

Свойства CSS

CSS позволяет манипулировать следующими свойствами элементов:

Свойства шрифта

- font-family - определяет используемый элементом шрифт. Если указать URL(file), то шрифт автоматически установится на компьютер пользователя;
- font-style - стиль шрифта (normal, italic);
- font-variant - варианты отображения шрифта (normal, small-caps);
- font-weight - жирность шрифта (normal, bold, bolder, lighter, значение от 100 до 900);
- font-size - размер шрифта (размер, xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger);
- font - обобщает вышеперечисленные свойства (любая комбинация из вышеперечисленных значений);

Свойства текста

- word-spacing - расстояние между словами (значение, normal);
- text-decoration - декорация текста (none, underline, overline, line-through, blink);
- letter-spacing - расстояние между буквами (значение, normal);
- vertical-align - позиционирование по отношению к другим элементам стоящим в одном ряду (baseline, sub, super, top-text, top, middle, bottom, bottom-text, %);
- text-transform - изменение текста (none, Capitalize, UPPERCASE, lowercase);
- text-align - положение текста (left, right, center, justify);
- text-indent - отступ (значение, %);
- line-height - отступ сверху (normal, значение, %);

Свойства фон и цвет

- color - цвет элемента (значение);
- background-color - цвет фона элемента (значение);
- background-image - изображение фон (none, URL);
- background-repeat - варианты повторения фонового изображения (repeat, repeat-x, repeat-y, no-repeat);
- background-attachment - возможность прокрутки фонового изображения (scroll, fixed);
- background-position - положение фонового изображения (%ширины%высоты, top, middle, bottom, left, center, right);
- background - обобщает вышеперечисленные свойства (любая комбинация из вышеперечисленных значений);

Свойства блока

- margin-top - определяет отступ сверху (значение, %, auto);
- margin-right - определяет отступ справа (значение, %, auto);
- margin-bottom - определяет отступ снизу (значение, %, auto);
- margin-left - определяет отступ слева (значение, %, auto);
- margin - обобщает все вышеперечисленные свойства;
- padding-top - отступ от верхнего border'a (значение, %);
- padding-right - отступ от правого border'a (значение, %);
- padding-bottom - отступ от нижнего border'a (значение, %);
- padding-left - отступ от левого border'a (значение, %);
- padding - обобщает все вышеперечисленные свойства;
- border-top-width - толщина верхнего border'a (значение, thin, medium, thick);
- border-right-width - толщина правого border'a (значение, thin, medium, thick);
- border-bottom-width - толщина нижнего border'a (значение, thin, medium, thick);
- border-left-width - толщина левого border'a (значение, thin, medium, thick);
- border-width - обобщает все вышеперечисленные свойства;
- border-color - Цвет border'a. (значение);

- border-style - стиль border'ов (none, dotted, dashed, solid, double, groove, ridge, inset, outset);
- border-top - обобщает вышеперечисленные свойства для верхнего border'a;
- border-right - обобщает вышеперечисленные свойства для правого border'a;
- border-left - обобщает вышеперечисленные свойства для левого border'a;
- border-bottom - обобщает вышеперечисленные свойства для нижнего border'a;
- border - обобщает все вышеперечисленные свойства;
- width - ширина элемента (значение, %);
- height - высота элемента (значение, %);
- float - расположение элемента (left, right, none);
- clear - расположение других элементов вокруг данного (left, right, both, none);

Классификационные свойства

- display - определяет, как будет отображаться элемент (none, block, inline, list-item);
- white-space - определяет, как будут отображаться пробелы между элементами (normal, pre, nowrap);
- list-style-type - определяет вид list-item маркера в списках (disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none);
- list-style-image - задает вид list-item маркера из картинки (none, URL);
- list-style-position - определяет положение маркера в зависимости от list item элемента (inside, outside);
- list-style - обобщает вышеперечисленные свойства;

Свойства элемента

- position - определяет, как будет отображаться элемент по отношению к другим элементам документа (relative, absolute);
- top - определяет позицию элемента TOP относительно элемента родителя (значение, %);
- left - определяет позицию элемента LEFT относительно элемента родителя (значение, %);
- width - определяет ширину элемента (значение, %, auto);
- height - определяет высоту элемента (значение, %, auto);
- overflow - режим отображения содержимого элемента, при несоответствии размера элемента, размеру содержимого (none, clip, scroll);
- visibility - управление видимостью элемента в документе (hidden, " ");

Правила CSS

Лист стилей представляет собой набор правил CSS. Правила задаются в форме простых и групповых селекторов, ID селекторов, правил наследования и правил определения классов элементов, псевдоклассов и псевдоэлементов.

Селекторы

Простейшее правило CSS задается следующим образом:

Селектор { свойство CSS: значение }

Селектор это любой из рассмотренных нами ранее тегов HTML (например BODY, P, H1, LI). Далее в фигурных скобках декларируется значение свойств CSS определяющих стиль данного элемента в документе. Например:

H1 { color:red } - задает стиль заголовкам первого уровня.

Для удобства применения можно декларировать в одном правиле несколько свойств CSS для нескольких селекторов. Например:

BODY { background-color:white; color:black; font-family:Times New Roman; font-style:normal; font-size:10pt } - данное правило задает стиль всему телу документа (вместо атрибутов тега BODY).

H1, H2, H3, H4, H5, H6 { color:black; font-style:italic } - данное правило задает стиль заголовкам всех уровней в документе.

Правила наследования

При задании правил можно использовать принцип наследования, который задается при помощи контекстных селекторов, стиль которых зависит от вложенности в другие элементы. Контекстные селекторы задаются без разделительных запятых.

Основной_селектор Вложенный_селектор { свойство CSS: значение }

Например:

P STRONG { color:red; font-weight:bold } - задает стиль для элементов STRONG заключенных в контейнере абзаца.

Кроме этого следует помнить о внутреннем наследовании стилей в документе. Например стиль элемента BODY (в части тех свойств CSS, значения которых не переопределены в персональных правилах вложенных элементов) распространяется на все дочерние элементы и без их перечисления в контекстных селекторах.

Классы элементов

В правилах CSS используется понятие класса. Класс элемента задается следующим образом:

Селектор.Класс { свойство CSS: значение }

Например:

H1.normal { color:black; font-style:normal; font-family:serif } - класс строгого стиля заголовка;

H1.funny { color:blue;font-style:italic; font-family:fantasy } - класс легкомысленного стиля заголовка;

Для использования в HTML документе элемента конкретного класса применяется атрибут CLASS:

```
<H1 CLASS=normal>Технические параметры устройства DJP67575</H1>
```

```
<H1 CLASS=funny>Свежие анекдоты про программистов</H1>
```

В CSS можно определять классы не связанные с конкретными элементами:

.Класс { свойство CSS: значение }

Например:

.red { color:red } - класс красного стиля элементов;

В HTML документе использование данного стиля выглядит так:

```
<p>Ничего не получается сказал он и набрал команду <kbd class=red>FORMAT C:</kbd></p>
```

Псевдоклассы и псевдоэлементы

В CSS определена особая группа классов и элементов автоматически поддерживаемых CSS совместимыми браузерами.

Селектор:псевдокласс { свойство CSS: значение }

Селектор:псевдоэлемент { свойство CSS: значение }

Например вот так можно задать стили для псевдоклассов элемента **A**:

a:link { color:blue; text-decoration:none } - стиль непосещенных ссылок;

a:active { color:red; text-decoration:underline } - стиль активизированных ссылок;

a:visited { color:navy; text-decoration:none } - стиль посещенных ссылок;

a:hover { color:red; text-decoration:underline } - стиль ссылок над которыми находится курсор-указатель;

a:focus { background-color:silver } - стиль ссылок получивших фокус ввода;

Манипулируя свойствами псевдоклассов можно задать разные стиль для выделенных посещенных и выделенных непосещенных ссылок:

a:focus:visited { background-color:silver } - посещенные;

a:focus:link { background-color:grey } - непосещенные;

А вот так можно задать стиль псевдоэлементу **first-letter** элемента **P**:

p:first-letter { color:red; font-weight:bold } - стиль для первой буквы абзаца;

Как использовать стили?

Браузер использует css файл для форматирования документа. Существует три способа вставки стилевой информации в html-файл:

Внешний файл css

Идеально подходит для применения стиля ко многим страницам. Использование внешнего файла css позволяет менять оформление всего сайта, изменяя всего один файл. На каждой странице сайта надо вставить ссылку на внешний файл стиля, используя тег **<link>**. Тег **<link>** надо вставлять в раздел **<head>** документа.

```
<head>
```

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

Внутренний стиль

Внутренний стиль применяется в том случае, если вам надо изменить оформление только на одной странице. Это делается с помощью тега **<style>**, который надо поместить в раздел **<head>** документа.

```
<head>
```

```
<style type="text/css">
```

```
body {background-color: red}
```

```
p {margin-left: 20px}
```

```
</style>
```

```
</head>
```

Строчный стиль

Строчный стиль применяется для уникализации стиля конкретного элемента.

Для использования строчного стиля вы помещаете его в соответствующий тег. **Атрибут тега style** может содержать любое свойство CSS. Этот пример демонстрирует как изменить цвет и отступ слева параграфа:

```
<p style="color: red; margin-left: 20px">
```

Это параграф.

```
</p>
```

Чтобы узнать больше о CSS, посетите наш [учебник CSS](#).