**Основы CSS**

**Что такое CSS?**

Рассмотренные в первой главе теги HTML позволяют задать структуру сайта и разместить различные элементы на веб-странице. Раньше, для того, чтобы оформить веб-страницу, т.е. добавить цвета, выравнивание, параметры шрифтов и т.п., использовали специальные атрибуты тегов. Однако у такого способа были ограниченные возможности в оформлении и сайты получались скучные и однотипные.

Пример того, как не нужно оформлять веб-страницу!

<body bgcolor="green">

<h1 align="center">

<font color="red">Заголовок</font>

</h1>

</body>

В настоящее время для оформления веб-страниц используются каскадные таблицы стилей CSS. При помощи CSS можно менять цвет и шрифт у текста, изменять положение элементов на странице, их размеры, задавать элементам рамки, границы и отступы.

HTML – структура сайта

CSS – оформление сайта

**CSS (Cascading Style Sheets)** – каскадные таблицы стилей, которые применяются для описания внешнего вида веб-документа, написанного при помощи языка разметки HTML.

Термин «каскадные таблицы стилей» был предложен Хоконом Ли в 1994 году. Совместно с Бертом Босом он стал развивать CSS. В настоящее время используется CSS версии 3.

CSS представляет из себя набор правил, описывающих форматирование (изменение внешнего вида) отдельных элементов на веб-странице. Правило, состоит из двух частей: **селектора** и следующего за ним **блока объявлений**.

**Синтаксис CSS**

Селектор {

Свойство 1: значение 1;

Свойство 2: значение 2;

}

Блок объявлений (стили)

CSS правило

Первым всегда указывается *селектор*, он сообщает браузеру, к какому элементу веб-страницы будет применен стиль. Селекторами могут быть любые теги, формирующие тело сайта, а также идентификаторы, классы и атрибуты тегов.

Далее следует *блок объявлений*, который начинается открывающейся фигурной скобкой { и заканчивается закрывающейся фигурной скобкой }. Между фигурными скобками указываются команды CSS – объявления.

Каждое *объявление* состоит из двух частей: свойства и его значения. Объявление всегда должно заканчиваться точкой с запятой.

*Свойство* – это команда форматирования, которая задает конкретный стиль для элемента. Каждое свойство имеет предопределенный набор значений. После имени свойства указывается двоеточие, которое отделяет название свойства от допустимого значения.

Пример CSS правила:

h1 {

color: red;

font-size: 20px;

}

В данном примере селектором является элемент h1 – заголовок первого уровня. Заметьте, что тег h1 записывается в CSS без угловых скобок. К заголовку применяются два стиля:

* color: red – свойства color со значение red задает красный цвет шрифта у заголовка;
* font-size: 20px – свойство font-size со значением 20px задает размер шрифта у заголовка.

Выше рассмотрен многострочный вариант расположения команд CSS. Он считается более наглядным, безопасным и удобным. Поэтому именно многострочный вариант чаще рекомендуется для использования. Однако иногда можно применять и однострочный вариант расположения команд.

h1 {color: red; font-size: 20px;}

Для задания комментарий в CSS используется конструкция /\* … \*/. Такая конструкция позволяет комментировать одну или несколько строк CSS.

/\* здесь пишется комментарий \*/

/\*

и здесь можно

тоже указать комментарий\*/

**Подключение CSS к HTML**

Для того, чтобы использовать стили CSS в веб-документе, необходимо их сначала подключить. В зависимости от способа подключения выделяют внешние таблицы стилей, внутренние и встроенные стили.

1. **Внешняя таблица стилей** представляет собой созданный в текстовом редакторе файл с расширением CSS, в котором находится весь CSS-код веб-страницы. Внутри файла содержатся только стили без HTML-разметки.

Внешняя таблица стилей (CSS-файл) подключается к html-документус помощью тега **<link>**, расположенного внутри раздела <head></head>.

<head>

<link href ="style.css" rel = "stylesheet">

</head>

Тег <link>имеет два обязательных атрибута: href и rel.В атрибуте href указывается путь к внешнему CSS-файлу. Атрибут rel = "stylesheet" показывает тип ссылки. Значение stylesheet указывает на то, что подключается файл стилей CSS.

У тега <link> имеется также атрибут type = "text/css", но по стандарту HTML5 он не является обязательным, поэтому его можно не указывать.

Внешнюю таблицу стилей можно подключить ко всем страницам сайта.

К веб-странице можно присоединить несколько таблиц стилей, добавляя последовательно несколько тегов <link>, каждый из которых будет указывать на свой файл с таблицей.

*Пример веб-страницы с внешней таблицей стилей (style.css)*

***Файл index.html***

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Способы подключения CSS</title>

<link href ="style.css" rel = "stylesheet">

</head>

<body>

<p>Внешняя таблица стилей</p>

</body>

</html>

***Файл style.css***

body{

background-color: grey; /\*цвет фоновой заливки: серый \*/

}

p{

color: blue; /\*цвет текста: синий\*/

font-size: 16px; /\*размер текста: 16px\*/

text-align: center; /\*выравнивание текста: по центру\*/

}

В данном примере к файлу index.html подключается внешняя таблица стилей – файл style.css. В CSS файле задаются стили для тела сайта body и для абзаца p. В комментариях расшифрованы использованные свойства и их значения. В результате применения стилей CSS веб-страница будет оформлена следующим образом: на сером фоне по центру отображается синий текст с размером букв 16px.

Использование внешней таблицы стилей является самым удобным способом подключения CSS. Можно выделить следующие преимущества такого способа:

* меньший размер html-страницы и более чистая структура кода;
* быстрая скорость загрузки html-страницы;
* для разных веб-страниц может быть использован один css файл;
* удобство в редактировании стилей.

1. **Внутренние стили** являются частью кода html-страницы. Они располагаются внутри тега **<style></style>**, расположенного в разделе <head></head>.

Рассмотренный пример с использованием внутренних стилей будет выглядеть следующим образом:

***Файл index.html***

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Способы подключения CSS</title>

<style>

body{

background-color: grey;

}

p{

color: blue;

font-size: 16px;

text-align: center;

}

</style>

</head>

<body>

<p>Внешняя таблица стилей</p>

</body>

</html>

Внутренние стили имеют приоритет над внешними, но уступают встроенным стилям.

1. **Встроенные стили (inline-стили)** заключаются в использовании атрибута style непосредственно у того элемента, который необходимо стилизовать. В качестве значения атрибута указываются CSS-свойства.

<body style = "background-color: grey; ">

<ul>

<li style = "color: red; font-size: 18px; "> Монитор </li>

<li> Клавиатура</li>

</ul>

</body>

Минус от использования встроенных стилей заключается в том, что они создают дополнительные неудобства: смешивается структура документа с его оформлением, поиск и правка таких стилей занимает достаточно много времени. Однако встроенные стили обладают наибольшим приоритетом.

**Селекторы CSS**

Селекторы представляют структуру веб-страницы. С их помощью создаются правила для форматирования элементов веб-страницы. Селекторами могут быть теги, их классы и идентификаторы, а также атрибуты.

1. **Селектор тега**

В качестве селектора может выступать любой тег HTML.

В рассмотренных выше примерах использовались именно селекторы тегов.

**Синтаксис CSS**

тег {

свойство 1: значение 1;

свойство 2: значение 2;

}

Селектор тега применяется, когда необходимо задать один стиль для всех элементов с одинаковым тегом. Например, если нужно задать отступ у первой строки для всех абзацев или цвет всех заголовков второго уровня.

Однако, часто необходимо изменить цвет не всех заголовков на веб-странице, а только одного или двух. CSS предоставляет такую возможность с помощью селекторов id (идентификатор) и class (класс).

1. **Селектор id (идентификатор)** предназначен для применения стиля к уникальным элементам на веб-странице. **Каждый идентификатор может встречаться на странице только один раз.**

Чтобызадать данный селектор, следует для тега, к которому нужно применить стиль, добавить атрибут id и в его значении которого указать уникальное имя. В CSS коде селектор id начинается с символа #, после которого идет имя идентификатора.

**Синтаксис CSS**

#имя идентификатора {

свойство 1: значение 1;

свойство 2: значение 2;

}

*Пример использования селектора id*

***Файл index.html***

<p id = "first">Первый абзац </p>

<p> Второй абзац </p>

<p> Третий абзац </p>

<p> Четвертый абзац </p>

<p> Пятый абзац </p>

***Файл style.css***

#first {

color: red; /\*цвет текста: красный\*/

font-weight: bold; /\*толщина текста: полужирное начертание\*/

}

В данном примере указанные стили будут применены только к первому абзацу, для которого задан id = "first".

1. **Селектор class (класс)** позволяет применить стиль к нескольким подобным элементам на веб-странице.

Для использования селектора class следует для тегов, к которым необходимо применить стиль, добавить атрибут class. В качестве значения указать определённое имя класса. В CSS коде селектор class начинается с точки, после которого идет имя класса.

Также можно использовать классы и с указанием тега, к которому они применяются.

**Синтаксис CSS**

тег.имя класса {

свойство 1: значение 1;

свойство 2: значение 2;

}

.имя класса {

свойство 1: значение 1;

свойство 2: значение 2;

}

Имя идентификатора и класса всегда задается на английском языке и начинаться с буквы. Оно может содержать цифры, символ дефиса ( - ) и нижнего подчеркивания ( \_ ).

*Пример использования селектора class*

***Файл index.html***

<p id = "first" class= "odd">Первый абзац </p>

<p> Второй абзац </p>

<p class= "odd"> Третий абзац </p>

<p> Четвертый абзац </p>

<p class= "odd"> Пятый абзац </p>

***Файл style.css***

#first {

color: red; /\*цвет текста: красный\*/

font-weight: bold; /\*толщина текста: полужирное начертание\*/

}

.odd{

background-color: #eee; /\*цвет заливки: серый\*/

}

В данном примере все нечетные абзацы с атрибутом class= "odd" будут выделены серым цветом.

Для уточнения элемента, к которому применяется стиль, в CSS можно указывать тег, у которого задан класс.

p.odd{

background-color: #eee;

}

Классы можно задавать не только для одинаковых тегов. В следующем примере атрибут class="is\_border" указан у разных тегов, служащих для отображения текстовой информации: у заголовка и у абзаца.

***Файл index.html***

<h1 class= "is\_border">Заголовок с рамкой </h1>

<p class= "is\_border"> Абзац с рамкой </p>

***Файл style.css***

.is\_border{

border: 1px solid #000; /\*рамка: толщина 1px сплошная черного цвета\*/

}

Для одного элемента иногда необходимо применить несколько классов, при этом их имена в значении атрибута следует указывать через пробел. Например, в следующем примере для заголовка заданы два класса с именами heading и logo. У каждого класса в CSS описаны свои стили.

***Файл index.html***

<h1 class= "heading logo">Заголовок</h1>

***Файл style.css***

.heading{

text-transform: uppercase; /\*все символы заглавные\*/

width: 200px; /\*ширина текстового блока: 200px\*/

}

.logo{

border: 1px solid #abc;

}

1. **Универсальный селектор**

Часто возникает необходимость использования единого стиля одновременно для всех элементов веб-страницы. Для этого используется универсальный селектор. Стиль, определённый для универсального селектора, будет выполнен над всеми элементами.

Универсальный селектор записывается символом \*.

**Синтаксис CSS**

\* {

свойство 1: значение 1;

свойство 2: значение 2;

}

**Основные свойства стилей CSS**

При изучении способов подключения и селекторов CSS уже были рассмотрены некоторые примеры свойств стилей и их значений.

В данной теме будут представлены часто используемые свойства CSS и их возможные значения.

1. **Фон элемента – свойство background**

Свойство background позволяет установить характеристики фона для различных элементов веб-страницы.

***Характеристики фона***

а) Свойство ***background-color***– устанавливает цвет фона у элемента.

**Синтаксис CSS**

background-color: red;

background-color: #00af45;

background-color: #a0c;

background-color: rgba(255, 128, 128, 0.5);

background-color: transparent;

В примере выше показаны различные способы задания цвета на веб-странице:

* background-color: red; – используется название цвета на английском языке. При таком подходе есть ограничения, т.к. невозможно получить различные оттенки цветов.
* background-color: #00af45; – используется шестнадцатеричный код числа в модели RGB. Такой способ позволяет выбрать один из 16 млн. цветов.

Цифровая модель RGB расшифровывается как Red-Green-Blue, то есть Красный-Зеленый-Синий. Цвет в RGB представляется в виде трех пар шестнадцатеричных цифр (от 0 до 9, A, B, C, D, E, F), где каждая пара отвечает за свой цвет: две первые цифры – красный, две в середине – зеленый, две последние цифры – синий.

Чем больше значение цифры, тем сильнее данный оттенок проявит себя в цвете. Например, в цвете, заданном как #00af45, большее значение у зеленой составляющей (рис. 71). В результате получится некоторый оттенок зеленого цвета.

#00af45

red: 00

green: af

blue: 45

Рис. 71 Представление цвета в RGB

* background-color: #a0c; – сокращенная форма записи цвета в RGB. Используется, если совпадают два числа или две буквы одного и того же цвета.

#a0c

#aa00cc

Рис. 72 Сокращенная форма записи цвета в RGB

* background-color: rgba(255, 128, 128, 0.5); – используется десятичный код числа в модели RGB и значение прозрачности.

Каждая из трех компонент RGB (красный, зеленый и синий) может принимать значение от 0 до 255. Последний параметр (альфа-канал) отвечает за прозрачность фона: значение 0 соответствует полной прозрачности, 1 – непрозрачность, промежуточное значение вроде 0.5 – полупрозрачности.

rgba(255, 128, 128, 0.5)

red: 255

green: 128

blue: 128

полупрозрачный

Рис. 73 Представление цвета в rgba

* background-color: transparent; – устанавливает прозрачный фон.

При подборе цвета лучше использовать любой графический редактор (Gimp, Photoshop). В нем можно выбрать нужный цвет из палитры, и редактор покажет шестнадцатеричный и десятичный код выбранного цвета.

Подобрать цвета также можно с помощью различных онлайн сервисов: https://colorscheme.ru/, https://color.romanuke.com/ и др.

б) Свойство ***background-image***– устанавливает одно или несколько фоновых изображений для элемента.

**Синтаксис CSS**

background-image: url(img/foto.jpg); /\*одно фоновое изображение\*/

background-image: url(<http://site.ru/rose.png>);

background-image: url(t1.png), url(tr.png); /\*несколько фоновых изображений\*/

background- image: none; /\*отменяет фоновое изображение\*/

В качестве значения используется путь к графическому файлу, который указывается внутри конструкции url(). Путь к файлу при этом можно писать, как в кавычках (двойных или одинарных), так и без них.

Можно добавить несколько фоновых изображений для элемента, перечислив их параметры через запятую.

в) Свойство ***background-position***– указывает, расположение фонового изображения.

**Синтаксис CSS**

background-position: left; /\*слева\*/

background-position: right; /\*справа\*/

background-position: top; /\*сверху\*/

background-position: bottom; /\*снизу\*/

background-position: center; /\*по центру\*/

Часто расположение фонового изображения задается сразу для двух осей, в этом случае значения свойства background-position указываются через пробел. Например, если изображение нужно вывести в правом верхнем углу, то команда будет выглядеть следующим образом:

background-position: top right;

Можно управлять расстоянием для фонового изображения от левого и верхнего краев элемента с помощью точных значений, указанных в единицах измерения CSS (например, в пикселях или %).

background-position: 10px 25px;

Такая запись означает, что фоновое изображение удалено от левого края на 10 пикселей и от верхнего края на 25 пикселей. Здесь важен порядок записи значений: первое значение отвечает за положение фонового рисунка по горизонтали, второе — по вертикали. Допустимы и отрицательные значения (например, они могут быть полезны в случае, если с левой или верхней стороны фоновой картинки есть область, которую необходимо спрятать).

г) Свойство ***background-repeat***– определяет, нужно ли повторять фоновое изображение.

**Синтаксис CSS**

background-repeat: repeat-x; /\*повтор по горизонтали\*/

background-repeat: repeat-y; /\*повтор по вертикали\*/

background-repeat: repeat; /\*повтор по вертикали и горизонтали\*/

background-repeat: no-repeat; /\*нет повтора\*/

background-repeat: space; /\*изображение повторяется столько раз, чтобы полностью заполнить область; если это не удается, между картинками добавляется пустое пространство\*/

background-repeat: round; /\*изображение повторяется так, чтобы в области поместилось целое число рисунков; если это не удается сделать, то фоновые рисунки масштабируются\*/

д) Свойство ***background-size***– определяет размер фонового изображения.

**Синтаксис CSS**

background-size: 300px; /\*ширина изображения в px\*/

background-size: auto 30%; /\* ширина изображения вычисляется автоматически с сохранением пропорций, высота изображения – 30%\*/

background-size: auto auto; /\*исходные размеры изображения\*/

background-size: cover; /\*масштабирование изображения с сохранением пропорций на всю ширину или высоту элемента\*/

background-size: contain; /\*масштабирование изображения с сохранением пропорций, так чтобы изображение целиком поместилось в элемент\*/

background-size: 6px, cover; /\*размеры для двух фоновых картинок: у первой – ширина 6px, вторая – растянута на всю область фона\*/

Если в значении свойства background-size указано одно значение, то это значение ширины изображения, высота при этом вычисляется автоматически, исходя из пропорций картинки (значение auto). Если в значении свойства background-size указано два значения, то первое значение – это ширина изображения, второе – высота.

Значение auto, заданное одновременно для ширины и высоты (background-size: auto auto), сохраняет исходные размеры фоновой картинки. Если auto установлено только для одной стороны картинки (background-size: auto 30%), то размер этой стороны вычисляется автоматически исходя из пропорций картинки.

Значения для нескольких фонов указываются через запятую.

В CSS используется несколько единиц измерения для определения размеров элементов. К ним относятся:

* пиксели (px) – единицы фиксированного размера;
* проценты (%) – масштабируют размер элемента в процентах относительно родительского элемента;
* относительная единица (em) – 1em равен текущему размеру элемента, 2em увеличивает размер в 2 раза и т.д. (em часто используется для размера шрифта).

д) Свойство ***background-attachment***– определяет, будет ли изображение прокручиваться вместе с содержимым элемента.

**Синтаксис CSS**

background-attachment: scroll; /\*фоновое изображение прокручивается вместе с текстом и другим содержимым \*/

background-attachment: fixed; /\*фоновое изображение остается неподвижным\*/

background-attachment: local; /\*фоновое изображение прокручивается вместе содержимым элемента\*/

Рассмотренные характеристики рекомендуется записывать в краткой форме, используя свойство background. Порядок написания значений может быть произвольным – браузер сам определит соответствие свойств и значений.

**Краткая запись:**

**background**:

background-color

background-image

background-repeat

background-position

background-size

Рис. 76 Краткая запись свойства background

background: #ff0 url(img/foto.jpg) repeat-x top/50px;

Как видно на рисунке и в примере, значения всех свойств указываются через пробел. Исключением являются свойства background-position и background-size – их необходимо разделять знаком слэша /.

1. **Ширина и высота элемента – свойства width и height**

**Синтаксис CSS**

width: 200px;

height: 50%;

В качестве значений принимаются любые единицы размеров, принятые в CSS – например, пиксели (px), проценты (%) и др. При использовании процентной записи ширина элемента вычисляется в зависимости от ширины родительского элемента. Если родитель явно не указан, то им выступает окно браузера.

1. **Рамка вокруг элемента – свойство border**

Свойство border позволяет установить характеристики рамок вокруг элемента на веб странице.

**Характеристики рамки**

а) Свойство ***border-color***– цвет рамки.

Свойство задаёт цвет рамок всех сторон одновременно. С помощью уточняющих свойств можно установить свой цвет для рамки каждой стороны элемента.

б) Свойство ***border-style***– стиль рамки.

Стиль рамки определяет ее отображение, без этого свойства рамка не будет видна вообще.

**Синтаксис CSS**

border-style: solid; /\*сплошная рамка\*/

border-style: dotted; /\*точечная рамка\*/

border-style: dashed; /\*пунктирная рамка\*/

border-style: double; /\*двойная рамка\*/

border-style: inset; /\*трехмерная рамка с углублением\*/

border-style: outset; /\*трехмерная выпуклая рамка\*/

border-style: groove; /\*трехмерная рамка с прорезанным желобом\*/

border-style: ridge; /\*трехмерная рамка с оттиснутой бороздой\*/

Стили рамок

Рис. 77 Значения свойства border-style

в) Свойство ***border-width***– толщина рамки.

Толщину рамки можно задать для всех сторон одинаковую или для каждой из 4 сторон разную.

**Синтаксис CSS**

border-width: 1px; /\*толщина всех сторон – 1px\*/

border-width: 1px 2px; /\*1px – верхняя и нижняя сторона, 2px – левая и правая\*/

border-width: 1px 2px 3px; /\*1px – верхняя, 2px – левая и правая, 3px – нижняя\*/

border-width: 1px 2px 3px 4px; /\*1px – верхняя, 2px – правая, 3px – нижняя, 4px – левая\*/

Рассмотренные характеристики рамки рекомендуется перечислять в одну строку, разделяя пробелом, используя свойство border.

**Краткая запись:**

**border:**

border-width

border-style

border-color

Рис. 78 Краткая запись свойства border

border: 1px solid black;

При этом заданные свойства будут применяться ко всем границам элемента одновременно. Также возможно каждую границу задавать отдельно. Для этого используются следующие свойства:

* border-top – стиль верхней границы;
* border-bottom – стиль нижней границы;
* border-left – стиль левой границы;
* border-right – стиль правой границы.

1. **Цвет текста – свойство color**

Цвет текста можно задавать любым способом, также как и цвет фона.

**Синтаксис CSS**

color: red;

color: #78fa2e;

color: RGB(34,21,56);

1. **Шрифт – свойство font**

Свойство **font** позволяет установить характеристики шрифта текста внутри элемента на веб странице.

**Характеристики шрифта**

а) Свойство ***font-family***– устанавливает семейство шрифта, которое будет использоваться для оформления текста внутри элемента.

Список шрифтов может включать одно или несколько значений, разделенных запятой. Если в имени шрифта содержатся пробелы, например, Times New Roman, то оно должно заключаться в одинарные или двойные кавычки.

**Синтаксис CSS**

font-family: Cambria, "Times New Roman", serif;

В данном примере, браузер обрабатывает список шрифтов, заданный свойством font-family, следующим образом:

1. сначала он проверяет, установлен ли шрифт Cambria на компьютере, и, если да, использует его в качестве шрифта для текста внутри элемента;
2. если Cambria не установлен, то ищет шрифт Times New Roman, в случае успешного поиска использует его для шрифта;
3. если ни один из указанных шрифтов не найден, то применяется первый найденный браузером на компьютере шрифт из семейства serif.

В CSS все шрифты разделены на семейства, каждое семейство состоит из набора шрифтов, обладающих общими характеристиками. Существует всего пять семейств шрифтов:

* *sans-serif* – шрифты без засечек (Helvetica, Geneva, Arial, Verdana, Trebuchet, Univers), считается что они лучше читаются на экране компьютера, чем шрифты семейства serif;
* *serif*– шрифты с засечками (Times New Roman, Times, Garamond, Georgia), засечки – это декоративные штрихи и черточки по краям букв;
* *monospace* – семейство, состоящее из шрифтов, символы которых имеют одинаковую фиксированную ширину (Courier, Courier New, Andele Mono);
* *cursive* – шрифты, имитирующие рукописный текст (Comic Sans, Gabriola, Monotype Corsiva, Author, Zapf Chancery);
* *fantasy* – художественные и декоративные шрифты (Western, Woodblock, Klingon). Они не очень широко распространены, доступны не на всех компьютерах и редко используются в веб-дизайне.

б) Свойство ***font-weight***– задает насыщенность шрифта.

Значение устанавливается от 100 до 900 с шагом 100. Сверхсветлое начертание, которое может отобразить браузер, имеет значение 100, а сверхжирное – 900. Нормальное начертание шрифта, которое установлено по умолчанию, эквивалентно 400, стандартный полужирный текст — значению 700.

Насыщенность шрифта также задаётся с помощью ключевых слов.

**Синтаксис CSS**

font-weight: normal; /\*нормальный шрифт (по умолчанию)\*/

font-weight: bold; /\*полужирный шрифт\*/

font-weight: bolder; /\*насыщенность шрифта больше, чем у родителя\*/

font-weight: lighter; /\*насыщенность шрифта меньше, чем у родителя\*/

font-weight: 500; /\*средняя насыщенность шрифта\*/

в)Свойство ***font-style*** – устанавливает стиль начертания шрифта.

**Синтаксис CSS**

font-style: normal; /\*обычное начертание шрифта (по умолчанию)\*/

font-style: italic; /\*начертание курсивом\*/

font-style: oblique; /\*наклонное начертание\*/

Разница между курсивом и наклонным начертанием заключается в том, что курсив вносит небольшие изменения в структуру каждого символа, в то время как наклонное начертание представляет собой наклонную версию прямого шрифта.

г) Свойство ***font-size*** – устанавливает размер (высоту) шрифта.

Существует несколько способов указания размера шрифта, наиболее распространенные из них:

* *px (пиксели);*
* *% (проценты);*
* *em* (размер шрифта изменяется относительно размера шрифта в родительском элементе);
* *ключевые слова*

CSS предлагает семь ключевых слов, которые позволяют назначить размер шрифта относительно размера по умолчанию (16px): *xx-small, x-small, small, medium, large, x-large и xx-large*.

Среднее значение medium - размер шрифта по умолчанию в браузерах. Остальные значения уменьшают или увеличивают размер шрифта с различными коэффициентами. Самый маленький размер шрифта xx-small равен примерно 9 пикселям, каждый последующий размер примерно на 20% больше предыдущего.

Другой набор ключевых слов: *larger* и *smaller* устанавливает изменение размера шрифта относительно родителя.

**Синтаксис CSS**

font-size: 20px;

font-size: 80%;

font- size: 0.8em;

font- size: small;

Указывать стиль шрифта можно при помощи сокращенной записи. В данном случае важен порядок следования значений:

**Краткая запись:**

**font**:

font-style

font-variant

font-weight

font-size

font-family

Рис. 79 Краткая запись свойства font

font: bold 24px Arial, Verdana, sans-serif;

В качестве обязательных значений свойства font указывается размер шрифта и его семейство. Остальные значения задаются при желании.

1. **Выравнивание текста по горизонтали – свойство text-align**

**Синтаксис CSS**

text-align: left; /\*выравнивание по левому краю\*/

text-align: right; /\*выравнивание по правому краю\*/

text-align: center; /\*выравнивание по центру\*/

text-align: justify; /\*выравнивание по ширине\*/

1. **Изменение регистра символов – свойство text-transform**

**Синтаксис CSS**

text-transform: capitalize; /\*каждое слово начинается с прописной (заглавной) буквы\*/

text-transform: lowercase; /\*все символы выводятся строчными буквами\*/

text-transform: uppercase; /\*все символы выводятся прописными буквами\*/

1. **Оформление текста – свойство text-decoration**

**Синтаксис CSS**

text-decoration: line-through; /\*перечёркнутый текст\*/

text-decoration: overline; /\*линия над текстом\*/

text-decoration: underline; /\*подчёркнутый текст\*/

text-decoration: none; /\*отменяет все эффекты, в том числе подчеркивание у ссылок\*/

1. **Оформление списков – свойство list-style**

Свойство list-style позволяет установить стиль маркера у списка на веб странице.

**Характеристики стиля маркера**

а) Свойство ***list-style-type***– изменяет тип маркера или удаляет маркер для маркированного и нумерованного списков.

**Синтаксис CSS**

list-style-type: none; /\*маркер отсутствует\*/

/\*значения для маркированного списка\*/

list-style-type: disc; /\*маркер в виде точки (по умолчанию) \*/

list-style-type: circle; /\*маркер в виде кружка\*/

list-style-type: square; /\*маркер в виде квадрата\*/

/\*значения для нумерованного списка\*/

list-style-type: decimal; /\*арабские числа (по умолчанию) \*/

list-style-type: lower-alpha; /\*строчные латинские буквы\*/

list-style-type: upper-alpha; /\*заглавные латинские буквы\*/

list-style-type: lower-roman; /\* римские числа в нижнем регистре \*/

list-style-type: upper-roman; /\*римские числа в верхнем регистре\*/

б) Свойство ***list-style-position*** – определяет расположение маркера.

**Синтаксис CSS**

list-style-position: outside; /\*маркеры размещаются за пределами блока с текстом\*/

list-style-position: inside; /\*маркер является частью текстового блока\*/

в) Свойство ***list-style-image*** – позволяет установить в качестве маркера списка изображение. Путь к изображение указывается в скобках после слова url.

**Синтаксис CSS**

list-style-image: url(img/book.svg);

Все три свойства форматирования можно объединить в одно свойство list-style. Значения свойств могут располагаться в произвольном порядке, часть значений можно опустить.

**Краткая запись:**

list-style:

list-style-type

list-style-position

list-style-image

Рис. 82 Краткая запись свойства list-style

list-style: circle inside;

1. ***Отступы***

Для создания промежутков между элементами на веб-странице используются отступы, которые могут быть внешними и внутрениими.

Свойство ***padding –*** внутренние отступы – расстояние от содержимого до края блока.

Свойство ***margin –*** внешние отступы (поля) – расстояние между блоками.

Значение свойств может быть указано в любых единицах CSS – px, %, em и т.д.

**Синтаксис CSS**

margin: 10px; /\*поля вокруг блока с элементом10px \*/

padding: 5px; /\* отступы от элемента до края блока 5px \*/

Величина отступа может быть независимо определена для каждой стороны элемента. Для этого используются свойства:

* padding-top – отступ сверху;
* padding-bottom – отступ снизу;
* padding-left – отступ слева;
* padding-right – отступ справа;
* margin-top – поле сверху;
* margin-bottom – поле снизу;
* margin-left – поле слева;
* margin-right – поле справа.



Рис. 83 Отличие между свойствами padding и margin

Также можно задать поля или отступы для четырёх сторон, используя сокращенную запись margin и padding.

**Синтаксис CSS**

padding: 2px 4px 5px 10px; /\*отступ сверху– 2px, справа – 4px, снизу – 5px, слева - 10px \*/

padding: 3px 6px 9px; /\*отступ сверху– 3px, справа и слева – 6px, снизу – 9px\*/

padding: 6px 12px; /\*отступ сверху и снизу – 6px, справа и слева – 12px\*/

Эти же правила относятся и к свойству margin.