. К ним относятся:

- *pt* – типографский пункт;
- *pc* – пика;
- *ex* – высота строчной буквы "x" в шрифте.

Данная группа величин привычна полиграфистам. Очень часто в графических пакетах (например, Photoshop) размер шрифта устанавливается в пунктах.

Третью группу составляют величины, которые являются **относительными**, то есть реальный размер элемента вычисляется относительно какой-либо иной величины. К ним относятся:

- *em* – вычисляется относительно размера шрифта элемента;
- *px* – пиксель. Вычисляется относительно устройства отображения;
- *%* – процент. Естественно, процентные соотношения не являются единицами измерения в привычном смысле слова. Они вычисляются относительно размеров элемента-предка.

Данная группа пришла, в общем-то, из языка HTML. Так что очень многие пользуются именно этими единицами. Конечно, есть еще очень много причин, которые приводят к тому, что первые две группы неудобны в использовании. Поговорим об этом подробнее.

Реальные единицы измерения

Реальные единицы измерения великолепно работают в физическом мире, однако для использования на экране монитора они оказываются **мало пригодны**. Причина на редкость проста: мониторы имеют совершенно разные физические параметры. К ним относятся:

- размер монитора по диагонали (варьируется от 14 дюймов до 21 дюйма и больше);
- разрешение монитора (варьируется от 640x480 до 1600x1200 и выше);

Кроме того, на различных платформах могут быть различные экранные разрешения. Так, например, на платформе Macintosh разрешение 72 пикселя на дюйм, тогда как на платформе Windows – 96 пикселей на дюйм.

К чему это приводит? Если на мониторе с диагональю 21 дюйм шрифт размером 0.5 дюйма будет смотреться нормально, то на мониторе с диагональю 14 дюймов он будет слишком велик.

Точно так же, если при разрешении 640x480 текст размером *3mm* читается нормально, то при разрешении 1600x1200 невозможно будет ничего разобрать.

Кроме того, браузер сам по себе не в состоянии корректно отобразить величину в реальных единицах. Давайте рассмотрим, почему это происходит. Как известно, изображение на экране монитора строится на основе пикселей. Пиксель является **минимальной единицей измерения**, то есть не бывает величины *0.5 пикселя*. Чтобы отобразить на экране, скажем, два дюйма, операционная система переводит реальную величину в пиксели. И тут вступают в расчет все вышеперечисленные факторы. Во-первых, физические размеры монитора, во-вторых, разрешение монитора, в-третьих, экранное разрешение.

Возьмем конкретный пример. Монитор размером 13 дюймов по горизонтали и 9 дюймов по вертикали с разрешением 1024x768 в операционной системе Windows, то есть с экранным разрешением 96 пикселей на дюйм. Тогда величина два дюйма будет соответствовать $96 * 2 = 192$ пикселям. Однако реальная высота этих двух дюймов будет $(768/9) * 2 = 171$ пиксель. То есть получается несоответствие: браузер считает, что два дюйма – это 192 пикселя, а реальный размер двух дюймов на мониторе – 171 пиксель. Таким образом, если вы устанавливаете размер элемента в два дюйма, то на этом конкретном мониторе реальный размер элемента будет равен 2,22 дюйма. В то же время на другом мониторе с другими параметрами реальный размер может быть совершенно иным.

Исходя из всего этого, можно сделать простой вывод: **использование в CSS реальных единиц измерения крайне ограничено**. Принципиально использовать реальные единицы измерения можно только в том случае, когда **известны параметры устройства вывода**. Таким устройством может быть принтер, потому что для него известны геометрические размеры страницы и разрешение (например, 300 точек на дюйм и страница формата А4). Однако особого смысла использование реальных единиц и для печати не имеет.

Типографские единицы

Типографские единицы измерения для дизайнера даже привычнее, чем реальные единицы, потому что они сталкиваются с ними повсеместно, в любом графическом пакете размер шрифта задается с помощью пунктов (*pt*). Так что использовать типографские единицы в CSS хочется по привычке. Однако придется изменить свои привычки по той простой причине, что пункты (*pt*) и пики (*pc*) принципиально ничем не отличаются от реальных единиц измерения. Это становится понятно, если вспомнить, что такое пункт. Так вот один пункт равен 1/72 дюйма, а пика равна 12 пунктам, то есть 1/6 дюйма. Таким образом, **на использование данных типографских величин накладывается абсолютно такое же ограничение, как и на использование реальных величин**. Получается, что использовать пункты и пики надо только в той таблице стилей, которая предназначается для вывода страницы на печать.

Надо сказать, что для принтера действительно лучше использовать типографские единицы, потому что они для него являются родными. Дело в том, что некоторые достаточно старые браузеры некорректно переводят пиксели в пункты. Например, если вы укажете размер шрифта 16 пикселей, то некоторые браузеры и распечатают его как шестнадцать точек. Но экранное разрешение значительно отличается от разрешения принтера. Например, принтер с разрешением 600dpi выведет 16 пикселей как 0,03 дюйма, что читаться не будет совершенно.

Все вышесказанное ни в коей мере не относится к браузерам пятых и тем более шестых версий. Они совершенно корректно трансформируют пиксели при печати, так что можно не беспокоиться за неправильную распечатку страниц.

Кроме пунктов и пик есть еще одна достаточно интересная типографская единица – *ex*, которая соответствует высоте строчной буквы "x". В отличие от всех предыдущих единиц, эта единица является относительной. Относительные единицы просто незаменимы в резиновой верстке, когда главным является сохранение пропорций. Что касается высоты буквы "x", то она может значительно варьироваться от шрифта к шрифту. У большинства шрифтов **высоты буквы "x" равна половине высоты шрифта**, однако у некоторых декоративных шрифтов может составлять всего одну треть от высоты шрифта.

Для чего же можно использовать единицу измерения, которая базируется на высоте буквы "x"? На самом деле, область применения данной единицы достаточно узка. Ее предпочтительно использовать для выравнивания по вертикали, чтобы точно выровнять какой-либо блок по линии текста. Кроме того, можно делать рамки, отступы и поля шириной по высоте текста. Вот, в общем-то, и все.

Для установки размера шрифта *ex* подходит слабо, потому что он обладает узким спектром значений. Например, невозможно задать размер шрифта на 25% больше, чем у родительского элемента. Для установки высоты строки тоже не подходит, потому что надо знать полную высоту шрифта, иначе расстояние между строками может быть совсем не таким, каким задумывалось.

Что касается поддержки единицы *ex* в различных браузерах, то можно сказать следующее. В четвертых версиях браузеров Microsoft и Netscape данная единица измерения не поддерживается вовсе. В браузере **Internet Explorer 5.0** *1ex* равен *0.5em*, что абсолютно неверно, потому что у разных шрифтов соотношение высоты буквы "x" и высоты шрифта может быть и *0.3em*, и *0.6em*. Так что использованию данной единицы измерения мешает еще и плохая реализация ее поддержки браузерами.

Свойства таблиц стилей

СВОЙСТВА ШРИФТА

font-family

Используется для указания шрифта или шрифтового семейства, которым будет отображаться элемент.

```
p {font-family: Times New Roman, sans-serif;}
```

font-style

Задаёт способ начертания шрифта: normal – Нормальный (по умолчанию), italic – Курсив, oblique – Наклонный.

```
p {font-style: italic;}
```

font-variant

Задаёт варианты начертания шрифта: normal – Нормальный (по умолчанию), small-caps – Все буквы заглавные).

```
p {font-variant: small-caps;}
```

font-weight

Определяет степень жирности шрифта с помощью параметров: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900

```
b {font-weight: bolder;}
```

font-size

Устанавливает размер шрифта. Параметр может указываться как в относительной (проценты), так и абсолютной величине (пункты, пиксели, сантиметры)

```
h1 {font-size: 200%;}  
h2 {font-size: 150px;}  
h3 {font-size: 400pt;}
```

Примечания

font-family – Можно указать до трех семейств шрифта. Это необходимо, так как, если на компьютере клиента отсутствует, скажем, семейство Verdana, а в правиле указано только оно, то разработчик не может предположить, как страница будет выглядеть на экране пользователя. В том, случае, когда в правилах указано несколько семейств, браузер будет последовательно перебирать все перечисленные, пока не найдет того, которое присутствует на машине клиента. Браузер при запуске сам распределяет имеющиеся семейства для подстановки вместо стандартных.

font-weight – Значения bolder и lighter являются относительными. Т.е. если элемент является дочерним к элементу со значением font-weight: bold и правило font-weight для него задано lighter, то он будет отображен нормальным шрифтом.

font-size – В абсолютных размерах можно использовать некоторые predetermined значения: xx-small | x-small | small | medium | large | x-large | xx-large. При использовании относительных размеров задаются размеры шрифта относительно родительского элемента. Возможные значения: larger, smaller. И наконец, можно использовать точные значения – задаются размеры шрифта в единицах длины, допускаемых в рекомендациях W3C. Это: px – логические пиксели, pt – логические пункты, em – единицы размера шрифта. em подразумевает, что размер шрифта устанавливается относительно размера шрифта в родительском элементе. 1em равно размеру шрифта родительского элемента. Либо используются проценты – задаются размеры шрифта в процентах по отношению к родительскому элементу.

ЦВЕТ ЭЛЕМЕНТА И ЦВЕТ ФОНА

color

Определяет цвет элемента

```
i {color: green;}
```

background-color

Устанавливает цвет фона для элемента – именно для элемента, а не для странички. Разные браузеры отображают это свойство по-разному: Microsoft IE отводит под фон элемента всю доступную ширину страницы, а Netscape Navigator – лишь ширину, занимаемую этим элементом.

```
h4 {background-color: yellow;}
```

В скриптовых языках используйте `object.style.backgroundColor [= sColor]`

background-image

Устанавливает или получает фоновый рисунок для элемента. Может принимать значения

- none – По умолчанию. Используется цвет родительского объекта
- Url – Задается абсолютный или относительный путь к рисунку

```
h3 {background-image: url(image.gif);}
```

В скриптовых языках используйте `object.style.backgroundImage [= sLocation]`

background-attachment

Устанавливает или получает поведение фонового рисунка для элемента. Может принимать значения

- scroll – По умолчанию. Фоновый рисунок прокручивается вместе с элементом
- fixed – Фоновый рисунок не прокручивается вместе с элементом

```
body {background-attachment: fixed;}
```

В скриптовых языках используйте `object.style.backgroundAttachment [= sAttachment]`

background-repeat

Свойство, определяющее, надо ли повторять фоновый рисунок для заполнения пространства элемента.

Значения:

- repeat – заполнять все пространство элемента;
- repeat-x – заполнить только первый горизонтальный ряд;
- repeat-y – заполнить только первый вертикальный столбец;
- no-repeat – не заполнять.

```
body { background-image: url(kvadrat.gif); background-repeat: repeat-y; }
```

background-position

Начальное положение фонового рисунка.

Значениями свойства являются две величины в численном (или процентном) выражении, а также символные коды:

- 0% 0% – значение по умолчанию;
- top left или left top – значение 0% 0%;
- top, top center или center top – значение 50% 0%;
- right top или top right – значение 100% 0%;
- left, left center или center left – значение 0% 50%;
- center или center center – значение 50% 50%;
- right, right center или center right – значение 100% 50%;
- bottom left или left bottom – значение 0% 100%;
- bottom, bottom center или center bottom – значение 50% 100%;
- bottom right или right bottom – значение 100% 100%.

Если указана одна величина, она интерпретируется как значение отступа по горизонтали, а значение по вертикали принимается равным 50%.

```
body { background-image: url(banner.gif); background-position: top center }
```

background

Фон элемента. Одновременное задание свойств background-color, background-image, background-repeat, background-attachment и background-position.

```
td { background: url(http://сервер.com/metal.jpg) }  
p { background: url(sphere.gif) gray 50% repeat fixed }
```

СВОЙСТВА ТЕКСТА

text-decoration

Устанавливает эффекты оформления шрифта, такие, как подчеркивание или зачеркнутый текст

```
h4 {text-decoration: underline;} - подчеркивание  
a {text-decoration: none;} - стандартный текст  
i {text-decoration: line-through;} - зачеркивание  
b {text-decoration: overline;} - надчеркивание
```

text-transform

Задаёт преобразование регистра текста при отображении

```
h4 {text-transform: capitalize;} - Первая буква каждого слова преобразуется в заглавную  
a {text-transform: uppercase;} - Все буквы преобразуются в заглавные  
i {text-transform: lowercase;} - Все буквы преобразуются в строчные  
b {text-decoration: none;} - Отменяет установленные преобразования
```

text-align

Определяет выравнивание элемента. Возможные значения: left, right, center, justify

```
p {text-align: justify}  
h5 {text-align: center}
```

text-indent

Устанавливает отступ первой строки текста. Чаще всего используется для создания параграфов с табулированной первой строкой.

```
p {text-indent: 50pt;}
```

line-height

Управляет интервалами между строками текста.

```
p {line-height: 50 %}
```

word-spacing

Устанавливает интервалам между словами. Можно использовать отрицательные значения

```
p {word-spacing: 50 %}
```

letter-spacing

Устанавливает интервалам между буквами

```
p {letter-spacing: 50 pt}
```

vertical-align

Устанавливает вертикальное положение элемента. Может принимать следующие значения: baseline middle sub super text-top text-bottom top bottom

```
p {vertical-align: 50 pt}
```

white-space

Определение вида пробелов.

Значения:

- normal – автоматическое форматирование текста. Например, удаляются пробелы, следующие друг за другом;
- pre – текст остается без изменения (по аналогии с элементом PRE);
- nowrap – запрет на автоматический разрыв строк.

СВОЙСТВА СПИСКОВ

list-style-type

Определение вида маркеров в списке.

Значения:

- none – без маркеров;
- disc – круги;
- circle – окружности;
- square – квадраты;
- decimal – арабские цифры;
- lower-roman – римские цифры на основе строчных латинских букв (i, v, x...);
- upper-roman – римские цифры на основе прописных латинских букв (I, V, X...);
- lower-alpha – строчные латинские буквы (a, b, c...);
- upper-alpha – прописные латинские буквы (A, B, C...).

```
li {list-style-type: lower-alpha}
```

list-style-image

Определение рисунка, который будет использоваться в качестве маркера.

Значения:

- none – рисунок не используется;
- адрес (URL) графического файла.

```
ul {list-style-image: url(marker.gif)}
```

list-style-position

Положение маркера относительно списка.

Значения:

- inside – маркер внутри списка (компактная форма);
- outside – маркер вне списка.

list-style

Определяет вид маркеров. В качестве значения может использоваться несколько величин свойств list-style-type, list-style-position, list-style-image (перечисленных выше).

```
ul {list-style: circle outside}
```

СВОЙСТВА ГРАНИЦ ЭЛЕМЕНТОВ

width

Ширина объекта.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

min-width и max-width

Минимально возможная и максимально допустимая ширина объекта.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

height

Высота объекта.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

min-height и max-height

Минимально возможная и максимально допустимая высота объекта.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

position

Способ вычисления координат границ элемента.

Значения:

- normal – положение элемента определяется браузером;
- relative – вначале вычисляется обычное положение элемента, затем рассчитывается смещение относительно этого положения;
- absolute – координаты вычисляются относительно границ контейнера, содержащего элемент;
- fixed – координаты вычисляются относительно границ контейнера, содержащего элемент, и запрещается прокрутка.

top, bottom, right, left

Координаты верхней, нижней, правой и левой границ элемента.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

margin-top, margin-right, margin-bottom, margin-left

Размер пустого пространства рядом с определенной стороной элемента.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

margin

Величина отступов вокруг элемента.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

```
p { margin: 12px 12px 12px 12px }
```

padding-top, padding-right, padding-bottom, padding-left

Величина отступа между указанной границей элемента и его содержимым.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

padding

Величина отступа между границами элемента и его содержимым.

Значения:

- auto – определяется браузером;
- числовое;
- процентное.

```
ol { padding: 4px 4px 4px 4px }
```

border-top-width, border-right-width, border-bottom-width, border-left-width

Ширина элементов рамки.

Значения:

- thin – тонкая;
- medium – средняя;
- thick – широкая;
- числовое.

border-width

Толщина рамки.

Значения:

- thin – тонкая;
- medium – средняя;
- thick – широкая;
- числовое.

```
table {border-width: 2px;}
```

border-top-color, border-right-color, border-bottom-color, border-left-color

Цвет элементов рамки.

border-color

Цвет границы. Может принимать значение transparent для задания невидимой, но имеющей ширину, границы, что иногда полезно

```
table {border-color: green;}
```

border-top-style, border-right-style, border-bottomstyle, border-left-style

Вид элементов рамки.

Значения:

- none или hidden – рамка не показана;
- dotted – рамка из точек;
- dashed – пунктирная рамка;
- solid – сплошная рамка;
- double – двойная рамка;
- groove – двугранная рамка;
- ridge – такая же, как groove, но светлая и темная грани расположены иначе;
- inset – одна грань, наклоненная внутрь;
- outset – такая же, как inset, но свет падает по-другому.

border-style

Вид рамки.

Значения такие же, как и у предыдущих свойств.

border-top

Одновременное задание значений свойств border-top-width, border-top-style и border-top-color.

```
h1 { border-top: thin solid blue }
```

border-bottom

Одновременное задание значений свойств border-bottom-width, border-bottom-style и border-bottom-color.

border-left

Одновременное задание значений свойств border-left-width, border-left-style и border-left-color.

border-right

Одновременное задание значений свойств border-right-width, border-right-style и border-right-color.

border

Одновременное задание значений свойств border-width, border-style и border-color.

```
p { border: medium double red }
```

border-collapse

Задаёт стиль рисования таблицы. Может принимать следующие значения:

- separate (по умолчанию) – ячейки отделены друг от друга
- collapse – ячейки не имеют промежутков между собой

```
<table style ="border-collapse:collapse">
```

overflow

Способ изменения размеров объекта, если его содержимое не может быть показано целиком.

Значения:

- auto – определяется браузером;
- visible – размер границ увеличивается в такой степени, чтобы содержимое было видимым;
- hidden – размеры границ не устанавливаются в зависимости от размеров содержимого;
- scroll – устанавливается режим прокрутки.

float

Способ обтекания элемента другими.

Значения:

- none – обтекания нет;
- left – слева;
- right – справа.

```
img { float: left }
```

clear

Способ расположения элементов, следующих за элементом, выровненным по левому или правому краю.

Значения: none, left, right, both.

clip

Определение видимой части объекта.

Значения:

- auto – определяется браузером;
- прямоугольник, «вырезающий» часть объекта.

```
p { clip: rect(5px, 10px, 10px, 5px); }
```

visibility

Видимость или невидимость объекта.

Значения:

- visible – видимый;
- hidden – невидимый.

z-index

Способ перекрытия объектов другими. Объект, имеющий большее значение индекса, располагается выше.

Значения:

- auto – «нулевой уровень», задаваемый по умолчанию;
- отрицательное числовое – объект располагается «ниже» нулевого уровня;
- положительное числовое – объект располагается «выше» нулевого уровня.

СВОЙСТВА ТАБЛИЦ

display

Создание таблиц и элементов таблиц.

Значения:

- table – таблица (аналог элемента TABLE);
- table-caption – заголовок таблицы (аналог элемента CAPTION);
- table-column – колонка таблицы (аналог элемента COL);
- table-column-group – группа колонок (аналог элемента COLGROUP);
- table-row – строка таблицы (аналог элемента TR);
- table-row-group – группа строк (аналог элемента TBODY);
- table-header-group – группа строк в начале таблицы (аналог элемента THEAD);
- table-footer-group – группа строк в конце таблицы (аналог элемента TFOOT);
- table-cell – ячейка таблицы (аналог элемента TD);

row-span

Количество строк, которые должна занять ячейка.

column-span

Количество столбцов, которые должна занять ячейка.

border-collapse

Способ прорисовки рамки.

Значения:

- separate – стиль броузеров (трехмерная рамка);
- collapse – стиль текстовых редакторов (плоская рамка).

border

Характеристика рамки.

Значение свойства состоит из трех элементов: толщина, тип, цвет.

Возможные типы:

- none или hidden – рамка не показана;
- dotted – рамка из точек;
- dashed – пунктирная рамка;
- solid – сплошная рамка;
- double – двойная рамка;
- groove – двугранная рамка;
- ridge – такая же, как groove, но светлая и темная грани расположены иначе;
- inset – одна грань, наклоненная внутрь;
- outset – такая же, как inset, но свет падает по-другому.

```
table { border: 1px solid blue; }  
<table style="border-collapse: collapse; border: solid green;">  
<td style="border: double">
```

vertical-align

Способ (степень) вертикального выравнивания.

Значения:

- процентное;
- top – по верхнему краю;
- middle – по центру;
- bottom – по нижнему краю;
- baseline – по первой строке текста в ячейке.

table-layout

Алгоритм форматирования таблицы.

Значения:

- auto – стандартный;
- fixed – ускоренный.

ПСЕВДО-ЭЛЕМЕНТЫ

Псевдо-элементы являются особой группой, позволяющих объединять несколько стилей для какого-либо объекта. Например, вы можете задать свойства для первой буквы параграфа. Для этого вы назначаете для тега **P** псевдо-элемент **first-letter**, в котором устанавливаете различные стили:

```
p:first-letter { float: right; font-size: 2em; color: red; }
```

:first-letter

Устанавливает стили для первой буквы объекта

:first-line

Устанавливает стили для первой строки объекта

:hover

Устанавливает стили для элемента, когда пользователь подводит курсор мыши к ссылке. Данный псевдокласс часто используется вместе с псевдоклассами `:active`, `:link` и `:visited`

:active

Устанавливает стили для элемента, когда ссылка становится активной, но переход по ссылке еще не совершен. Данный псевдокласс часто используется вместе с псевдоклассами `:hover`, `:link` и `:visited`

:link

Устанавливает стили для элемента, когда ссылка не является часто посещаемой. Данный псевдокласс часто используется вместе с псевдоклассами `:hover`, `:active` и `:visited`

:visited

Устанавливает стили для элемента, когда ссылка недавно посещалась. Данный псевдокласс часто используется вместе с псевдоклассами `:hover`, `:active` и `:link`. (Используйте данный псевдокласс перед ними, чтобы не подавлять их поведение)

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

cursor

Вид указателя мыши, расположенного над текущим элементом.

Значения:

- `auto` – вид указателя определяется браузером;
- `default` – вид указателя определяется операционной системой;
- `crosshair` – крестик;
- `pointer` – указующий перст;
- `move` – четырехглавая стрелка;
- `e-`, `ne-`, `nw-`, `n-`, `se-`, `sw-`, `s-`, `w-resize` – стрелки для перемещения границ. Приставки обозначены по аналогии с частями света (`sw` – юго-запад);
- `text` – текстовый указатель;
- `wait` – песочные часы;
- `help` – стандартный указатель со знаком вопроса;
- ссылка на нестандартный указатель (URL).

```
a { cursor: pointer url(giper.cur) }
```