

ЛАБОРАТОРНАЯ РАБОТА №3

CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) - Каскадные таблицы стилей - это свод стиливых 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.

Приведу ряд доводов в пользу использования CSS:

HTML в чистом виде имеет весьма ограниченный набор инструментов не позволяющий решать те или иные дизайнерские и функциональные задумки веб-мастера. Ну вот хотя бы, к примеру, взять больной вопрос всех начинающих веб-ремесленников "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!! А сколько их еще таких "больных вопросов"? - тьма.. Тут то и приходит на помощь CSS, который решает большинство задач касающихся дизайна сайта.

Предположим, Вы написали сайт в нем 100 страниц.. хороший сайт, информативный, люди на него ходят.. И вдруг по каким либо причинам Вам понадобилось изменить его дизайн, ну не знаю, мода изменилась, Вы нашли более лучшее дизайнерское решение, заказчику пришлось не по душе.. да мало ли еще почему.. Сколько времени и сил у Вас уйдёт на то что бы полностью переделать все 100 страниц сайта? CSS предлагает разумное решение этой задачи. А что если 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 :) фон на всех страницах залить зелёным, и т. д. ... а потом просто заставить эти 100 страниц HTML обращаться к файлу CSS и черпать из него нужную информацию? Теперь, когда Вам вздумается, к примеру, перекрасить все заголовки из красного в зеленый, Вам ненужно открывать все 100 страниц находить в них теги <h1> и указывать в каждом что ты теперь не красный, а зелёный! Вам нужно всего лишь открыть файл описание и изменить в нем цвет элемента <h1> на зелёный и всё!! Все заголовки на всех страницах сайта как по взмаху волшебной палочки станут зелеными.

Ввиду того, что CSS позволяет выносить повторяющиеся стиливые 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.

Речь пойдет о том, как внедрить CSS в документ HTML, то есть связать стиливое описание элемента непосредственно с самим элементом, каким либо HTML тегом.

Осуществить данную задачу можно *тремя* способами:

- Написать стиливое описание непосредственно в самом элементе. Такой способ хорош лишь в том случае если таковой элемент один единственный в HTML документе который нуждается в отдельном стиливом описании.
- Написать стиливое описание для всех идентичных элементов HTML документа. Такой способ оправдывает себя, если стиль страницы принципиально отличается от общего дизайна сайта (группы взаимосвязанных страниц).
- Вынести стиливое 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.

Давайте более подробно рассмотрим каждый вариант, а заодно познакомимся с правилами синтаксиса написания CSS.

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. Сейчас речь идет о том, как применить CSS к какому либо HTML тегу.

По такому же принципу можно указать индивидуальный стиль практически для каждого HTML элемента.

Пример 1:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Атрибут 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 еще раз повторюсь, 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.

Взгляните на *пример 2* ниже (к нему будут комментарии).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
<title>Ter 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 тем самым указав, что все заголовки <h1>,<h2> - будут синими, а параграфы <p> - красными. Представьте, как мы облегчили бы себе работу, будь на странице сотня таких параграфов и штук пятнадцать заголовков, да и сам документ стал меньше весить за счет "удаления" всех повторяющихся стилевых описаний для каждого отдельно взятого элемента.

Теперь обещанные комментарии:

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:

p {color: #ff0000; font-size: 14px}

Селектор (HTML элемент)

Блок объявления стилей в фигурных скобках

Свойство CSS

Значение свойства CSS

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 небольшой файл с расширением *.css (обычно файл со стилями называют style.css).

Все! файл со стиливым описанием создан! Теперь осталось совсем чуть чуть, а именно 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 наша служебная ссылка приобретает следующий вид:

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

Повторюсь, что бы уж точно развеять возможные недопонимания. 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 и наслаждаемся результатом..

Пример 3:

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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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, имеется 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". Представьте сколько б килобайт мы выиграли, будь на этом сайте сотня полноценных страниц!? А также, сколько б времени сэкономили, если бы нам понадобилось изменить что-либо в его дизайне!?

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.

Если Вы помните, из курса HTML, 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 htmlловский 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.

А вот в *примере 4* ниже используется тег <style> в заголовке документа:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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 когдалибо живущих на Земле был зарегистрирован в Анголе в 1956 году. Этот самец
```


весил около 12 тонн, 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.

Пишется так:

Ссылка без подчёркивания

Пример 5:

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 PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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 а {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 - меню становится "живым".

Отступ первой строки.

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

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

В *примере 6* ниже расстояние отступа от левого края задаётся в пикселях (px):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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 когдалибо 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 7:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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 "сидят" символы в текстовой строке, Например буква "А" сидит прямо на этой линии, а вот строчная буква "у" сидит на ней же, но свесив ноги.

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 8:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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>н</span>
<span style="vertical-align: -10px">а</span>
<span style="vertical-align: -20px">и</span>
<span style="vertical-align: -30px">с</span>
<span style="vertical-align: -40px">к</span>
<span style="vertical-align: -50px">о</span>
<span style="vertical-align: -60px">с</span>
<span style="vertical-align: -70px">о</span>
<span style="vertical-align: -80px">к</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 9:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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 появиться
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

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 10:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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.

А вот свойство letter-spacing определяет расстояние между символами в тексте и так же как и может word-spacing быть задано следующими значениями:

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

Пример 11:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<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 сто процентов берется высота шрифта
- ✓ 0.5 - Множитель. Может быть использовано любое число больше нуля. Так, например множитель 0.5 будет равняться половинному межстрочному расстоянию, а 2 - двойному.
- ✓ px - Пиксели и любые другие единицы измерения, принятые в CSS.

Пример 12:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<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

Задание 1.

Самостоятельно на практике проработать все примеры из лабораторной работы.
Включить скрины в отчет.

Задание 2.

Самостоятельно сверстать веб-сайт, состоящий из минимум двух страниц, между которыми возможен переход по ссылкам.

Обязательно использование:

- ✓ Стиля, встроенного в страницу1;
- ✓ Стиля, вынесенного в отдельный файл, для страницы 2;
- ✓ Интерлиньяжа
- ✓ Межсимвольного расстояния.
- ✓ Расстояния между словами.
- ✓ Пробелов и переносов строки.
- ✓ Вертикального выравнивания
- ✓ Трансформации текста
- ✓ Отступа первой строки.
- ✓ Оформления текста.
- ✓ Выравнивания текста.

Задания по вариантам: (вариант – номер по журналу)

- | | |
|----------------|--------------------------|
| 1. Компьютеры | 16. Мобильная связь |
| 2. Интеллект | 17. Авиация/космонавтика |
| 3. Учеба | 18. Медицина |
| 4. Игрушки | 19. Ремонт в быту |
| 5. Одежда | 20. Праздники |
| 6. Туризм | 21. Животные |
| 7. Погода | 22. Сказки |
| 8. Мебель | 23. Растения |
| 9. Книги | 24. Транспорт |
| 10. Музыка | 25. Интернет |
| 11. Автомобили | 26. Финансы |
| 12. Политика | 27. Человек |
| 13. Религия | 28. География |
| 14. Красота | 29. Пища |
| 15. История | 30. Архитектура |

Отчет Должен содержать:

- ✓ Титульный лист, номер и наименование лабораторной работы;
- ✓ задание на лабораторную работу;
- ✓ описание использованных тегов со скриншотами результатов;
- ✓ выводы.