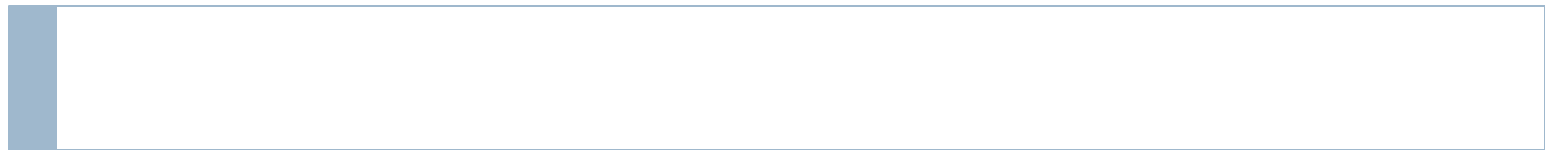


CSS 3



Применение CSS к HTML-документу

Есть три способа применить правила CSS к HTML-документу.

Метод 1: Инлайн/In-line (атрибут style)

Можно применять CSS к HTML с помощью HTML-атрибута style. Красный цвет фона можно установить так:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>This is a red page</p>
  </body>
</html>
```



Метод 2: Внутренний (тэг style)

Второй способ вставки CSS-кодов - HTML-тэг <style>. Например:

```
<html>
  <head>
    <title>Example</title>
    <style type="text/css">
      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p>This is a red page</p>
  </body>
</html>
```



Метод 3: Внешний (ссылка на таблицу стилей)

Внешняя таблица стилей это просто текстовый файл с расширением .css. Вы можете поместить таблицу стилей на ваш web-сервер или на жёсткий диск, как и другие файлы.

Например, скажем, ваша таблица стилей называется style.css и находится в папке style. Это можно проиллюстрировать так:

Весь фокус состоит в том, чтобы создать ссылку из HTML-документа на таблицу стилей (style.css).

Это можно сделать одной строчкой HTML-кода:

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



Эту строку кода нужно вставлять в разделе header HTML, то есть между тэгами <head>и </head>.

```
<html>  
  <head>  
    <title>My document</title>  
    <link rel="stylesheet" type="text/css" href="style.css" />  
  </head>  
  <body>  
    ...
```

Эта ссылка указывает браузеру, что он должен использовать правила отображения HTML-файла из CSS-файла.

Самое важное здесь то, что несколько HTML-документов могут ссылаться на одну таблицу стилей. Иначе говоря, один CSS-файл можно использовать для управления отображением множества HTML-документов.



CSS

Cascading Style Sheets

(Каскадные Документы Стилей)

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

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



Селекторы CSS

- ▣ Селекторы тэгов
 - ▣ Селектор id
 - ▣ Селектор class



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="test1">текст</p>**).
 - Для того, чтобы затем оформить данный элемент необходимо обратиться к идентификатору в таблицах стилей добавив перед ним символ "#"
(#идентификатор {color:red}).
-



пример

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



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="test1">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;  
□ }
```





Это поможет вам сэкономить уйму времени и сил. Если вы, например, хотите изменить цвет фона web-сайта из 100 страниц, таблица стилей избавит вас от необходимости вручную изменять все сто HTML-документов. Используя CSS, эти изменения можно сделать за несколько секунд, просто изменив один код в центральной таблице стилей.



Цвет и фон

Цвет переднего плана : свойство 'color'

Свойство color описывает цвет переднего плана элемента.

Например, можно сделать текст абзаца тёмно-красными. Все абзацы обозначаются HTML-элементом <p>.

```
p{  
margin:0 0 10px 0;  
font-size: 20px;  
background-size:70% 70%;  
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)).



Свойство 'background-color'

Свойство **background-color** описывает цвет фона элемента.

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

Можно также применять это свойство к другим элементам, в том числе - к заголовкам и тексту.

```
body {  
background-color: #FFCC66;  
  
opacity:0.5;  
width:100%;  
height:100%;  
background-image: url("1234.gif");  
}
```



Повторение/мультипликация фонового изображения [background-repeat]

Свойство **background-repeat** управляет этим.
В таблице указаны четыре значения background-repeat.

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

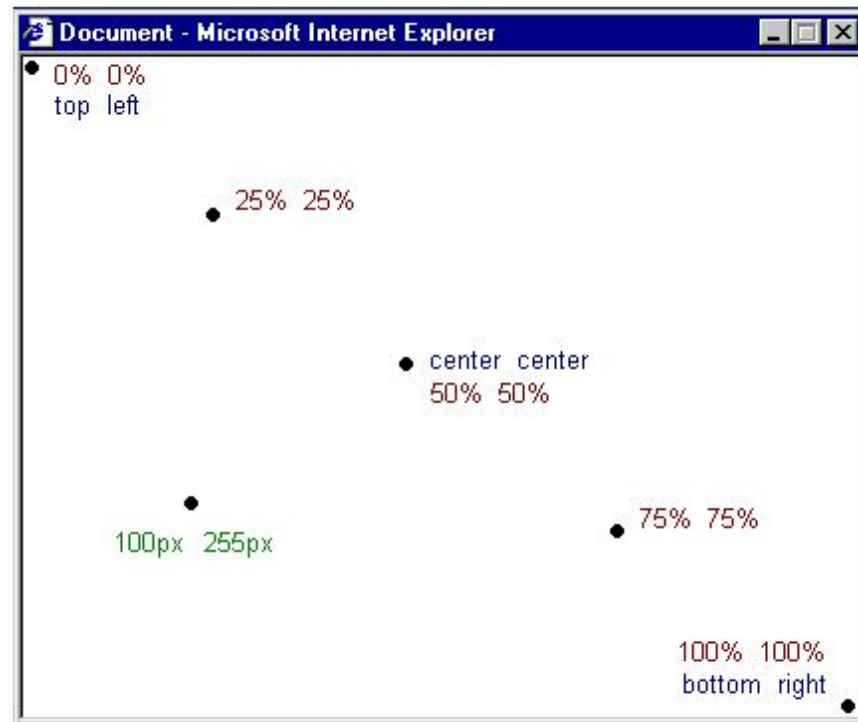


Расположение фонового рисунка [background-position]

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 (пиксели, сантиметры, и т. п.), либо вы можете использовать слова **top**, **bottom**, **center**, **left** и **right**. Модель иллюстрирует сказанное:



background-image

- Устанавливает фоновое изображение для элемента.
- **Линейный градиент**

background: linear-gradient(позицияto top right, начальный цвет#000, конечный цвет#fff);

- **background-image: linear-gradient(top, #FF4444, #009999);**



Позиция	Угол	Описание
to top	0deg	Снизу вверх.
to left	270deg	Справа налево.
to bottom	180deg	Сверху вниз.
to right	90deg	Слева направо.
to top left		От правого нижнего угла к левому верхнему.
to top right		От левого нижнего угла к правому верхнему.
to bottom left		От правого верхнего угла к левому нижнему.
to bottom right		От левого верхнего угла к правому нижнему.



Радиальный градиент

- Радиальный градиент создаётся с помощью свойства `background` или `background-image` с параметром `radial-gradient`. В простейшем случае для задания радиального градиента понадобится всего два параметра: начальный и конечный цвет.

- **`height: 100px;`**
- **`width: 100px;`**
- **`border-radius: 50px;`**
- **`Background-image: radial-gradient(#55ddff, #0081b5);`**



-
- Начальную точку градиента можно задавать в любом месте элемента, для этого вначале указывается её позиция.

background: radial-gradient(позиция at top left, конечный цвет #000, начальный цвет #fff);



background-image

- Один элемент может иметь несколько фоновых изображений одновременно.

Для этого необходимо перечислить путь к ним через запятую в свойстве `background-image`.

1. Картинки будут накладываться друг на друга в перечисленной очередности (т.е. первая заданная картинка будет отображаться поверх последующих).
2. Для того, чтобы применить к фоновым картинкам свойства оформления необходимо перечислить необходимые значения в нужном порядке через запятую (к примеру в свойстве `"background-position:bottom right, center;"` `bottom right` будет применено к первой, а `center` ко второй фоновой картинке).



**□ background-image:url(wislink.gif),url(mountimg3.jpg); background-position:bottom right, center;
background-size:150px 40px,100% 100%;
background-repeat:no-repeat,no-repeat;**



Сокращённая запись [background]

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

Например, посмотрите на эти строки:

```
background-color: #FFCC66;  
background-image: url("1234.gif");  
background-repeat: no-repeat;  
background-position: right bottom;
```

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

```
background: #FFCC66 url("1234.gif") no-repeat right  
bottom;
```



Ссылки

Что такое псевдокласс?

Псевдокласс позволяет учитывать различные условия или события при определении свойств HTML-тэга.

Рассмотрим пример. Как вы знаете, ссылки специфицируются в HTML тэгом `<a>`. В CSS мы также можем использовать `a` в качестве селектора:

```
a {  
  color: blue;  
}
```

Ссылка может иметь разные состояния. Например, её уже посетили/`visited` или ещё нет. Можно использовать псевдоклассы для установки разных стилей посещённых и непосещённых ссылок.

```
a:link {  
  color: blue;  
}
```

```
a:visited {  
  color: red;  
}
```

Используйте `|a:link|` и `|a:visited|` для непосещённых и посещённых ссылок, соответственно. Активные ссылки имеют псевдокласс `a:active`, и `a:hover`, когда указатель - над ссылкой.



Псевдокласс: link

Псевдокласс :link используется для ссылок на страницы, которые пользователь ещё не посещал.

В примере кода непосещённые ссылки - синие.

```
a:link {  
color: #6699CC;  
}
```

Непосещённые ссылки - светло-синие

[Это - непосещённая ссылка](#)



Псевдокласс: visited

Псевдокласс `:visited` используется для ссылок на страницы, которые пользователь посетил. В примере кода посещённые ссылки - фиолетовые.

```
a:visited {  
  color: #660099;  
}
```

Посещённые ссылки - тёмно-фиолетовые

[Это - посещённая ссылка](#)



Псевдокласс: active

Псевдокласс :active используется для активных ссылок.

В примере активные ссылки имеют жёлтый фон.

```
a:active {  
background-color: #FFFF00;  
}
```

Активные ссылки имеют жёлтый фон

[Нажмите здесь и держите клавишу нажатой](#)



Псевдокласс: `hover`

Псевдокласс `:hover` используется для ссылок, над которыми находится указатель мыши.

Это можно использовать для создания интересных эффектов. Например, если мы хотим, чтобы ссылки становились оранжевыми и курсивными при прохождении указателя над ними, то наш CSS должен выглядеть так:

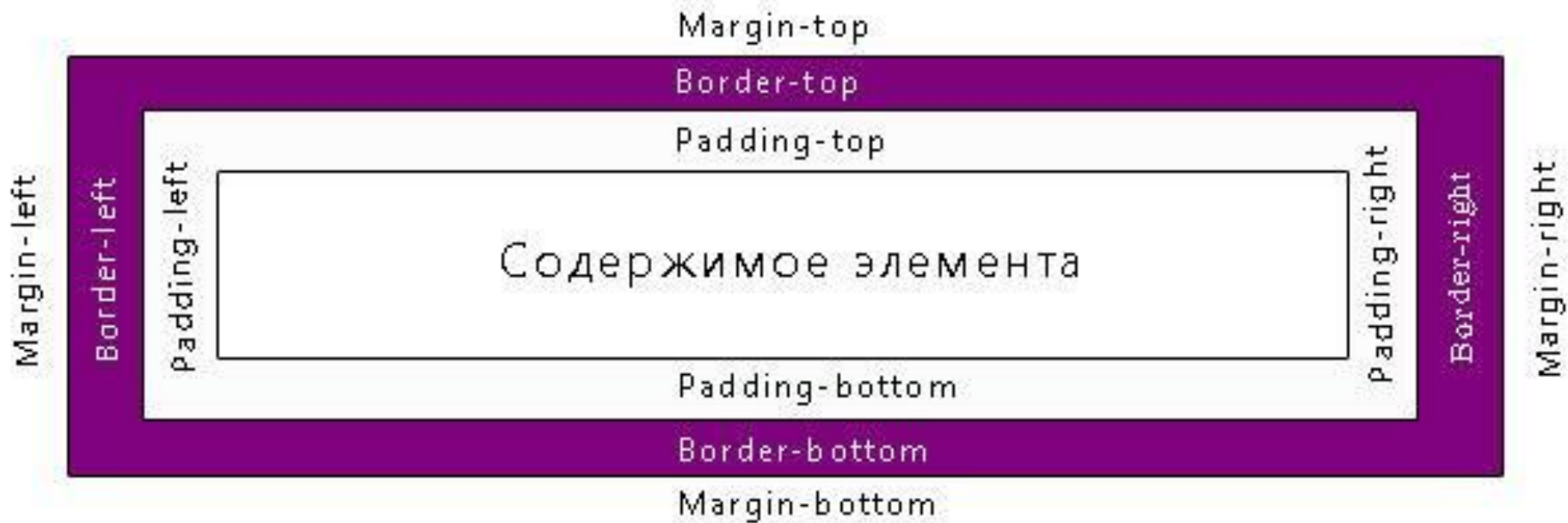
```
a:hover {  
  color: orange;  
  font-style: italic;  
}
```


Hover-эффект ссылки!

Проведите указателем мыши над этой ссылкой



Блочная модель



-
- Все элементы в CSS являются прямоугольными блоками. Каждый такой блок имеет зону **content**, в которой располагается содержимое элемента (т.е. текст, изображения и т.д.). Вокруг зоны content могут располагаться необязательные зоны: padding, border и margin.
 - Свойства width и height устанавливают ширину и высоту только блока content, а не элемента целиком.
 - Итоговый размер элемента помимо размеров content будет включать в себя еще и размеры padding, border и margin
-
- 

```
□ .ex2 {  
□   border:5px brown solid;  
□   margin-top:30px;  
□   margin-left:250px;  
□   margin-right:70px;  
□   padding: 5px;  
□ }
```




примеры

- p{
- background-color: yellow;
- margin: 0;
- padding: 0;
- }



Block header

- p{
- background-color: yellow;
- margin: 0;
- padding: 20px;
- }

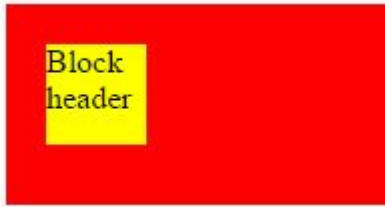


Block header

padding раздвигает блок, увеличивает его ширину и высоту.



-
- margin: 20px;
 - padding: 0;



- margin: -20px;
- padding: 0;



Проценты у padding и margin

- значение высчитывается из ширины родителя, а не от его высоты



box-sizing

- Свойство `box-sizing` позволяет чтобы свойства `width` и `height` задавали размеры не контента, а размеры блока.
 - **Значения**
 - `content-box` Свойства `width` и `height` задают ширину и высоту контента и не включают в себя значения отступов, полей и границ.
 - `border-box` Свойства `width` и `height` включают в себя значения полей и границ, но не отступов (`margin`).
 - `padding-box` Свойства `width` и `height` включают в себя значения полей, но не отступов (`margin`) и границ (`border`).
-
- ▶ `inherit` Наследует значение родителя.

границы

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



- **.bor2 {**
- **border-style:solid;**
- **border-color:green;**
- **border-width:2px;**
- **}**

- Рассмотренные ранее свойства могут также применяться к отдельным сторонам границы
 - **border-left-style:solid;**
 - **border-right-style:solid;**
-



Сокращенная запись

- Стенографическая форма записи объединяет все свойства оформления границ в одном свойстве **border**.
- Порядок следования свойств:
- `border-width`
- `border-style`
- `border-color`
- можно пропускать неиспользуемые свойства.
- **`border:2px solid green;`**



Создание элементов со сглаженными углами

- С помощью **border-radius** можно делать углы элементов сглаженными
 - **border-radius: 5px;**
 - **border-top-left-radius** делает сглаженным только верхний левый угол элемента;
 - **border-top-right-radius** делает сглаженным только верхний правый угол элемента;
 - **border-bottom-left-radius** делает сглаженным только нижний левый угол элемента;
 - **border-bottom-right-radius** делает сглаженным только нижний правый угол элемента.
-



Вставка изображений в качестве границы

- ▣ **border-image** позволяет вставлять произвольные изображения в качестве границы элемента.
- ▣ **border-image:путь(1) отступ(2) ширина(3) повторение(4);**
- ▣ **border-image:url("images.jpg") 30 30 round;**



stretch	Растягивает рисунок границы до размеров элемента. Это значение используется по умолчанию.
repeat	Повторяет рисунок границы.
round	Повторяет рисунок и масштабирует его так, чтобы на стороне элемента оказалось целое число изображений



position

- Устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице.
- **position: absolute | fixed | relative | static | inherit**



absolute	Указывает, что элемент абсолютно позиционирован. Положение элемента задается свойствами left, top, right и bottom. Если у родителя значение position установлено как static или родителя нет, то отсчет координат ведется от края окна браузера. Если у родителя значение position задано как fixed, relative или absolute, то отсчет координат ведется от края родительского элемента.
fixed	Привязывается к указанной свойствами left, top, right и bottom точке на экране и не меняет своего положения при прокрутке веб-страницы.
relative	Положение элемента устанавливается относительно его исходного места. Добавление свойств left, top, right и bottom изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения.
static	Элементы отображаются как обычно. Использование свойств left, top, right и bottom не приводит к каким-либо результатам.
inherit	Наследует значение родителя.



Единицы измерения

Абсолютные единицы

- К абсолютным единицам измерения относятся сантиметры (cm), миллиметры (mm) и дюймы (in).
- дюймы (in)
- сантиметры (cm)
- миллиметры (mm)
- пункты (pt)
- пики (pc)

$$1 \text{ in} = 96 \text{ px}$$

В спецификации css $1 \text{ pt} = 1/72 \text{ in}$, а $1 \text{ pc} = 12 \text{ pt}$.

В свойстве font-size задание отрицательного значения в единицах длины, например **-25cm**, **недопустимо**.



Относительные единицы

- em (кегельная)
 - x-height (ex)
 - px (пикселы)

 - em — это размер шрифта, который определяется высотой заглавной буквы и размеры выносных элементов сверху и снизу(вверху могут появиться диакритические знаки, например: ё или й).
 - ex – высота строчной буквы. ex в разных браузерах может быть разным
 - Пиксель — это мельчайшая точка, которую можно установить на экране компьютера.
-



Проценты (%)

- % можно писать не у всех правил, а только у некоторых и нужно понимать от чего они считаются.



display

- Многоцелевое свойство, которое определяет, как элемент должен быть показан в документе



Значение	Описание
block	Элемент показывается как блочный. Применение этого значения для встроенных элементов, например тега <code></code> , заставляет его вести подобно блокам — происходит перенос строк в начале и в конце содержимого.
inline	Элемент отображается как встроенный. Использование блочных тегов, таких как <code><div></code> и <code><p></code> , автоматически создает перенос и показывает содержимое этих тегов с новой строки. Значение <code>inline</code> отменяет эту особенность, поэтому содержимое блочных элементов начинается с того места, где окончился предыдущий элемент.
inline-block	Это значение генерирует блочный элемент, который обтекается другими элементами веб-страницы подобно встроенному элементу. Фактически такой элемент по своему действию похож на встраиваемые элементы (вроде тега <code></code>). При этом его внутренняя часть форматируется как блочный элемент, а сам элемент — как встроенный.



display: block

- По умолчанию присвоен p, div, form, ul, h1 и т.д.
- на строку, на которой расположился блок, уже нельзя добавить другой элемент, даже когда остаётся пустое пространство. Искл., при применении свойств position, float, transform.
- если не задано значение width, элемент растягивается на весь родительский контейнер
- если задано значение width, ширина складывается из margin, border, padding, width ([подробнее](#)). Поэтому красная рамка выходит за пределы родительского блока при width: 100%.
- ...



display: inline

- По умолчанию присвоен a, span, b, em и т.д.
- элементы следуют друг за другом.
- width и height игнорируются. Его ширина - это ширина содержимого плюс margin, border и padding.
- margin-top и margin-bottom игнорируются.
- совместно с padding-top и padding-bottom следует корректировать line-height, иначе строки будут "наезжать" друг на друга.



display: inline-block

- По умолчанию присвоен `img`, `input` и т.д. По отношению к внешним элементам ведёт себя как `inline`, а внутренним `block`.
- элементы следуют друг за другом.
- имеет `width` и `height`, `margin-top` и `margin-bottom`.
- если не задано значение `width`, растягивается по ширине самого длинного элемента внутри.



примеры

- `<header>блок</header>`
- `<div class="layout">`
- `<div id="col1">блок 1</div>`
- `<div id="col2">блок 2</div>`
- `<div id="col3">блок 3</div>`
- `</div>`



У колонок с заданной шириной стоит свойство `width`, а их положение слева или справа задаётся соответственно свойством `left` или `right`. Резиновая ширина оставшейся колонки строится после одновременного добавления `left` и `right`, значения которых совпадают с шириной фиксированных колонок.

- ❑ `.layout { position: relative; width: 80%; height: 20%;}`
- ❑ `div { position: absolute; height: 40px;}`
- ❑ `#col1 { background: #00FF00; left: 0; right: 300px; }`
- ❑ `#col2 { background: #0000FF; width: 200px; right: 100px; }`
- ❑ `#col3 { background: #FF0000; width: 100px; right: 0; }`



ширина всех колонок задана в процентах

- #col1 { background: #00FF00; width: 33%; }
- #col2 { background: #0000FF; left: 33%; width: 34%; }
- #col3 { background: #FF0000; right: 0; width: 33%; }



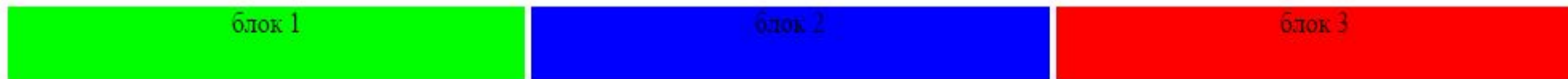
ДВЕ КОЛОНКИ ИЗ ТРЁХ РЕЗИНОВЫЕ

- #col1 {background: #00FF00; left: 0; right: 200px; margin-right: 50%; }
- #col2 { background: #0000FF; width: 50%; right: 200px; }
- #col3 { background: #FF0000; width: 200px; right: 0; }



Вычисляемое значение

- `display: inline-block;`
- `margin: 0 auto;`
- `width: calc(25% - 10px);`



Шрифты

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: имя семейства/family-name и общее/родовое семейство/generic family.

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/feet".



Times New Roman
Garamond
Georgia

Эти три шрифта
принадлежат к общему
семейству serif. У них
имеются т. н. "засечки".

Trebuchet
Arial
Verdana

Эти три шрифта
принадлежат к общему
семейству sans-serif.
У них нет "засечек".

Courier
Courier New
Andale Mono

Эти три шрифта
принадлежат к общему
семейству monospace.
Символы этих шрифтов
имеют одинаковую
ширину (т. н.
"моноширинные шрифты").



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;}
```

Заголовок 1 выведен шрифтом Arial

А заголовок 2 - Times New Roman

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;}
```

Заголовок 1 выведен шрифтом Arial

А заголовок 2 - Times New Roman - italic



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.

Sans Book SC

ABCabc

Sans Bold SC

ABCabc

Serif Book SC

ABCabc

Serif Bold SC

ABCabc

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;}
```

ЗАГОЛОВОК 1 - SMALL CAPS

А заголовок 2 - normal



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;}
```

Текст в ячейках - жирным шрифтом

Здесь обычный текст



Размер шрифта [font-size]

Размер шрифта устанавливается свойством **font-size**.

Используются различные единицы измерения (например, пиксели и проценты) для описания размера шрифта.

Вот примеры:

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

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

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

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

**Заголовок <h1> размер 30px**

**Заголовок <h2> размер 1cm**

**Заголовок <h3> размер 120%**

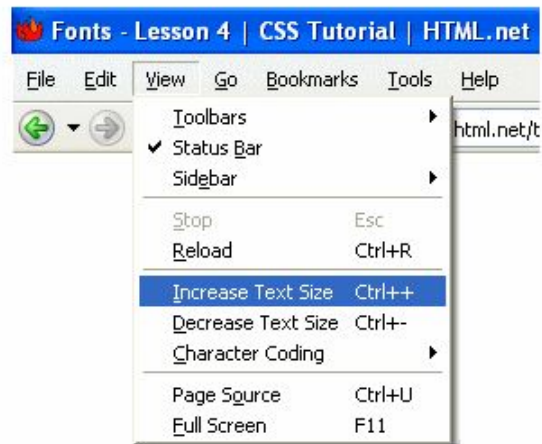
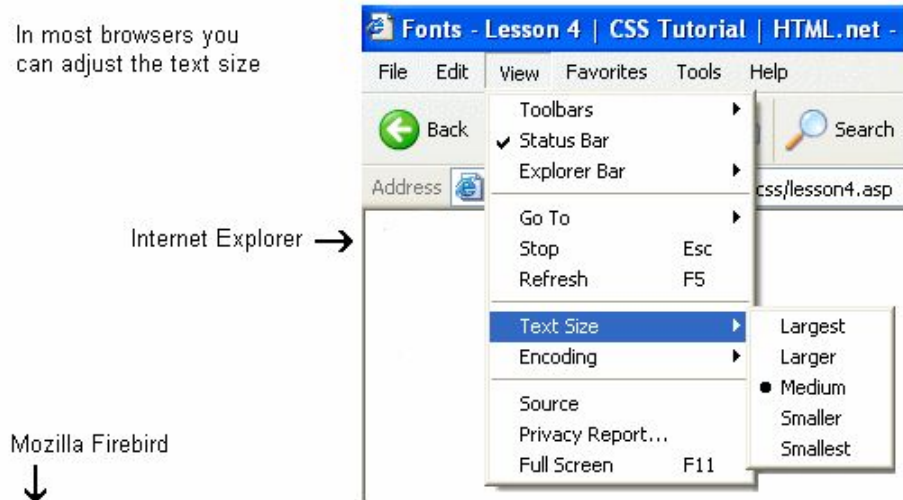
Параграф <p> размер 1em



Есть одно отличие в указанных единицах измерения: 'px' и 'pt' дают абсолютное значение размера шрифта, а '%' и 'em' - относительные. Многие пользователи не могут читать мелкий текст, по разным причинам. Чтобы сделать ваш web-сайт доступным для всех, вы должны использовать относительные значения, такие как '%' или 'em'.

Вот иллюстрация того, как настроить размер шрифта в Mozilla Firefox и Internet Explorer. Попробуйте сами - прекрасное свойство, как вы полагаете?

In most browsers you can adjust the text size



## Сокращённая запись [font]

Используя сокращённую 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 таков:

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



# Текст

## Форматирование и установка стиля текста - ключевая проблема для любого web-дизайнера

### 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;
}
```

### Отступ

Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antiquae, si peraque dure dicere credit eos, ignave multa fatetur, et sapit et mecum facit et Iova iudicat aequo. Non equidem insector delendave carmina Livi esse reor, memini quae plagosum mihi parvo Orbilium dictare, sed emendata videri pulchraque et exactis minimum distantia miror. Inter quae verbum emicuit si forte decorum, et si versus paulo concinnior unus et alter, venditque poema.

Indignor quicumque reprehendi, non quia crasse compositum illepedeve putetur, sed quia nuper, nec veniam antiquis, sed honorem et praemia posci. Recte necne crocum floresque perambulet Attae fabula si dubitem, clament perisse pudorem cuncti paene patres, ea cum reprehendere coner, quae gravis Aesopus, quae doctus Roscius egit, vel quia nil rectum, nisi quod placuit sibi, ducunt, vel quia turpe putant parere minoribus, et quae imberbes senes.

Quod si tam Graecis novitas invisae fuisset quam nobis, quid nunc esset vetus? Aut quid haberet quod legeret tereretque viritum. Ut primum positus mugari Graecia bellis coepit et in vitium fortuna labier aequa, nunc athletarum studiis, nunc arsit equorum, marmoris aut eboris fabros aut aeris amavit, tibicinibus, nunc est gavisa tragoedia; puella.

Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes, nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem aut ambulemus voluntatis esse, is non videt quae quamque rem res consequatur. Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes, nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem aut ambulemus voluntatis esse, is non videt quae quamque rem res consequatur. Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes, nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem.

Brevi vel toto est iunior anno. Utor permissio, caudaeque pilos ut equinae paulatim vello unum, demo etiam unum. Si meliora dies, ut vina, poemata reddidit, scire velim, chartis perficit quotus pretium quotus arroget annus. Scriptor abhinc reddidit misso annos centum qui decidit, inter perfectos veteresque referri debet an inter vilis atque perfectos novos? Excludat iurgia finis. `»Est vetus atque probus, centum qui perficit annos. Quid, qui deperit nihil perfectos uno mense vel? »Iste quidem veteres inter ponetur honeste, qui vel mense brevi vel toto est iunior anno. Utor permissio, caudaeque nisi pilos ut equinae paulatim vello et virtutem, demo etiam unum, dum cadat elusus ratione ruentis acervi, qui reddit in fastos et virtutem aestimat annis miraturque nihil nisi quod Ennius et sapines et fortis et alter Homerus, ut critici dicunt, leviter curare videtur, quo promissa cadant et somnia Pythagorea. Naevius in manibus non est et sanctum mentibus haeret paene recens? Adeo sanctum est vetus omne poema. Ambigitur quotiens, uter utro sit prior, aufert Pacuvius docti famam senis Accius alti, dicitur Afrani toga convenisse Menandro, Plautus. Hos ediscit et hos arto stipata theatro spectat. Roma potens; habet hos nisi numeratque poetas ad ambigitur tempus Livi scriptoris ab aevo.`

Ennius et sapines et fortis et alter Homerus, ut critici dicunt, leviter curare videtur, quo promissa cadant et somnia Pythagorea. Naevius in manibus non est et mentibus haeret paene recens? Adeo sanctum est vetus omne poema. Ambigitur quotiens, uter utro sit prior, aufert Pacuvius docti famam senis Accius alti, dicitur Afrani toga convenisse Menandro, Plautus ad exemplar Siculi proprore Epicharmi, vincere Caecilium gravitate, Terentium arte. Hos ediscit et hos arto stipata theatro spectat. Roma potens; habet hos numeratque poetas ad nostrum tempus Livi scriptoris ab aevo et mentibus haeret paene recens. Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antiquae, si peraque dure dicere credit eos, ignave multa fatetur, et sapit et mecum facit et Iova iudicat aequo. Non equidem insector delendave carmina Livi esse reor, memini quae plagosum mihi parvo Orbilium dictare, sed emendata videri pulchraque et exactis minimum distantia miror. Inter quae verbum emicuit si forte habet hos nisi numeratque poetas.



## Выравнивание текста [text-align]

CSS-свойство `text-align` соответствует атрибуту, используемому в старых версиях HTML. Текст может быть выровнен `left`, `right`, `center` 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;
}
```

---



## Выравнивание текста

### Выравнивание текста в таблице

| Заголовок 1 | Заголовок 2 |
|-------------|-------------|
| Ячейка 1    | Ячейка 2    |
| Ячейка 3    | Ячейка 4    |

### Выравнивание по формату (justify) в параграфах

Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique, si peraque dure dicere credit eos, ignave multa fatetur, et sapit et mecum facit et Iova iudicat aequo. Non equidem insector delendave carmina Livi esse reor, memini quae plagosum mihi parvo Orbilium dictare; sed emendata videri pulchraque et exactis minimum distantia miror. Inter quae verbum emicuit si forte decorum, et si versus paulo concinnior unus et alter, venditque poema.

Indignor quicquam reprehendi, non quia crasse compositum illepedeve putetur, sed quia nuper, nec veniam antiquis, sed honorem et praemia posci. Recte necne crocum floresque perambulet Attae fabula si dubitem, clament periisse pudorem cuncti paene patres, ea cum reprehendere coner, quae gravis Aesopus, quae doctus Roscius egit; vel quia nil rectum, nisi quod placuit sibi, ducunt, vel quia turpe putant parere minoribus, et quae imberbes senes.

Quod si tam Graecis novitas invisae fuisset quam nobis, quid nunc esset vetus? Aut quid haberet quod legeret tereretque viritum. Ut primum positus nugari Graecia bellis coepit et in vitium fortuna labier aequa, nunc athletarum studiis, nunc arsit equorum, marmoris aut eboris fabros aut aeris amavit, tibicinibus, nunc est gavisae tragoedis; puella.

Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes; nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem aut ambulemus voluntatis esse, is non videt quae quamque rem res consequatur. Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes; nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem aut ambulemus voluntatis esse, is non videt quae quamque rem res consequatur. Haec disserens qua de re agatur et in quo causa consistat non videt. Non enim si alii ad alia propensiores sunt propter causas naturales et antecedentes, idcirco etiam nostrarum voluntatum atque appetitionum sunt causae naturales at antecedentes; nam nihil esset in nostra potestate si res ita se haberet. Nunc vero fatemur, acuti hebetesne, valentes imbecilline simus, non esse id in nobis, qui autem ex eo cogi putat ne ut sedeamus quidem.



## Декоративный вариант [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;
}
```

**Этот текст подчёркнут**

```
h2 {
text-decoration: overline;
}
```

**Этот текст надчёркнут**

```
h3 {
text-decoration: line-through;
}
```

**~~Этот текст перечёркнут~~**



## Интервал между буквами [letter-spacing]

Интервал между буквами текста можно специфицировать свойством `letter-spacing`. Значение - нужная величина. Например, если вам необходимо 3px между буквами в параграфах `<p>` и 6px - в заголовках `<h1>`, то используется такой код:

```
h1 {
 letter-spacing: 6px;
}
```

```
p {
 letter-spacing: 3px;
}
```

### Межсимвольный интервал

*Interdum volgus rectum videt, est ubi peccat. Si veteres ita miratur laudatque poetas, ut nihil anteferat, nihil illis comparet, errat. Si quaedam nimis antique, si peraque dure dicere credit eos, ignave multa fatetur, et sapit et mecum facit et Iova iudicat aequo. Non equidem insector delendave carmina Livi esse reor, memini quae plagosum mihi parvo Orbilius dictare; sed emendata videri pulchraque et exactis minimum distantia miror. Inter quae verbum emicuit si forte decorum, et si versus paulo concinnior unus et alter, venditque poema.*

*Indignor quicumque reprehendi, non quia crasse compositum illepedeve putetur, sed quia nuper, nec veniam antiquis, sed honorem et praemia posci. Recte necne crocum floresque perambulet Attae fabula si dubitem, clament periisse pudorem cuncti paene patres, ea cum reprehendere coner, quae gravis Aesopus, quae doctus Roscius egit; vel quia nil rectum, nisi quod placuit sibi, ducunt, vel quia turpe putant parere minoribus, et quae imberbes senes.*

*Quod si tam Graecis novitas invisae fuisset quam nobis, quid nunc esset vetus? Aut quid haberet quod legeret tereretque viritim. Ut primum positus nugari Graecia bellis coepit et in vitium fortuna labier aequa, nunc athletarum studiis, nunc arsis equorum, marmoris aut eboris fabros aut aeris amavit, tibicinibus, nunc est gavisae tragoedis; puella.*





## Трансформация текста [text-transform]

Свойство **text-transform** управляет регистром символов. 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).



отобразим все заголовки верхним регистром.

```
h1 {
text-transform: uppercase;
}
```

```
li {
text-transform: capitalize;
}
```

## **ЭТОТ ЗАГОЛОВОК - В ВЕРХНЕМ РЕГИСТРЕ**

- Петер Хансон
- Макс Ларсон
- Джо Доу
- Пола Джоунз
- Моника Левински
- Доналд Дак

Обратите внимание, как мы, с помощью CSS, капитализируем все имена

