



- **Ищу рекламные места. Куплю интернет трафик дорого**
- **Покупаю ссылки с сайтов. Обращайтесь**
- **Как зарабатывать до \$100 в сутки в интернете?**

Что есть CSS

- [Немного о CSS](#)
- [Структура и правила](#)
- [Внутренние таблицы стилей](#)
- [Глобальные таблицы стилей](#)
- [Связанные таблицы стилей](#)

Свойства CSS

- [Свойства Font](#)
- [Свойства Text](#)
- [Свойства Color и Background](#)
- [Свойства Box](#)
- [Классификация](#)

Дополнения

- [Меры длины](#)
- [Процентные меры](#)
- [Цвета](#)
- [Ссылки](#)

Часть I. Что есть CSS



- **Ищу рекламные места. Куплю интернет трафик дорого**
- **Покупаю ссылки с сайтов. Обращайтесь**
- **Как зарабатывать до \$100 в сутки в интернете?**

Немного о CSS

Так что это такое - CSS? Cascading Style Sheets (Таблицы Каскадных Стилей) - это язык, содержащий набор свойств для описания внешнего вида любых HTML документов. С его помощью дизайнер имеет полный контроль над стилем и расположением каждого элемента веб страницы, что проще и гораздо функциональнее использования обычного набора HTML тегов. Приведу пример: Вам нужно создать жирный красный подчеркнутый текст.

ПРИМЕР HTML:

```
<font color="red"><strong><u> Какой-то текст </u></strong></font>
```

А если подобный стиль нужно использовать несколько раз? Хорошо если раз 5, а если 10-20? Вот тут нам и поможет CSS. Существует три вида таблиц стилей: Внутренние таблицы стилей, Глобальные таблицы стилей и Связанные таблицы стилей. Внутренние таблицы стилей (Inline Style Sheets) при помощи специального атрибута помещаются прямо в HTML теги. Глобальные (Global Style Sheets) определяют стиль элементов во всем документе. Связанные (Linked Style Sheets) могут быть использованы для нескольких документов сразу и хранятся во внешнем файле. Подробнее обо всем этом написано ниже.

Структура и правила

Селекторы (Selectors):

Синтаксис:

селектор {свойства}

Любой элемент HTML - это возможный CSS селектор. Свойства селектора определяют стиль элемента, для которого он определен.

ПРИМЕР:

```
H1 {color:red; size:20pt;}
```

Все элементы H1 в документе будут красного цвета, размером в 20 точек (pt, point).

Классовые селекторы (Class Selectors):

Синтаксис:

селектор.класс {свойства}

CLASS - атрибут элемента в HTML, определяющий его класс. В CSS можно описать собственные стили для различных классов одних и тех же элементов.

ПРИМЕР:

```
H1.blue {color:blue; size:20pt;}
```

Все элементы H1 с атрибутом CLASS="blue" станут синими.

Классы могут так же быть описаны без явного привязывания их к определенным элементам.

Синтаксис:

.класс {свойства}

ПРИМЕР:

```
.green {color:green;}
```

В данном случае все элементы с атрибутом CLASS="green" станут зелеными.

ID селекторы (ID Selectors):

Синтакс:

```
#id {свойства}
```

ID - индивидуально именованный стиль. С его помощью можно создавать стилистические исключения среди элементов одного класса.

Индификаторы используются в основном для придания одному или нескольким элементам одного класса индивидуальных свойств. Скажем, Вы создали класс blue - синий курсив. Но Вам понадобился жирный подчеркнутый текст синим курсивом. Конечно, можно создать новый класс, но зачем? Проще описать ID. Например "boldunderline". И все элементы класса blue с значением ID "boldunderline" станут жирным подчеркнутым синим курсивом. Произойдет как бы синтез свойств класса blue и идентификатора boldunderline.

ПРИМЕР:

```
<html>
<head>
<title> Пример CSS </title>
</head>
<style>
.blue {color:blue; font-style:italic}
#boldunderline {text-decoration:underline; font-weight:bold}
</style>
<body>
<p class="blue"> Здравствуйте, это моя домашняя страница. </p>
<p class="blue" id="boldunderline"> Пока еще в стадии разработки ... </p>
<p id="boldunderline">... Но скоро откроется </p>
</body>
</html>
```

Как видно из примера, атрибут ID может быть использован без указания класса (последний параграф примера. Тогда параграф будет обладать только свойствами ID "boldunderline" (в примере - жирный, подчеркнутый текст).



- **Ищу рекламные места. Куплю интернет трафик дорого**
- **Покупаю ссылки с сайтов. Обращайтесь**
- **Как зарабатывать до \$100 в сутки в интернете?**

Контекстуальные селекторы (Contextual Selectors):

Контекстные селекторы - это сочетания нескольких обыкновенных селекторов. Стиль задается только элементам в заданной последовательности в зависимости от каскадного порядка.

ПРИМЕР:

```
P EM {color:silver;}
```

В данном примере все элементы EM внутри элементов P будут иметь заданный стиль.

Придание нескольким элементам одинаковых свойств:

Скажем Вам нужно придать нескольким элементам Вашей веб страницы одинаковых свойств. В этом случае при определении селекторы перечисляются через запятую перед блоком свойств.

ПРИМЕР:

```
h1,h2,h3,p,strong {color:green; font-style:italic;}
```

Все элементы h1, h2, h3, p и strong будут зелеными.

Псевдоклассы и псевдоэлементы :

Синтаксис:

```
селектор:псевдокласс { свойства }  
селектор.класс:псевдокласс { свойства }  
селектор:псевдоэлемент { свойства }  
селектор.класс:псевдоэлемент { свойства }
```

Псевдоклассы и псевдоэлементы - это особые классы и элементы,

присущие CSS и автоматически определяемые поддерживаемыми CSS браузерами. Псевдоклассы различают разные типы одного элемента, создавая при определении собственные стили для каждого из них. Псевдоэлементы являются частями других элементов, задавая этим частям отличный от элемента в целом стиль.

Список псевдоклассов и псевдоэлементов :

Anchor Pseudo Classes - эти псевдоклассы элемента ``, обозначающего ссылку. Псевдоклассы этого элемента: (ссылка), `active` (активная ссылка), `visited` (посещенный ранее URL), `hover` (псевдокласс, возникающий при поднесении курсора к ссылке, не работает в Нетскейпе).

First Line Pseudo-element - `first-line`. Этот псевдоэлемент может быть использован с `block-level` элементами (`p`, `h1` и т.д.). Он изменяет стиль первой строки этих элементов.

First Letter Pseudo-element - `first-letter`. Похож на `first-line`, но влияет не на всю строку, а только на первый символ.

ПРИМЕР :

```
a:link,a:visited {color:blue}
a:active {color:red}
a:hover {text-decoration:none}
```

В данном примере все элементы `Anchor` (ссылки) будут синими. При нажатии (в активном состоянии) поменяют цвет на красный. И при подведении курсора мышки исчезнет подчеркивание.

Примечание : описания нескольких свойств для одного селектора, контекстуального селектора, класса, индивидуально именованного стиля или группы объединенных селекторов отделяются друг от друга точкой с запятой ";".

Внутренние Таблицы Стилей

Как уже говорилось, использование Внутренних стилей мало чем отличается от использования обычных HTML тегов. Они задают стиль только одному элементу документа при помощи атрибута `STYLE` в HTML теге.

ПРИМЕР HTML:

```
<font color="blue" size="3" face="Arial"> Вперед в будущее </font>
```

ПРИМЕР INLINE STYLE SHEET:

```
<font style="color:blue; font-size:12pt; font-family:Arial"> Вперед в будущее </font>
```

Как можно заметить, код Inline Style Sheet получился больше чем HTML. Поэтому ISS следует использовать только если необходимо задать определенному элементу свой индивидуальный стиль, существующий в классификации CSS и нереализованный в HTML. Или же при необходимости абсолютно позиционировать данный элемент.

Глобальные Таблицы Стилей

Глобальные стили задают вид элементов всего документа. Для этого используется тег `<STYLE type="text/css">`. Он размещается в заголовке документа.

ПРИМЕР:

```
<html>
<head> <title> Пример Глобальных Таблиц Стилей </title>
</head>
<STYLE type="text/css">
h1{color:red; font-style:italic; font-size:32px}
.blue{color:blue}
#bold{font-weight:bold}
</STYLE>
<body>
<h1> Этот заголовок написан крупным красным курсивом </h1>
Вот <font class="blue"> это </font> слово - синие, а <font id="bold"> это</font> - жирное.
</body>
</html>
```

В данном примере все элементы H1 будут написаны крупным красным курсивым, все элементы с указанным классом BLUE будут синими, а все элементы с идентификатором ID="Bold" станут жирными. Для простоты вместо `<STYLE type="text/css">` можно использовать просто тег `<STYLE>`, что менее грамотно.

Связанные Таблицы Стилей

Связанные таблицы стилей используются для придания нескольким документам одного стиля и хранятся в отдельном файле. Это очень привлекательно, когда нужно выдержать сайт в одном стиле, не утруждая себя составлением таблиц для каждого документа.

ПРИМЕР:

Файл styles.css

```
<STYLE type="text/css">
body {background:black; font-size:9pt; color:red; font-family:Arial Black}
.base{color:blue; font-style:italic}
h1 {color:white}
#bold {font-weight:bold}
</STYLE>
```

В самих же HTML документах делается ссылка на этот файл при помощи тега <LINK>. Выглядит это так: <LINK rel="STYLESHEET" TYPE="text/css" HREF="путь до файла">

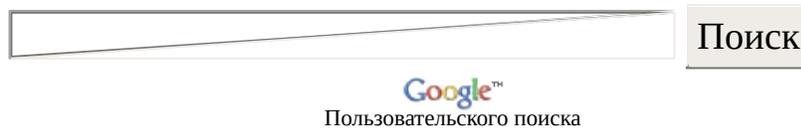
ПРИМЕР:

Файл Index.html

```
<html>
<head>
<title> Просто еще один пример </title>
</head>
<LINK rel="stylesheet" type="text/css" href="styles.css">
<body>
Содержание Документа
</body>
</html>
```

На этом я заканчиваю первую часть руководства и перехожу ко второй части.

Часть II. Свойства CSS



- [Ищу рекламные места. Куплю интернет трафик дорого](#)
- [Покупаю ссылки с сайтов. Обращайтесь](#)
- [Как зарабатывать до \\$100 в сутки в интернете?](#)

Свойства Font

- **font-family**

Возможные значения:

[1] любой шрифт

**Применимо для:* всех элементов

Описание: это свойство определяет используемый элементом шрифт. Если указать URL, то шрифт автоматически установится на компьютер пользователя

ПРИМЕР:

```
font-family:Arial Black URL('arialblack.ttf')
```

- **font-style**

Возможные значения:

[1] normal - без изменений

[2] italic - курсив

**Применимо для:* всех элементов

Описание: стиль элемента. Курсивный или обычный

ПРИМЕР:

font-style:italic

- **font-variant**

Возможные значения:

[1] normal - без изменений

[2] small-caps - заменяет все маленькие буквы на большие

**Применимо для:* всех элементов

Описание: варианты отображения шрифта. Нетскейп не поддерживает это свойство

ПРИМЕР:

font-variant:small-caps

- **font-weight**

Возможные значения:

[1] normal - без изменений

[2] bold - жирный

[3] bolder - очень жирный (в MSIE не отличается от bold, в Нетскейпе от нормал)

[4] lighter - тонкий (не отличается от normal)

[5] любое значение от 100 до 900

**Применимо для:* всех элементов

Описание: выделение (жирность) элемента

ПРИМЕР:

font-weight:bold

- **font-size**

Возможные значения:

[1] размер (+)

[2] xx-small, x-small, small, medium, large, x-large, xx-large - любое из этих значений

[3] smaller, larger - любое из этих значений

**Применимо для:* всех элементов

Описание: размер шрифта

ПРИМЕР:

font-size:30pt

- **font**

Возможные значения:

[1] font-family

[2] font-style

[3] font-variant

[4] font-weight

[5] font-size

**Применимо для:* всех элементов

Описание: обобщает вышеперечисленные свойства

ПРИМЕР:

font: italic bolder Arial 12pt

Свойства Text

- **word-spacing**

Возможные значения:

[1] длина (+)

[2] normal - без изменений

**Применимо для:* всех элементов

Описание: расстояние между словами. Не работает ни в Нетскейпе, ни в MSIE

ПРИМЕР:

word-spacing:0.4em

<UL

- **text-decoration**

Возможные значения:

[1] none - нет

[2] underline - подчеркнутый

[3] overline - надчеркнутый (не поддерживается в Нетскейпе)

[4] line-through - перечеркнутый

[5] blink - мигающий (не поддерживается в IE)

**Применимо для:* всех элементов

Описание: "украшение" текста

ПРИМЕР:

text-decoration:line-through

- **letter-spacing**

Возможные значения:

[1] длина (+)

[2] normal - без изменений

**Применимо для:* всех элементов

Описание: расстояние между буквами. Не работает в Нетскейпе

ПРИМЕР:

letter-spacing:100

- **vertical-align**

Возможные значения:

[1] baseline

[2] sub

[3] super

- [4] top-text
- [5] top
- [6] middle
- [7] bottom
- [8] bottom-text
- [9] процент

**Применимо для: inline элементов*

Описание: позиционирование элементов по отношению к другим элементам стоящих в одном ряду. Не работает в Нетскейпе

ПРИМЕР:

vertical-align:top-text

- **text-transform**

Возможные значения:

- [1] none - нет
- [2] Capitalize - каждое слово начинается с большой буквы
- [3] UPPERCASE - каждая буква текста становится заглавной
- [4] lowercase - каждая буква текста становится маленькой

**Применимо для: inline элементов*

Описание: изменение текста. Не работает в Нетскейпе

ПРИМЕР:

text-transform:Capitalize

- **text-align**

Возможные значения:

- [1] left - текст слева
- [2] right - текст справа
- [3] center - текст по центру
- [3] justify - текст "растянут"

**Применимо для: block-level элементов*

Описание: положение текста

ПРИМЕР:

text-align:right

- **text-indent**

Возможные значения:

[1] длина (+)

[2] процент (+)

**Применимо для:* block-level элементов

Описание: отступ

ПРИМЕР:

text-indent:30 em

- **line-height**

Возможные значения:

[1] normal - без изменений

[2] длина (+)

[3] процент

**Применимо для:* всех элементов

Описание: отступ сверху

ПРИМЕР:

line-height:100%

Свойства Color и Background

- **color**

Возможные значения:

[1] цвет (+)

**Применимо для:* всех элементов

Описание: цвет

ПРИМЕР:

color:#f00000

- **background-color**

Возможные значения:

[1] цвет (+)

**Применимо для:* всех элементов

Описание: цвет фона элемента

ПРИМЕР:

background-color:#f00000

- **background-image**

Возможные значения:

[1] none - нет

[2] URL (+)

**Применимо для:* всех элементов

Описание: фоновое изображение

ПРИМЕР:

background-image:URL(cool.gif)

- **background-repeat**

Возможные значения:

[1] repeat - размножает фоновое изображение во всех направлениях

[2] repeat-x - размножает фоновое изображение горизонтально

[3] repeat-y - размножает фоновое изображение вертикально

[4] no-repeat - не повторяющиеся изображение

**Применимо для:* всех элементов

Описание: повторения фонового изображения

ПРИМЕР:

background-repeat:no-repeat

- **background-attachment**

Возможные значения:

[1] scroll - фоновое изображение скроллится вместе с содержанием документа

[2] fixed - не скроллится. Фиксируется в одном месте. Не работает в Нетскейпе

**Применимо для:* всех элементов

Описание: возможность прокрутки фонового изображения

ПРИМЕР:

background-attachment:fixed

- **background-position**

Возможные значения:

[1] процент от ширины + процент от высоты (+)

[2] top, middle, bottom - одно из значений

[3] left, center, right - одно из значений

[4] расстояние от левого края + расстояние от вершины

**Применимо для:* block-level и replaced элементов

Описание: положение фонового изображения (работает с background-repeat равным repeat-x, repeat-y или no-repeat)

ПРИМЕР:

background-position:50%0%

- **background**

Возможные значения:

- [1] background-color
- [2] background-image
- [3] background-position
- [4] background-attachment
- [5] background-repeat

**Применимо для:* всех элементов

Описание: обобщает вышеперечисленные свойства

ПРИМЕР:

```
background:no-repeat black fixed 50%0%
```

Свойства Box

- **margin-top**

Возможные значения:

- [1] длина (+)
- [2] процент (+)
- [3] auto - автоматически

**Применимо для:* всех элементов

Описание: определяет отступ сверху

ПРИМЕР:

```
margin-top:100
```

- **margin-right**

Возможные значения:

- [1] длина (+)
- [2] процент (+)
- [3] auto - автоматически

**Применимо для:* всех элементов

Описание: определяет отступ справа

ПРИМЕР:

margin-right:100%

- **margin-bottom**

Возможные значения:

[1] длина (+)

[2] процент (+)

[3] auto - автоматически

**Применимо для:* всех элементов

Описание: определяет отступ снизу

ПРИМЕР:

margin-bottom:100em

- **margin-left**

Возможные значения:

[1] длина (+)

[2] процент (+)

[3] auto - автоматически

**Применимо для:* всех элементов

Описание: определяет отступ слева

ПРИМЕР:

margin-left:100pt

- **margin**

Возможные значения:

[1] margin-top

[2] margin-right

[3] margin-left

[4] margin-bottom

**Применимо для:* всех элементов

Описание: обобщает все вышеперечисленные свойства

ПРИМЕР:

background:100pt

- **padding-top**

Возможные значения:

[1] длина (+)

[2] процент (+)

**Применимо для:* осех элементов

Описание: отступ от верхнего border'a

ПРИМЕР:

padding-top:100pt

- **padding-right**

Возможные значения:

[1] длина (+)

[2] процент (+)

**Применимо для:* всех элементов

Описание: отступ от правого border'a

ПРИМЕР:

padding-right:100%

- **padding-bottom**

Возможные значения:

[1] длина (+)

[2] процент (+)

**Применимо для:* всех элементов

Описание: отступ от нижнего border'a

ПРИМЕР:

```
padding-bottom:100em
```

- **padding-left**

Возможные значения:

[1] длина (+)

[2] процент (+)

**Применимо для:* всех элементов

Описание: отступ от левого border'a

ПРИМЕР:

```
padding-top:100
```

- **padding**

Возможные значения:

[1] padding-top

[2] padding-right

[3] padding-left

[4] padding-bottom

**Применимо для:* всех элементов

Описание: обобщает вышеперечисленные свойства. Можно задать несколько значений одновременно (до четырех) для разных сторон. Если установлено одно значение - задается единый отступ для всех сторон, если два - то задаются различные отступы для прилежащих сторон, а если четыре - то задаются индивидуальные отступы для всех сторон.

ПРИМЕР:

padding:100px

- **border-top-width**

Возможные значения:

[1] длина (+)

[2] thin, medium или thick

**Применимо для:* всех элементов

Описание: толщина верхнего border'a

ПРИМЕР:

border-top-width:100pt

- **border-right-width**

Возможные значения:

[1] длина (+)

[2] thin, medium или thick

**Применимо для:* всех элементов

Описание: толщина правого border'a

ПРИМЕР:

border-right-width:thick

- **border-bottom-width**

Возможные значения:

[1] длина (+)

[2] thin, medium или thick

**Применимо для:* всех элементов

Описание: толщина нижнего border'a

ПРИМЕР:

border-bottom-width:100em

- **border-left-width**

Возможные значения:

[1] длина (+)

[2] thin, medium или thick

**Применимо для:* всех элементов

Описание: толщина левого border'a

ПРИМЕР:

border-left-width:medium

- **border-width**

Возможные значения:

[1] border-top-width

[2] border-right-width

[3] border-left-width

[4] border-bottom-width

**Применимо для:* всех элементов

Описание: толщина border'ов. Можно задать несколько значений одновременно (до четырех) для разных border'ов. Если установлено одно значение - задается единая толщина для всех сторон, если два - то задаются различная толщина для прилежащих сторон, а если четыре - то задаются индивидуальная толщина для всех сторон

ПРИМЕР:

border-width: 15pt

- **border-color**

Возможные значения:

[1] цвет (+)

**Применимо для:* всех элементов

Описание: Цвет border'a. Не работает в Нетскейпе

ПРИМЕР:

```
border-color:green
```

- **border-style**

Возможные значения:

[1] none

[2] dotted, dashed, solid, double, groove, ridge, inset, outset

**Применимо для:* всех элементов

Описание: стиль border'ов. Можно задать несколько значений одновременно (до четырех) для разных border'ов. Если установлено одно значение - задается единый стиль для всех сторон, если два - то задаются различные стили для прилежащих сторон, а если четыре - то задаются индивидуальные стили для всех сторон

ПРИМЕР:

```
border-style: dotted groove
```

```
<UL
```

- **border-top**

Возможные значения:

[1] border-top-width

[2] border-style

[3] border-color

**Применимо для:* всех элементов

Описание: обобщает вышеперечисленные свойства для верхнего border'a

ПРИМЕР:

```
border-top: 100em red groove
```

- **border-right**

Возможные значения:

- [1] border-right-width
- [2] border-style
- [3] border-color

**Применимо для:* всех элементов

Описание: обобщает вышеперечисленные свойства для правого border'a

ПРИМЕР:

border-right: 5pt magenta solid

- **border-left**

Возможные значения:

- [1] border-left-width
- [2] border-style
- [3] border-color

**Применимо для:* всех элементов

Описание: обобщает вышеперечисленные свойства для левого border'a

ПРИМЕР:

border-left: 15pc coral inset

- **border-bottom**

Возможные значения:

- [1] border-bottom-width
- [2] border-style
- [3] border-color

**Применимо для:* всех элементов

Описание: обобщает вышеперечисленные свойства для нижнего border'a

ПРИМЕР:

border-bottom: 30 orange outset

- **border**

Возможные значения:

[1] border-width

[2] border-style

[3] border-color

**Применимо для:* всех элементов

Описание: обобщает вышеперечисленные свойства

ПРИМЕР:

border: thick black double

- **width**

Возможные значения:

[1] длина (+)

[2] процент (+)

**Применимо для:* block-level и replaced элементов

Описание: ширина элемента

ПРИМЕР:

width:10%

- **height**

Возможные значения:

[1] длина (+)

[2] процент (+)

**Применимо для:* block-level и replaced элементов

Описание: высота элемента

ПРИМЕР:

height:100pt

- **float**

Возможные значения:

[1] left - слева

[2] right - справа

[3] none - по умолчанию

**Применимо для: всех элементов*

Описание: расположение элемента

ПРИМЕР:

float:right

- **clear**

Возможные значения:

[1] left - слева

[2] right - справа

[3] both - с двух сторон

[4] none - по умолчанию

**Применимо для: всех элементов*

Описание: расположение других элементов вокруг данного

ПРИМЕР:

clear:both

Классификация

- **display**

Возможные значения:

[1] none - не отображается

[2] block - разбивает строку до и после элемента (т.е. элемент не может

находится на одной линии с другими элементами)

[3] inline - не разбивает строку

[4] list-item - разбивает линию строку до и после элемента + добавляет маркер как у list-item элементов

**Применимо для:* всех элементов

Описание: определяет, как будет отображаться элемент

ПРИМЕР:

display:none

- **white-space**

Возможные значения:

[1] normal - "сжимает" несколько подряд идущих пробелов в один

[2] pre - допускает отображение нескольких подряд идущих пробелов

[3] nowrap - не допускает перенос строки без тега

**Применимо для:* block-level элементов

Описание: определяет, как будут отображаться пробелы между элементами

ПРИМЕР:

white-space:nowrap

- **list-style-type**

Возможные значения:

[1] disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha или upper-alpha

[2] none - никакой

**Применимо для:* элементов со значением display равным list-item

Описание: определяет вид list-item маркера. Если значение list-style-image равно none или не уточнено

ПРИМЕР:

list-style-type:lower-alpha

- **list-style-image**

Возможные значения:

[1] none - нет

[2] URL (+)

**Применимо для:* элементов со значением display равным list-item

Описание: задает вид list-item маркера в виде картинки

ПРИМЕР:

list-style-image:URL(cool.gif)

- **list-style-position**

Возможные значения:

[1] inside - при переносе следующие строки будут отображаться без отступа

[2] outside - по умолчанию

**Применимо для:* элементов со значением display равным list-item

Описание: определяет положение маркера в зависимости от list item элемента

ПРИМЕР:

list-style-position:inside

- **list-style**

Возможные значения:

[1] list-style-type

[2] list-style-position

[3] list-style-image

**Применимо для:* элементов со значением display равным list-item

Описание: обобщает вышеперечисленные свойства

ПРИМЕР:

`list-style:inside`

Часть III. Дополнения



- **Ищу рекламные места. Куплю интернет трафик дорого**
- **Покупаю ссылки с сайтов. Обращайтесь**
- **Как зарабатывать до \$100 в сутки в интернете?**

Меры длины

Синтаксис: "+" (можно опустить) или "-" затем [число] плюс [единица измерения] (без пропусков)

ПРИМЕР:

-566pt

Единицы длины : *ex* - *x-height*, ширина буквы "x" используемого элементом шрифта

px - *pixels*, пиксели

in - *inches*, дюймы

cm - *centimeters*, сантиметры

mm - *millimeters*, миллиметры

pt - *points*, точка (1pt = 1/72in)

pc - *picas* (1pc = 12pt)

Процентные меры

Синтаксис: "+" или "-" затем [число] плюс "%" (без пропусков)

ПРИМЕР :

-566%

Цвета

Синтаксис: [color]

ПРИМЕР :

magenta

Значением цвета может быть его название (*red* , *lightgreen*, *coral* и т.д.) или *RGB* значение

Способы выразить цвет в *RGB* (*red green blue*) :

- *#rrggbb* (например, *#00cc00*)
- *rgb(x,x,x)* -- где "x" число от 0 до 255 (например, *rgb(0,204,0)*)
- *#rgb* (например, *#0c0*)
- *rgb (x%,x%,x%)* -- где "x%" число от 0.0 до 100.0 (например, *0%,80%,0%*)

Все примеры отображают один и тот же цвет

Ссылки

Синтаксис: "URL" , потом в скобках приводится ссылка.Ссылку также можно, помимо скобок, заключить в одинарные или двойные кавычки (без пропусков)

ПРИМЕР :

URL('cool.gif')