

## Лабораторная работа №2

**CSS (Cascading Style Sheets)** – Каскадные таблицы стилей – это свод стилевых описаний, тех или иных HTML тегов (далее элементов HTML), который может быть применён как к отдельному тегу – элементу, так и одновременно ко всем идентичным элементам на всех страницах сайта.

CSS по сути своего рода дополнение к HTML, которое значительно расширяет его возможности.

HTML в чистом виде имеет весьма ограниченный набор инструментов, не позволяющий решать те или иные дизайнерские и функциональные задумки веб-мастера. "Как убрать подчеркивание ссылки?" или "Как сделать, чтобы, наведя курсором на эту самую ссылку, она меняла цвет и подсвечивалась?" – с помощью одного HTML этого никак не сделать. Тут и приходит на помощь CSS, который решает большинство задач, касающихся дизайна сайта.

Предположим, Вы написали сайт, в нем 100 страниц. И вдруг по каким либо причинам Вам понадобилось изменить его дизайн. Сколько времени и сил у Вас уйдёт на то что бы полностью переделать все 100 страниц сайта? CSS предлагает разумное решение этой задачи: один раз в отдельном файле полностью описать весь дизайн сайта. Допустим: все заголовки **<h1>** делать красным цветом, параграфы **<p>** писать курсивом, ссылки **<a>** не подчёркивать и т.д. Затем просто заставить эти 100 страниц HTML обращаться к файлу CSS и черпать из него нужную информацию? Теперь для изменения дизайна Вам не нужно открывать все 100 страниц, находить в них теги **<h1>** и указывать в каждом новый формат. Достаточно всего лишь открыть файл-описание и изменить описание элемента **<h1>**.

Ввиду того, что CSS позволяет выносить повторяющиеся стилевые описания одних и тех же элементов в один файл, происходит значительная "разгрузка" документов HTML, а это экономия объема, трафика, времени, денег. HTML-код становится лёгким, удобным для чтения и редакции.

### Внедрение CSS в HTML документ.

В этой главе речь пойдет о том, как внедрить CSS в документ HTML, то есть связать стилевое описание элемента непосредственно с самим элементом, каким либо HTML тегом.

Осуществить данную задачу можно тремя способами:

1. Написать стилевое описание непосредственно в самом элементе. Такой способ хорош лишь в том случае, если таковой элемент – единственный в HTML-документе, который нуждается в отдельном стилевом описании.
2. Написать стилевое описание для всех идентичных элементов HTML-документа. Такой способ оправдывает себя, если стиль страницы принципиально отличается от общего дизайна сайта (группы взаимосвязанных страниц).
3. Вынести стилевое описание элементов HTML в отдельный файл CSS. Это позволит управлять дизайном всего сайта целиком, каждой страницей сайта, в которой указано обращение к CSS-файлу. Этот способ является наиболее эффективным использованием таблицы каскадных стилей.

### Атрибут style.

Практически каждый HTML-тег имеет атрибут style, который говорит о том, что к этому тегу применяется некое стилевое описание.

Пишется так:

`<p style=""> это параграф с индивидуальным стилем </p>`

Всё что будет написано между кавычками атрибута style и будет являться стилевым описанием для данного элемента, в данном случае элемента `<p>`

Например:

`<p style="color: #ff0000; font-size:12px"> это параграф с индивидуальным стилем</p>`

В данном случае мы указали, что этот параграф должен отображаться красным цветом и иметь размер шрифта в 12 пикселей. По такому же принципу можно указать индивидуальный стиль практически для каждого HTML элемента.

Пример:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>Ампубум style</title>
```

```
</head>
```

```
<body style="background-color: #c5ffa0">
```

```
<h1 style="color: #0000ff; font-size:18px">Всё о слонах</h1>
```

```
<p style="color: #ff0000; font-size:14px">На этом сайте Вы найдёте любую  
информацию о слонах.</p>
```

```
<h2 style="color: #0000ff; font-size:16px">Купить слона</h2>
```

```
<p style="color: #ff0000; font-size:14px">У нас Вы можете по выгодным ценам  
приобрести лучших слонов!!</p>
```

```
<h2 style="color: #0000ff; font-size:16px">Взять слона на прокат</h2>
```

```
<p style="color: #ff0000; font-size:14px">Только у нас Вы можете взять любых слонов  
на прокат!!</p>
```

```
</body>
```

```
</html>
```

Но такой способ внедрения CSS хорош лишь в том случае, если требуется задать определенный стиль малому числу HTML элементов.

### Тег `<style>`

Для того что бы описать необходимые элементы одновременно на всей странице в заголовке HTML документа, внедряют тег `<style>` `</style>` (не путайте с одноименным атрибутом) в котором и происходит описание нужных нам элементов.

Пример:

```
<!DOCTYPE HTML >
```

```
<html>
```

```
<head>
```

```

<title>Тее style</title>
<style type="text/css">
body {background-color: #c5ffa0} h1 {color: #0000ff; font-size:18px} h2 {color: #0000ff;
font-size:16px} p {color: #ff0000; font-size:14px}
</style>
</head>
<body>
<h1>Всё о слонах</h1>
<p>На этом сайте Вы найдёте любую информацию о слонах.</p>
<h2>Купить слона</h2>
<p>У нас Вы можете по выгодным ценам приобрести лучших слонов!!</p>
<h2>Взять слона на прокат</h2>
<p>Только у нас Вы можете взять любых слонов на прокат!!</p>
</body>
</html>

```

Как видно из примера, мы добились точно такого же результата, что и в первом случае, только теперь мы не прописываем каждому элементу стиль индивидуально, а вынесли его в "голову" документа, указав, что все заголовки <h1>, <h2> будут синими, а параграфы <p> – красными. Сам документ стал меньше весить за счет "удаления" всех повторяющихся стилевых описаний для каждого отдельно взятого элемента.

Тег <style> принято внедрять в заголовок HTML документа между тегами <head></head>.

Атрибут тега <style> type сообщает браузеру, какой синтаксис использовать для правильной интерпретации стилей. Для правильной интерпретации браузерами CSS значение type (MIME тип данных) должно равняться text/css.

Внутри тега <style> </style> идет непосредственное объявление стилей тех или иных HTML элементов согласно следующему синтаксису:



Если в блоке объявления стилей указывается несколько свойств элемента, то они между собой разделяются точкой с запятой.

## CSS в отдельном внешнем файле.

Итак, подошли мы к главному, достоинству CSS, а именно возможности выносить все сведения касающиеся дизайна сайта в отдельный внешний файл.

Открываем блокнот (или другой редактор) и пишем в нем следующий текст:

```
body {background-color: #c5ffa0}
a {color:#000060; font-weight: bold;}
a:hover {color:#ff0000; font-weight: bold; text-decoration:none} h1 {color: #0000ff; font-size:18px}
h2 {color: #ff00ff; font-size:16px} p {color: #600000; font-size:14px}
```

Далее сохраняем этот файл с расширением \*.css (обычно файл со стилями называют style.css).

Файл со стилевым описанием создан.! Теперь осталось заставить нужные страницы нашего сайта черпать информацию с этого файла.

Делается это с помощью тега <link> (связь). Тег <link> – многоцелевой, и служит для "связывания" HTML-документа с дополнительными внешними файлами, обеспечивающими его должную работу. Тег <link> является своего рода ссылкой, только предназначенной не для пользователей, а для программ обозревателей (браузеров). Так как <link> несёт в себе исключительно служебную информацию, он располагается в заголовке HTML документа между тегами <head></head> и не выводится браузерами на экран.

Тег <link> имеет атрибуты:

- href – Путь к файлу.
- rel – Определяет отношения между текущим документом и файлом, на который делается ссылка.
- shortcut icon – Определяет, что подключаемый файл является иконкой.
- stylesheet – Определяет, что подключаемый файл содержит таблицу стилей.
- application/rss+xml – Файл в формате XML для описания ленты новостей.
- type – MIME тип данных подключаемого файла.

Так как мы подключаем в качестве внешнего файла каскадную таблицу стилей, то служебная ссылка приобретает следующий вид:

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

Атрибуту rel присваиваем значение stylesheet, так как подключаем в качестве внешнего файла каскадную таблицу стилей, указываем путь к файлу css (в этом примере файл называется mystyle.css и лежит рядом с документом HTML, в котором прописывается данная ссылка), так же указываем, что данный файл текстовый и содержит в себе стилевое описание type="text/css".

Теперь вставляем эту строчку в заголовки страниц нашего сайта.

Пример:

### Файл mystyle.css

```
body {background-color: #c5ffa0}
a {color:#000060; font-weight: bold;}
```

a:hover {color:#ff0000; font-weight: bold; text-decoration:none} h1 {color: #0000ff; font-size:18px}

h2 {color: #ff00ff; font-size:16px} p {color: #600000; font-size:14px}

#### Файл index.html

```
<!DOCTYPE HTML >
<html>
<head>
<title>каскадная таблица стилей</title>
<link rel="stylesheet" href="mystyle.css" type="text/css">
</head>
<body>
<h2>Меню:</h2>
<a href="index.html">Всё о слонах.</a>
<a href="elephant.html">Купить слона.</a>
<a href="elephant1.html">Взять слона на прокат.</a>
<hr>
<h1>Всё о слонах</h1>
<p>На этом сайте Вы найдёте любую информацию о слонах.</p>
</body>
</html>
```

#### Файл elephant.html

```
<!DOCTYPE HTML >
<html>
<head>
<title>каскадная таблица стилей</title>
<link rel="stylesheet" href="mystyle.css" type="text/css">
</head>
<body>
<h2>Меню:</h2>
<a href="index.html">Всё о слонах.</a>
<a href="elephant.html">Купить слона.</a>
<a href="elephant1.html">Взять слона на прокат.</a>
<hr>
<h1>Купить слона</h1>
<p>У нас Вы можете по выгодным ценам приобрести лучших слонов!!</p>
</body>
```

</html>

**Файл elephant1.html**

<!DOCTYPE HTML >

<html>

<head>

<title>каскадная таблица стилей</title>

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

</head>

<body>

<h2>Меню:</h2>

<a href="index.html">Всё о слонах.</a>

<a href="elephant.html">Купить слона.</a>

<a href="elephant1.html">Взять слона на прокат.</a>

<hr>

<h1>Взять слона на прокат</h1>

<p>Только у нас Вы можете взять любых слонов на прокат!!</p>

</body>

</html>

В примере выше, "сайт о слонах", на данный момент, имеет три страницы, каждая из которых связана с одним единственным внешним css-файлом – mystyle.css. Таким образом, мы значительно его "разгрузили" и сделали дизайн всего сайта "мобильным".

Используйте атрибут style для какого либо элемента, если этот элемент с отличным от других элементов стилем один-единственный на всём сайте.

Используйте тег <style> со стилевым описанием, в том случае, если страница должна иметь индивидуальный дизайн, в корне отличный от других страниц сайта.

В большинстве случаев разумно выносить каскадную таблицу стилей в отдельный css-файл.

### **Свойства текста. Выравнивание текста.**

Для того, что бы выровнять текст, например по центру экрана, мы применяли к тегу, содержащему в себе текст, атрибут align (выравнивание) и одно из его возможных значений center (по центру) Запись имела такой вид:

<p align="center">текст по центру</p>

В CSS данную задачу берет на себя свойство text-align, которое выравнивает текстовое содержание относительно элемента родителя (например, блока div) или же окна браузера.

text-align (так же как и html-атрибут align) имеет следующие значения:

- left - Выровнять текст по левому краю элемента (по умолчанию).
- right - Выровнять текст по правому краю.
- center - Выровнять текст по центру.

- justify - Выровнять текст по обоим краям.

Теперь для того чтобы выровнять текст того же параграфа по центру следует писать так:

```
<p style="text-align: center">текст по центру </p>
```

– это в этом случае, если мы с помощью атрибута style внедряем CSS непосредственно в HTML тег. В примере ниже используется тег <style> в заголовке документа:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>Выравнивание текста</title>
```

```
<style type="text/css"> h1 { text-align: center } p { text-align: justify }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Всё о слонах</h1>
```

```
<p>Слон – самое крупное млекопитающее на нашей планете! Самый большой слон из когда-либо живущих на Земле был зарегистрирован в Анголе в 1956 году. Этот самец весил около 12 тон, а в высоту достигал 4,2 метра, что на метр выше среднего Африканского слона.</p>
```

```
<p>Слоны являются королевским символом Азиатской культуры и известны своей отличной памятью и высоким интеллектом. Аристотель однажды сказал, что слон – "животное, которое превосходит всех других в остроумии и интеллекте".</p>
```

```
</body>
```

```
</html>
```

## Оформление текста.

Свойство text-decoration позволяет декорировать текст, присвоив ему одно или несколько значений из ниже представленных вариантов оформления текста.

Возможные значения:

- blink - Текст будет мигать.
- line-through - Делает текст перечеркнутым.
- overline - Надчёркивание текста.
- underline - Подчеркивание текста.
- none - Текст без оформления.

Пример:

```
<a href="index.html" style="text-decoration:none">Ссылка без подчёркивания</a>
```

### Пример:

#### Файл mystyle.css

```
h1 {text-align: center}
h3 {text-align: left; text-decoration: underline} a {text-decoration: underline}
a:hover {text-decoration:none} p {text-align: justify}
```

#### Файл index.html

```
<!DOCTYPE HTML>
<html>
<head>
<title>Оформление текста</title>
<link rel="stylesheet" href="mystyle.css" type="text/css">
</head>
<body>
<h3>Меню:</h3>
<a href="index.html">Всё о слонах.</a><br>
<a href="elephant.html">Купить слона.</a>
<hr>
<h1>Всё о слонах</h1>
<p> Слон - самое крупное ... ..</p>
<p>Слоны являются ... ..</p>
</body>
</html>
```

Обратите внимание на внешний файл CSS: в нем мы "декорировали" ссылку – элемент `<a>`, причем делали это два раза: первый раз `a {text-decoration: underline}` сделали её подчеркнутой, хотя можно было этого и не делать, так как тег `<a>` подчеркнут по умолчанию, а второй раз использовали так называемый псевдокласс `hover` и запретили подчеркивание `a:hover {text-decoration:none}`.

Данный псевдокласс указывает на то, что применять к нему стилевое описание стоит лишь в том случае, если пользователь навел курсор на этот элемент. Так, если в примере навести курсор на одну из ссылок в меню, то подчеркивание исчезнет, что создаёт определенный динамический эффект.

### Отступ первой строки.

Свойство `text-indent` – задаёт отступ первой строки в текстовом блоке с левой стороны, делает "красную строку".

Расстояние от левого края окна браузера или же элемента родителя (блока, в который помещен блок с текстом) может быть задано в процентах от ширины окна браузера или же единицах измерения, принятых в CSS.



В примере ниже расстояние отступа от левого края задаётся в пикселях (px):

```
<!DOCTYPE HTML >
<html>
<head>
<title>Красная строка</title>
<style type="text/css"> h1 {text-align: center}
p {text-align: justify; text-indent: 20px}
</style>
</head>
<body>
<h1>Всё о слонах</h1>
<p>Слон - самое крупное млекопитающее на нашей планете! Самый большой слон
из когда- либо живущих на Земле был зарегистрирован в Анголе в 1956 году.
Этот самец весил около 12 тон, а в высоту достигал 4,2 метра, что на метр
выше среднего Африканского слона.</p>
<p>Слоны являются королевским символом Азиатской культуры и известны
своей отличной памятью и высоким интеллектом. Аристотель однажды сказал,
что слон - "животное, которое превосходит всех других в остроумии и
интеллекте".</p>
</body>
</html>
```

## Трансформация текста

Свойство text-transform трансформирует символы в указанном текстовом блоке, делая их заглавными или прописными по одному из правил в зависимости от присужденного значения данному свойству.

Значения:

- none - Текст отображается без каких-либо изменений.(по умолчанию)
- capitalize - Каждое слово в тексте отображается с заглавного символа.
- lowercase - Все символы преобразуются в нижний регистр.
- uppercase - Все символы преобразуются в верхний регистр.

Пример:

```
<!DOCTYPE HTML >
<html>
<head>
<title>Трансформация текста</title>
</head>
<body>
```

```

<p style="text-transform: capitalize">союз советских социалистических республик</p>
<p style="text-transform: lowercase">СССР ссср</p>
<p style="text-transform: uppercase">ссср СССР</p>
</body>
</html>

```

## Вертикальное выравнивание

Вертикальное выравнивание текста в строке устанавливает свойство vertical-align. Возможные значения свойства vertical-align:

- baseline - Выравнивает базовую линию элемента по базовой линии родителя.
- bottom - Выравнивает элемент по нижней части строки.
- middle - Выравнивает середину элемента по базовой линии родителя и прибавляет половину высоты родительского элемента.
- sub - Нижний индекс (размер шрифта не меняется).
- super - Верхний индекс (размер шрифта не меняется).
- text-bottom - Нижняя граница элемента выравнивается по нижнему краю строки.
- text-top - Верхняя граница элемента выравнивается по верхнему краю строки.
- top - Выравнивает элемент по верхней части строки.

Базовая линия – это линия, на которой располагаются символы в текстовой строке.

Взгляните на рисунок с разметкой строки:



Так же вертикальное выравнивание элемента относительно строки может выражаться в процентах, пикселях или любых других принятых в CSS единицах измерения, причем эти единицы могут принимать как положительные, так и отрицательные значения.

Пример:

```

<!DOCTYPE HTML>
<html>
<head>
<title>Вертикальное выравнивание текста</title>
</head>
<body>
<font size="+3">А и Б </font>
<span style="vertical-align: +5px">сидели на трубе </span>
<span style="vertical-align: bottom">А упало </span>
<span style="vertical-align: top">Б пропало.. </span>

```

```

<span style="vertical-align: 50%">что осталось на трубе?</span>
<hr>
формула воды: H<span style="vertical-align: sub">2</span>O
<hr>
<span>H</span>
<span style="vertical-align: -10px">a</span>
<span style="vertical-align: -20px">u</span>
<span style="vertical-align: -30px">c</span>
<span style="vertical-align: -40px">k</span>
<span style="vertical-align: -50px">o</span>
<span style="vertical-align: -60px">c</span>
<span style="vertical-align: -70px">o</span>
<span style="vertical-align: -80px">k</span>
</body>
</html>

```

### Пробелы и перенос строки.

Набранный текст в каком либо текстовом редакторе браузерами по умолчанию выводится на экран в виде сплошного текста, где переносы строк расставляются автоматически, а так же убираются лишние (более одного) пробелы между символами.

Свойство white-space имитирует работу тега <pre>, определяя показывать или нет пробелы между символов, если таковых больше чем один, а так же разрешает или запрещает перенос строки.

Может иметь следующие значения:

- normal - текст выводится как обычно (лишние пробелы убираются), переносы строк определяются автоматически. (по умолчанию)
- nowrap - запрещает автоматический перенос строки.
- pre - показывает текст в том виде в котором он был набран. пробелы и переносы строки не удаляются.

Пример:

```

<!DOCTYPE HTML >
<html>
<head>
<title>Пробелы и перенос строки</title>
</head>
<body>
<p style="white-space: pre"> Слон.
Дали туфельки слону.
Взял он туфельку одну

```

*И сказал: - Нужны пошире, И не две, а все четыре!*

*С. Я. Маршак.*

`</p>`

`<hr>`

`<p style="white-space: nowrap">`

*Это длинный предлинный текст, который вряд ли полностью поместится в одной строчке, по умолчанию в нужном месте, браузер перенес бы его на следующую строку, однако мы принудительно запретили это делать, с помощью значения nowrap свойства white-space. Так что теперь, по всей вероятности, в окне браузера появится горизонтальная полоса прокрутки.. и зачем, спрашивается, мы это сделали?*

`</p>`

`</body>`

`</html>`

При использовании nowrap текст в нужном месте можно переносить на следующую строку, используя тег `<br>`

### **Расстояние между словами.**

Свойство word-spacing задаёт расстояние между словами (группами символов, не разделёнными пробелом) в строке.

Значения:

- normal – Нормальное расстояние (по умолчанию);
- px – Расстояние задаётся в пикселях или любых других единицах измерения, принятых в CSS.

Пример:

`<!DOCTYPE HTML >`

`<html>`

`<head>`

`<title>Расстояние между словами</title>`

`</head>`

`<body>`

`<p align="left" style="word-spacing: 10px">Расстояние между словами равно десяти пикселям</p>`

`<p align="left" style="word-spacing: -10px">Расстояние между словами может иметь отрицательное значение</p>`

`</body>`

`</html>`

## Межсимвольное расстояние.

Свойство letter-spacing определяет расстояние между символами в тексте и так же может быть задано следующими значениями:

- normal – Нормальное расстояние (по умолчанию);
- px – Расстояние задаётся в пикселях или любых других единицах измерения, принятых в CSS.

Пример:

```
<!DOCTYPE HTML >
<html>
<head>
<title>Расстояние между символами</title>
</head>
<body>
<p style="letter-spacing: 5px">Расстояние между буквами равно пяти пикселям</p>
<p style="letter-spacing: -3px">А здесь буквы, из за отрицательного значения, будут наплывать друг на друга</p>
</body>
</html>
```

## Интерлиньяж

Интерлиньяж - это расстояние между строками текста.

Расстояние между строками текста можно задать, используя свойство line-height, сделать это можно следующими способами:

- normal – Норма (по умолчанию).
- % – Проценты, за 100 процентов берется высота шрифта.
- 0.5 – Множитель. Может быть использовано любое число больше нуля. Так, например, множитель 0.5 будет соответствовать половинному межстрочному расстоянию, а 2 – двойному.
- px – Пиксели и любые другие единицы измерения, принятые в CSS.

Пример:

```
<!DOCTYPE HTML >
<html>
<head>
<title>Интерлиньяж</title>
</head>
<body>
<div style="line-height: 150%">
строка первая <br> строка вторая <br> строка третья <br> строка четвертая
```

```
<br> строка пятая
</div>
<hr>
<div style="line-height: 0.5">
строка первая <br> строка вторая <br> строка третья <br> строка четвертая
<br> строка пятая
</div>
<hr>
<div style="line-height: 25px">
строка первая <br> строка вторая <br> строка третья <br> строка четвертая
<br> строка пятая
</div>
</body>
</html>
```

### **Задание на лабораторную работу**

Сверстать веб-сайт, состоящий не менее, чем из трёх страниц, между которыми организован переход по ссылкам.

Обязательно использование:

- Стиля, встроенного в страницу1;
- Стиля, вынесенного в отдельный файл, для страницы2;
- Интерлиньяжа;
- Межсимвольного расстояния;
- Расстояния между словами;
- Пробелов и переносов строки;
- Вертикального выравнивания;
- Трансформации текста;
- Отступа первой строки;
- Оформления текста;
- Выравнивания текста.