Практическое занятие №3 CSS

Задание №1

Проверьте запущен ли у Вас Open Server. Папка с материалами урока (css3.loc) должна быть размещена в директории /domains сервера. Заходим на http://css3.loc и начинаем урок.

Задание №2

На прошлых уроках Вы изучили базовые структуры HTML. Сейчас мы начнём изучать как можно изменить внешний вид этих элементов.

Откройте данное задание в браузере и убедитесь, что в файле задания уже есть некоторая вёрстка.

Теперь мы в эту вёрстку подключим файл со стилями CSS. Для этого нужно в секции <HEAD> разместить тег **<link>** как показано на рисунке ниже.

```
<link href="/tasks/task-2/style.css" type="text/css" rel="stylesheet">
```

Перезагрузите страницу и посмотрите, что изменилось на странице.

Этот тег очень похож на тег ссылки (<a>). Разберём его атрибуты:

- ***href*** – указывает путь к подключаемому файлу. Это может быть путь от корня сайта, как у нас в примере, так и полный путь с указанием домена. Можно подключать файлы стилей с с других сайтов.
- ***type*** - MIME-тип данных подключаемого файла. Для стилей «text/css»
- ***rel*** - Определяет отношения между текущим документом и файлом, на который делается ссылка.

У этого тега есть масса других параметров, но сейчас для нас они не так важны.

Задание №3

Использование отдельных файлов для хранения стилей CSS является хорошим тоном. Но иногда возникают ситуации, когда нужно разместить CSS код прямо на HTML странице. Для этих целей используется тег <style>. По стандарту этот тег размещается в секции <head>. Раскомментируйте тег <style> в файле задания данного урока. Проследите что изменилось после этого?

Создайте в папке с заданием файл style.css. Теперь перенесите содержание блока <style> в этот файл и подключите его на странице через тег link.

Задание №4

Для того что бы изменять внешний вид HTML элементов нужно сначала их выбрать т. е. показать браузеру с какими элементами мы хотим работать. Конструкции, которые помогают нам делать выборку HTML элементов называются селекторами. С самыми базовыми селекторами мы сейчас и познакомимся.

Работаем в файле style.css, который находится в папке задания.

Селектор тега выбирает все советующие теги на странице. Для того что бы создать селектор тега нужно просто указать название тега, без угловых скобок. Ниже находится код выбора всех параграфов, наберите его у себя в файле style.css.

```
p {

}
```

Сейчас мы выбрали все параграфы на странице, но их вид не изменился потому что мы не прописали им никаких свойств. Добавим свойство color, который определяет цвет текста.

```
p {
    color: red;
}
```

Отметьте для себя изменения, которые произошли на странице. Теперь самостоятельно добавьте тегу <h1> свойство «color: green;», а тегам <li> - «color: purple;».

Если мы хотим выбрать несколько тегов, мы можем их просто перечислить через запятую. Такое поведение характерно для всех видов селекторов. В качестве примера установим шрифт (font-family) сразу для параграфов и списков.

```
p, li {
    font-family: Verdana;
}
```

Когда перечислять все теги утомительно, на помощь приходит универсальный селектор (*), который выбирает вообще все элементы страницы, попробуем его на практике. Установим всем элементам страницы единый размер шрифта (font-size)

```
* {
    font-size: 20px;
}
```

Задание №5

Идентификатор (называемый также «ID селектор») определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты. По стандарту идентификатор должен быть уникальным. Создадим идентификатор у первого абзаца в файле задания.

```
<p id="first_p">The brown bear
```

Теперь в файле style.css прописываем стиль для данного идентификатора, как на изображении ниже.

```
#first_p {
    font-size: 24px;
    color: yellowgreen;
}
```

Обратите внимание, что изменения коснулись только элемента с идентификатором.

Классы используются для того что бы применить стили для нескольких различных элементов. На примере рассмотрим, как это работает. Создадим класс для второго абзаца и ссылки под ним как на картинке ниже.

```
<a class="redtext" href="https://en.wikipedia.org/wik
<p class="redtext" >Here are some bear species:</p>
```

Теперь прописываем стиль для класса, как на картинке ниже.

```
.redtext {
    color: red;
}
```

Обратите внимание что изменения коснулись всех элементов с этим классом, хотя они имеют разные теги.

В css предусмотрена возможность комментировать код, для этого используется конструкция /* */. Всё что находится внутри, считается комментарием.

Закомментируйте универсальный селектор в файле css, как показано ниже.

```
/*
* {
    font-family: Verdana;
}
*/
```

Как вы можете заметить закомментированный селектор перестал оказывать влияние на страницу.

Задание №6

В прошлых заданиях Вы ознакомились с базовыми селекторами. Сейчас Вы должны понимать, что один и тот же элемент на странице может быть выбран одновременно несколькими разными селекторами. Рассмотрим пример выборки в файле style.css этого задания.

Как Вы можете видеть все три выборки выбирают кроме всего прочего элемент <h1>. Вопрос что будет если начать задавать различные значения стилям в этих выборках, исследуем этот вопрос.

Зададим всем заголовкам целеный цвет:

```
h1, h2 {
    color: green;
}
```

В результате все три заголовка стали зелеными. Идем дальше. Добавим еще одно правило color там же, но с другим значением.

```
h1, h2 {
    color: green;
    color: greenyellow;
}
```

Что произошло? Мы можем сделать вывод, если у правил одинаковый «вес», то будет применяться то, что объявлено ниже. Отделите второй правило color в отдельный селектор, как показано на рисунке.

```
h1, h2 {
    color: green;

}

h1, h2 {
    color: greenyellow;
}
```

Заголовки остались сине-жёлтого цвета потому что этот цвет определяется ниже в таком же селекторе. Поменяйте местами эти селекторы, так что бы заголовки стали зеленого цвета.

Следующей особенностью CSS, которую мы должны изучить будет приоритет правил (специфичность). Каждый вид селектора имеет свой «вес». У селектора классов «.class» вес будет

всегда выше чем у селектора тегов «h1, h2». Поэтому где бы не находился селектор класса он будет приоритетнее. Проверим это.

Зададим пурпурный цвет всем элементам класса «.header» , как на рисунке ниже.

```
.header {
    color: purple;
}
```

Следующим шагом мы переносим селектор класса «.header» в начало файла. Наблюдаем что произойдёт. Пурпурный цвет должен оставаться.

Селектор идентификатора «#id» имеет еще больший вес чем селектор класса «.class». Перекрашиваем наш главный заголовок в красный цвет.

```
#main_header {
    color: red;
}
```

Заголовок <H1> должен стать красного цвета.

Немного информации по весам. Если определить вес селекторов через числа, мы получим приблизительно следующую таблицу:

Селектор тега:	1
Селектор класса:	10
Селектор ID:	100
Inline-стиль:	1000

Примеры:

Селектор	ID	Класс	Тег	Общий вес
p	0	0	1	1
.your_class	0	1	0	10
p.your_class	0	1	1	11
#your_id	1	0	0	100
#your_id p	1	0	1	101
#your_id .your_class	1	1	0	110
p a	0	0	2	2
#your_id #my_id .your_class p a	2	1	2	212

Задание №7

Говоря о приоритете правил CSS невозможно не упомянуть про атрибут style у HTML элементов. Этот атрибут напрямую задаёт элементу HTML правила стилей. При этом они имеют больший приоритет над другими правилами, за одним исключением. Рассмотрим, как это работает.

На данный момент в файле style.css данного урока объявлен стиль «#main_header». Добавьте к элементу <h1> атрибут «style» как показано ниже.

```
<h1 id="main_header" style="font-size: 32px; color: royalblue;" class="header"
```

Как Вы видите внешний вид элемента изменился.

Теперь заставим внешние стили снова действовать на элемент, для этого нужно добавить **!important** к свойствам, как показано ниже.

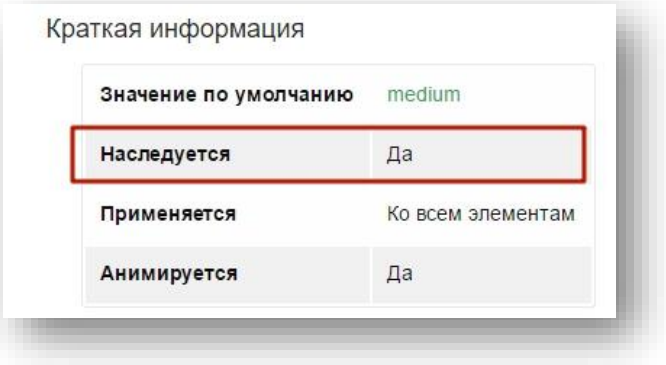
```
#main_header {
    font-size: 24px !important;
    color: brown !important;
    font-family: monospace !important;
}
```

Как Вы уже наверное догадались атрибуты отмеченные как **!important** имеют самый большой вес. Если же так случилось что у нескольких разных атрибутов есть **!important**, то будет действовать атрибут с наибольшим «весом». Добавим !important к стилям к атрибуте style, как показано на рисунке ниже:

```
<h1 id="main_header" style="font-size: 32px !important; color: royalblue !important;" class="header"
```

Задание №8

Наследование стилей – это автоматическое применение стилей родительских элементов к их потомкам. Не все свойства стилей наследуются, для того что бы узнать наследуется ли свойство нужно заглянуть в его описание. Описание всех свойств можно найти на этом сайте. Например, если мы откроем страницу с описанием свойства font-size, то увидим, что оно наследуется.



Открываем файл стилей style.css данного урока и прописываем следующие стили в селектор body.

```
body {
    font-family: Calibri;
}
```

Как вы можете видеть, что стиль шрифта поменялся на всей странице. Это означает что все потомки <body> «унаследовали» этот стиль.

Теперь добавим еще и цвет для <body> как показано ниже.

```
body {
    font-family: Calibri;
    color: orange;
}
```

Результат тот же, стили успешно наследуются элементами страницы.

Задание №9

Существует большое количество стилей, которые не наследуются и действуют только на свой «родной» HTML элемент. Что бы в этом убедиться мы будем использовать свойство border, которое добавляет рамку вокруг элемента. Откройте файл стилей style.css данного урока и добавьте указанное свойство, как показано на рисунке ниже.

```
ul {
    border: 1px solid;
}
```

Как Вы можете видеть рамка возникла только у элемента <ul> если бы свойство border было бы наследуемым, то все элементы <li> внутри <ul> тоже получили отдельные рамки.

Задание №10

Иногда возникает необходимость выбрать не сам элемент, а его потомков. В этом случае необходимо указать последовательность селекторов. Откройте файл стилей style.css данного урока и напечатайте код ниже.

```
ul li {
    color: red;
}
```

Как Вы сами заметили все элементы <li>, вложенные в <ul> стали красного цвета, в то же время элементы <li>, которые вложены в <ol> остались нетронутыми.

Добавляем стиль border, как показано ниже.

```
ul li {
    color: red;
    border: 1px solid;
}
```

Что бы показать Вам принцип действия этого селектора потомков мы добавляем еще один список вложенный первый пункт изначального списка <ul>, как показано на изображении ниже.

```
<ul>
    <li>
        <ol>
            <li>1</li>
            <li>2</li>
            <li>3</li>
        </ol>
    </li>
    <li>Collarus</li>
    <li>Horribilis</li>
    <li>Nelsoni (extinct)</li>
</ul>
```

Как Вы можете видеть, абсолютно все теги <li> красные и обрамлены рамкой, независимо от глубины вложенности.

Что бы ограничить глубину вложенности селектора нужно его изменить как показано на рисунке ниже.

```
ul > li {
    color: red;
    border: 1px solid;
}
```

Подумайте какие изменения произошли и почему?

Задания №11

Селекторы атрибутов как понятно из названия позволяют нам выбирать элементы HTML по значению атрибута в нём.

Открывайте файл style.css в папке задания. С помощью селектора атрибутов мы стилизуем рамку вокруг поля с логином в зеленый цвет, а поля с паролем в красный, как на изображении ниже.

Можно выбирать элементы, используя несколько селекторов. Например, селектор «input[attr]»

```
input[type="text"] {
    border: 1px solid green;
}

input[type="password"] {
    border: 1px solid red;
}
```

выберет тег «input» у которого присутствует атрибут «attr».

Теперь стилизуем кнопку отправки формы, как показано ниже.

```
form [type="submit"] {
    border: 1px solid orange;
    background-color: orange;
}
```

Задание №12

Мы немного разобрались с селекторами и сейчас приступаем к единицам изменения принятым в CSS. Начнем мы с цветов. В CSS используется несколько форматов обозначения цветов. С ними мы сейчас и познакомимся.

Для работы нам потребуется два важный свойства для работы с цветом:

color: <цвет> - задаёт цвет текста у элемента.

background-color: <цвет> - задаёт цвет заднего фона элемента.

Самым простым способом задать цвет является использование наименованных цветов. Полный их список можно посмотреть по ссылке http://www.colors.commutercreative.com/grid/ .

Откройте файл стилей этого задания и задайте главному заголовку на странице цвет текста BlueViolet, и цвет фона AntiqueWhite.

Система наименованных цветов имеет очень большой недостаток – количество цветов ничтожно мало. Система шестнадцатеричных кодов (***hexadecimal color codes HEX***) позволяет отобразить намного больше цветов. Перевод именованных цветов шестнадцатеричные можно посмотреть на внешнем ресурсе https://www.w3schools.com/colors/colors_names.asp

Задайте всем подзаголовкам (<h2>) цвет текста белый (#FFFFFF), а цвет фона черный(#000000). При этом в HEX если символы совпадают попарно, то второй символ в парах можно опустить, например, так

- #FF0000 - #F00
- #FFFFFF - #FFF
- #009900 - #090
- #008801 – тут нельзя упростить так как в последней паре символы не совпадают.

Система цветов **RGB (Red, Green, Blue)** очень широко используется в css. Всем абзацам пропишите цвет текста *rgb(54, 74, 101);* и цвет фона *rgb(23, 108, 224);*.

HSL (hue, saturation, lightness) — цветовая модель, в которой цветовыми координатами являются тон, насыщенность и светлота. Намного реже используется в вебе чем HEX и RGB.

Всем тегам <li> зададим цвет текста *hsl(182, 20%, 50%);* и цвет фона *hsl(350, 58.8%, 42.0%);*

Задание №13

Для того что можно было сделать цвета прозрачными, мы должны еще указывать степень прозрачности (альфа канал) от 0 до 1, где 0 означает, что цвета вообще не видно, а 1, что цвет полностью не прозрачный.

Откройте файл style.css в папке задания и поменяйте цвет заголовка H1 на *rgba(2, 93, 140, 0.6);* где, 0.6 как Вы уже поняли степень прозрачности.

Задайте всем подзаголовкам прозрачный цвет, но уже в системе цветов: *hsla(239, 45%, 22%, 0.4);*

Задание №14

Скорее всего Вы сталкивались со шрифтами в своей жизни. Вы постоянно используете разные шрифты в текстовых документах.

Свойство *font-family: <шрифт>;* устанавливает стиль шрифта для элемента HTML и его потомков. Об этом свойстве нужно знать несколько его особенностей:

1. Если название шрифта состоит из нескольких слов, то его нужно писать в кавычках, например *font-family: “Courier New”*;.
2. Если шрифт не установлен в операционной системе пользователя и не подключен на вашей HTML странице, то он не отобразиться и браузер будет использовать шрифт по умолчанию.

Откройте файл style.css в папке данного урока и измените стиль шрифта для главного заголовка и подзаголовков на «Georgia».

Теперь задаём параграфам шрифт «Roboto-Bold». Как Вы можете заметить ничего не изменилось, потому что скорее всего у Вас нет этого шрифта. Подключаем нужный нам шрифт в файле style.css следующим образом:

```
@font-face {
    font-family: "Roboto-Bold"; /* Имя шрифта */
    src: url(Roboto-Bold.ttf); /* Путь к файлу со шрифтом */
}
```

Задание №15

Google Fonts это бесплатное хранилище шрифтов. С помощью этого сервиса Вы можете очень быстро находить и подключать нужные шрифты на свои страницы.

Для того что бы подключить шрифт с сервиса Google Fonts нам необходимо:

1. Заходим на сайт Google Fonts https://fonts.google.com/ .
2. Выбираем любой понравившийся нам шрифт и заходим на его страницу.
3. На странице выбранного шрифта нажимаем кнопку «SELECT THIS FONT»
4. В правом нижнем углу страницы по клику открывается блок «Family Selected» с инструкцией как импортировать (тег <link>) и использовать (font-family) выбранный шрифт.
5. Открываем файл index-15.html в папке текущего задания и импортируем выбранный шрифт (используйте тег <link>).
6. Откройте файл style.css и примените выбранный шрифт к тегам <p>, для этого используйте свойство font-family.

Задание №16

Свойство **font-size** устанавливает размер шрифта. Для указания значения размера в CSS используется множество форматов.

Поскольку компьютерные экраны используют **пиксели** для отображения содержимого, это самая распространённая единица размера в CSS. Пиксели в CSS являются простыми, поскольку они определяют абсолютные значения и не зависят от других наследуемых свойств CSS.

Проценты — это *относительная* единица измерения. значение высчитывается относительно значения свойства родительского тега.

Em - является относительной единицей измерения и зависит от значения font-size элемента. Например, если у родителя font-size задан как 20px и вы применяете font-size: 0.8em к дочернему элементу, то этот дочерний элемент будет отображать font-size как 16px.

Существует еще множество других, менее распространённых единиц измерения.

1. Откройте файлы style.css и index-16.html в папке задания.
2. Задайте <h1> свойство font-size равное «40px»
3. Задайте <h2> свойство font-size равное «2em»
4. Задайте <body> свойств font-size равное «40px»
5. Понаблюдайте за изменениями на странице. Как Вы уже заметили размер текста тега <h1> остался неизменным, так как значение указано в пикселях, а размер <h2> изменился так как у него указано относительное значение (em).

Свойство **line-height** задаёт высоту строки текста. Обратите внимание что в теге <h2> буквы стали настолько большими что налезли друг на друга. Если мы посмотрим в стили то увидим, что свойство line-height указано в абсолютных значениях (*line-height: 24px;),* заменим их на *line-height: 1em;*. Теперь как бы Вы не изменяли размер шрифта, высота строки будет изменяться вместе с ним.

Задание №17

Свойство **font-weight** устанавливает насыщенность шрифта. Значение указывается в виде чисел от 100 до 900 с шагом 100 или с помощью заданных ключевых слов (bold | bolder | lighter | normal).

Свойство **font-style** определяет начертание шрифта — обычное (*normal*), курсивное (*italic*) или наклонное (*oblique*).

Откройте файл задания и отформатируйте текст, как показано в таблице ниже.

Элемент	**Насыщенность шрифта**	**Размер шрифта**	**Начертание шрифта**
Заголовок <h1>	*bold*	*32px*	
Первый абзац	*lighter*	*20px*	*italic*
Заголовок <h2>	*bolder*	*28px*	
Второй и все последующие абзацы		*20px*	
Последний абзац			*oblique*

Задание №18

Свойство **text-decoration** добавляет оформление текста в виде его подчёркивания (*underline*), перечёркивания (*line-through*) или линии над текстом (*overline*). Одновременно можно применить более одного стиля, перечисляя значения через пробел. Убрать все эффекты можно значением *none.*

Откройте файл style.css в папке урока. Уберите подчеркивание у всех ссылок на странице.

Сделайте так что бы «Ссылка1» была перечёркнутой.

Сделайте так что бы «Ссылка3» была подчеркнутой в обычном состоянии и при наведении на нее курсора была перечеркнутой для этой цели используйте псевдоселектор **:hover**, например *a:hover {}* .

Задание №19

Свойство **text-indent** устанавливает величину отступа первой строки блока текста, чаще всего это абзац <p>.

Откройте файлы style.css и index-19.html в папке задания. Задайте всем абзацам свойство *text-indent: 40px;*

Свойство **text-align** определяет горизонтальное выравнивание текста в пределах элемента. Может принимать следующие значения:

- *center* - выравнивание текста по центру элемента.
- *justify* – растягивание текста на всю ширину элемента
- *left* – выравнивание по левому краю (по умолчанию)
- *right* - выравнивание по правому краю (по умолчанию)

Выровняйте текст заголовка <h1> по центру. Выровняйте текст второго абзаца по правому краю.

Задание №20

Создайте страницу со своими персональными данными. Структура и вид страницы полностью на Ваше усмотрение. На странице должны быть использованы все те знания, которые Вы получили на сегодняшнем уроке (цвет текста, цвет фона, шрифты, размеры текста, выравнивание и прочее).

Домашнее задание

Скачать и установить пробную версию Adobe Photoshop с сайта http://www.adobe.com/ru/products/photoshop.html

Прочитайте пост на удалённом ресурсе https://habrahabr.ru/post/272649/

Прочитайте как работать с макетом http://tpverstak.ru/photoshop-dlya-verstaka/

Домашнее задание №2

В браузере Google Chrome установите расширение *PerfectPixel by WellDoneCode*

https://chrome.google.com/webstore/detail/perfectpixel-by-welldonec/dkaagdgjmgdmbnecmcefdhjekcoceebi?hl=ru