

Основные CSS свойства

Свойство color - цвет текста

Свойство **color** позволяет задать **цвет текста**. Цвет можно задавать следующими способами: английским **словом**, через решетку **#**, через **rgb**.

Способ первый - английское слово

Самый простой и понятный вариант задать цвет - указать его английским словом, например, **red** задает *красный* цвет, **blue** - *голубой*, **green** - *зеленый*, **black** - *черный*, **white** – *белый*.

В следующем примере всем абзацам на странице задан красный цвет:

```
<p>
    Абзац с текстом.
</p>
p {
    color: red;
}
```

Способ второй - через rgb

Суть этого способа заключается в следующем - для свойства **color** пишут значение – *rgb* (*красный, зеленый, голубой*) - и указывают, в каких пропорциях нужно брать эти три базовых цвета. Сами цвета могут изменяться от **0** до **255**. Причем ноль - это отсутствие цвета, а 255 - это чистый цвет (к примеру, чисто красный).

Например, первое значение **255**, второе **255**, а третье – **ноль**. Получится чисто желтый цвет.

```
<p>
    Абзац с текстом.
</p>
p {
    color: rgb(255, 255, 0);
}
```

Чисто **белый** цвет:

```
<p>
    Абзац с текстом.
</p>
p {
    color: rgb(255, 255, 255);
}
```

Чисто **черный** цвет:

```
<p>
```

Абзац с текстом.

```
</p>
```

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

Способ третий - через

Существует и третий вариант задать цвет - через **шестнадцатеричное** значение.

В этой системе вместо **rgb()** записывается решетка **#**, после которой идут **6** знаков. Первые 2 знака - красный цвет, вторые два знака - зеленый, и последние два знака - голубой. Цвета также изменяются от **0** до **255**, но в шестнадцатеричной системе это будет от **00** до **FF**.

К примеру, запись **rgb(255, 200, 255)** можно представить как **#FFC8FF** - это будет более компактно.

В следующем примере абзац красится в красный цвет:

```
<p>
```

Абзац с текстом.

```
</p>
```

```
p {  
    color: #FF0000;  
}
```

#000000 - черный цвет, а **#FFFFFF** - белый.

Кроме того, существует сокращенная форма записи - к примеру, **#FEDFED** можно переписать как **#FED** - то есть, если первая половина записи совпадает со второй половиной - то ее (вторую половину) можно не писать. В сокращенном варианте черный цвет будет **#000**, а белый - **#FFF**.

Свойства **width** и **height** - ширина и высота

Свойства **width** и **height** позволяют задать высоту и ширину элементу соответственно. Ширина и высота обычно измеряются в **пикселях** (обозначается **px**) или **процентах** (обозначается **%**).

```
<p>
```

Абзац с текстом.

```
</p>
```

```
p {  
    width: 300px;  
    height: 100px;  
}
```

Свойство **text-align** - выравнивание текста

Свойство **text-align** позволяет задать выравнивание текста. Текст можно выравнивать по **левому** краю (значение **left**), по **правому** (значение **right**), по **центру** (значение **center**) и **одновременно** и по правому, и по левому краю (значение **justify**).

Давайте посмотрим на примерах, что имеется в виду.

Значение left

Сделаем так, чтобы текст был выровнен по **левому** краю. Для этого нужно свойство **text-align** поставить в значение **left**:

```
<p>
```

```
    Абзац с текстом.
```

```
</p>
```

```
p {
```

```
    text-align: left;
```

```
}
```

В принципе, для абзацев значение **left** можно и не ставить - они по умолчанию и так выровнены по **левому** краю. Однако есть элементы, которые по умолчанию стоят **по центру** (это, например, теги **th**, которые делают ячейку-заголовок таблицы). И вот для них как раз-таки и может потребоваться выравнивание по левой стороне.

Значение right

Теперь поставим текст по **правому** краю:

```
<p>
```

```
    Абзац с текстом.
```

```
</p>
```

```
p {
```

```
    text-align: right;
```

```
}
```

Значение center

Поставим текст **по центру**:

```
<p>
```

```
    Абзац с текстом.
```

```
</p>
```

```
p {
```

```
    text-align: center;
```

```
}
```

Значение justify

Ну, и наконец выровняем текст одновременно и по правому, и по левому краю:

```
<p>
```

```
    Абзац с текстом.
```

```
</p>
```

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

Свойство font-weight - жирность

Свойство **font-weight** позволяет сделать текст жирным или наоборот - отменить жирность (к примеру, для заголовков, которые жирные по умолчанию). Чтобы сделать текст жирным, следует добавить значение **bold**, а чтобы отменить жирность - значение **normal**.

Сделаем все абзацы жирными, а все заголовки h3 - нежирными:

```
<h3> Это заголовок</h3>  
  
<p>  
    Абзац с текстом.  
</p>  
  
h3 {  
    font-weight: normal;  
}  
  
p {  
    font-weight: bold;  
}
```

Так код будет выглядеть в браузере:

Это заголовок

Абзац с текстом.

Их вид по умолчанию:

Это заголовок

Абзац с текстом.

Свойство font-style - курсив

Свойство **font-style** позволяет сделать текст курсивным или наоборот - отменить курсив. Чтобы сделать текст курсивом, следует добавить значение **italic**, а чтобы отменить курсив - значение **normal**.

```
<p>  
    Абзац с текстом.  
</p>  
  
p {  
    font-style: italic;
```

```
}
```

Так код будет выглядеть в браузере:

Абзац с текстом.

Свойство `font-size` - размер текста

Свойство `font-size` позволяет задать размер текста. Размер задается в **пикселях** (обозначаются `px`), в **пунктах** (обозначаются `pt`), в **процентах** (обозначаются `%`).

Значение в пунктах

Пункты можно понять, если вы работали с программой Word. Там вы можете задать, к примеру, шрифт размером 16 - это число и есть размер в пунктах. При этом этот шрифт в Word будет такого же размера, как и шрифт 16 пунктов в браузере.

Между пунктами и пикселями есть соответствие: **12pt = 16px**. При этом все размеры на экране на самом деле меряются в пикселях, даже если вы задаете их в пунктах. При этом, если после перевода пунктов в пиксели, полученные пиксели будут дробными - они округлятся до ближайшего целого.

Зададим тексту абзаца шрифт в 20 пунктов:

```
<p>
```

```
    Абзац с текстом.
```

```
</p>
```

```
p {
```

```
    font-size: 20pt;
```

```
}
```

Так код будет выглядеть в браузере:

Абзац с текстом.

Свойство `font-family` - тип шрифта

Свойство `font-family` позволяет задать тип шрифта (тип часто называют *семейством* шрифта).

Поставим для всех абзацев шрифт **Arial**:

```
<p>
```

```
    Абзац с текстом.
```

```
</p>
```

```
p {
```

```
    font-family: Arial;
```

```
}
```

Так код будет выглядеть в браузере:

Абзац с текстом.

Не все шрифты можно использовать, так как если на компьютере пользователя не окажется указанного шрифта - браузер вместо него возьмет стандартный шрифт (в результате на экране будет совсем не то, что мы задумывали). Поэтому необходимо использовать только **веб безопасные шрифты**.

Также для решения данной проблемы поступают так: перечисляют несколько похожих шрифтов через запятую. Например, **font-family: Georgia, "Times New Roman"**.

Когда браузер встречается первый шрифт в списке, он проверяет его наличие на компьютере пользователя. Если такого шрифта нет, берется следующий шрифт из списка и также анализируется на присутствие. Поэтому несколько шрифтов увеличивает вероятность, что хотя бы один из них будет обнаружен на клиентском компьютере.

Заканчивают список обычно ключевым словом, которое описывает тип шрифта (все шрифты относятся к какому-нибудь типу) — **serif**, **sans-serif**, **cursive**, **fantasy** или **monospace**. Если браузер не нашел ни одного из указанных шрифтов на компьютере пользователя, то он выберет один из шрифтов указанного типа.

Шрифты бывают с засечками **serif** и без засечек **sans-serif**. **Засечки** - это специальные штрихи на концах букв (выделены красным):

AaBbCc

Засечки являются отличительной особенностью шрифта, нет такого, чтобы был одновременно с засечками и без засечек. **Arial с засечками** будет уже какой-то другой шрифт, но никак не Arial.

AaBbCc

Если в имени шрифта содержатся пробелы, например, Times New Roman, оно должно заключаться в одинарные или двойные кавычки.

Веб безопасные шрифты

Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем. Поскольку о таком положении дел остается только мечтать, то абсолютно безопасных шрифтов не существует! Отдельные шрифты можно назвать безопасными с некоторыми оговорками. Вот они:

Значение	Описание
Arial	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.
Arial Black	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.
Comic Sans MS	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.

Courier New	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.
Georgia	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.
Impact	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.
Times New Roman	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.
Trebuchet MS	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.
Verdana	Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.

Свойство font - сокращение для шрифтов

Существует специальное свойство **font**, которое можно использовать вместо многих свойств, которые мы уже разобрали. Такие свойства в CSS называются **свойствами-сокращениями**. Зачастую их использование гораздо удобнее вместо множества других свойств.

Свойство **font** имеет следующий синтаксис: **курсив жирность размер_шрифта/интервал_между_строками семейство_шрифта**. Обязательными являются "размер_шрифта" и "семейство_шрифта", порядок имеет значение.

Свойство **font** является **свойством-сокращением**.

```
<p>
```

Длинный текст...

```
</p>
```

```
p {
```

```
font: 16px Arial;
```

```
}
```

Так код будет выглядеть в браузере:

Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.

```
<p>
```

Длинный текст...

```
</p>
```

```
p {  
    font: italic 16px/50px Arial, sans-serif;  
}
```

Так код будет выглядеть в браузере:

Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.

Свойство `text-indent` - красная строка

Свойство `text-indent` позволяет задать **красную строку**, то есть отступ первой строки текста (к примеру, в абзаце).

Сделаем красную строку для абзацев:

```
<p>
```

Длинный текст...

```
</p>
```

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

Так код будет выглядеть в браузере:

Безопасным шрифтом можно назвать такой шрифт, который является стандартным для всех операционных систем.

Свойство `vertical-align`

Свойство `vertical-align` позволяет задать выравнивание по вертикали. К сожалению, адекватным образом это выравнивание работает только для таблиц.

Значение **top** выравнивает по верхнему краю, значение **bottom** - по нижнему, а **middle** - по центру по вертикали (есть еще некоторые другие значения, они нас пока не интересуют).

Значение **top**

Сделаем так, чтобы по вертикали текст в ячейках таблицы был выровнен **по верхнему краю**:

```
td {  
    vertical-align: top;  
}  
  
<table border="1" height="200">
```

```

<tr>
    <td>Ячейка 1</td>
    <td>Ячейка 2</td>
    <td>Ячейка 3</td>
</tr>
<tr>
    <td>Ячейка 4</td>
    <td>Ячейка 5</td>
    <td>Ячейка 6</td>
</tr>
<tr>
    <td>Ячейка 7</td>
    <td>Ячейка 8</td>
    <td>Ячейка 9</td>
</tr>
</table>

```

Так код будет выглядеть в браузере:

Ячейка 1	Ячейка 2	Ячейка 3
Ячейка 4	Ячейка 5	Ячейка 6
Ячейка 7	Ячейка 8	Ячейка 9

Значение bottom

А теперь по нижнему краю:

```

td {
    vertical-align: bottom;
}
<table border="1" height="200">
    <tr>
        <td>Ячейка 1</td>
        <td>Ячейка 2</td>
        <td>Ячейка 3</td>
    </tr>
    <tr>
        <td>Ячейка 4</td>

```

```
        <td>Ячейка 5</td>
        <td>Ячейка 6</td>
    </tr>
    <tr>
        <td>Ячейка 7</td>
        <td>Ячейка 8</td>
        <td>Ячейка 9</td>
    </tr>
</table>
```

Так код будет выглядеть в браузере:

Ячейка 1	Ячейка 2	Ячейка 3
Ячейка 4	Ячейка 5	Ячейка 6
Ячейка 7	Ячейка 8	Ячейка 9

Значение middle

Ну, а теперь по центру:

```
td {
    vertical-align: middle;
}
<table border="1" height="200">
    <tr>
        <td>Ячейка 1</td>
        <td>Ячейка 2</td>
        <td>Ячейка 3</td>
    </tr>
    <tr>
        <td>Ячейка 4</td>
        <td>Ячейка 5</td>
        <td>Ячейка 6</td>
    </tr>
    <tr>
        <td>Ячейка 7</td>
        <td>Ячейка 8</td>
        <td>Ячейка 9</td>
```

</tr>

</table>

Так код будет выглядеть в браузере:

Ячейка 1	Ячейка 2	Ячейка 3
Ячейка 4	Ячейка 5	Ячейка 6
Ячейка 7	Ячейка 8	Ячейка 9