Вторая лекция
java for web
HTML & CSS

HTML(HyperText Markup Language)

Для создания веб-страниц используются языки HTML и CSS.

HTML отвечает за структуру и содержание страницы,

CSS — за внешний вид.

HTML язык разметки гипертекста - под гипертекстом в этом случае понимается текст, связанный с другими текстами указателями-ссылками.

HTML представляет собой достаточно простой набор кодов, которые описывают структуру документа. HTML позволяет выделить в тексте отдельные логические части (заголовки, абзацы, списки и т.д.), поместить на Web-страницу подготовленную фотографию или картинку, организовать на странице ссылки для связи с другими документами.

HTML не задает конкретные и точные атрибуты форматирования документа. Конкретный вид документа окончательно определяет только программа-броузер на компьютере пользователя Интернета.

HTML также не является языком программирования, но web-страницы могут включать в себя встроенные программы-скрипты на языках Javascript

Компоненты HTML

Основными компонентами HTML являются:

Тег (tag). Тег HTML это компонент, который командует Web- броузеру выполнить определенную задачу типа создания абзаца или вставки изображения.

Атрибут (или аргумент). Атрибут HTML изменяет тег. Например, можно выровнять абзац или изображение внутри тега.

Значение. Значения присваиваются атрибутам и определяют вносимые изменения. Например, если для тега используется атрибут выравнивания, то можно указать значение этого атрибута. Значения могут быть текстовыми, типа left или right, а также числовыми, как например ширина и высота изображения, где значения определяют размер изображения в пикселях.

Раздел документа HEAD

```
<head>
    <meta http-equiv="content-type" content="text/html" />
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Заголовок Страницы</title>
    <meta name="description" content="Краткое описание сайта" />
    <meta name="keywords" content="создание сайтов,продвижение сайтов,сео
    оптимизация сайтов,продвижение сайта в соц сетях,создание интернет магазина" />
   <link rel="stylesheet" href="/css/style.css" type="text/css" />
    <script src="https://code.jquery.com/jquery-3.1.0.min.js" type="text/javascript"></script>
</head>
```

<META>. Этот тег служит специальным целям, а именно - указания языка, на котором написан документ, его кодовой страницы, ключевых слов, используемых поисковыми системами для классификации этого документа и т.п. Теги <META> обычно вставляются в HTML-программу на заключительном этапе создания Web-страницы - публикации.

Для вставки в HTML-программу фрагмента программ, написанных на языке JavaScript или Viual Basic Script сценариев используют теги <SCRIPT> и </SCRIPT>.

Структура HTML-документа

```
<!DOCTYPE html>
<HTML>
<HEAD>
Шапка документа
<TITLE> Заголовок </TITLE>
</HEAD>
<BODY>
Тело документа
</BODY>
</HTML>
```

Каждый HTML-документ должен начинаться с декларации типа документа или «доктайпа». Тип документа нужен, чтобы браузер мог определить версию HTML и правильно отобразить страницу. Например, для старой версии HTML 4.01 доктайп выглядит так:

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

А для последней версии HTML 5 уже намного проще:

```
<!DOCTYPE html>
```

Раздел документа BODY

```
<body>


<div>
<p>Привет Мир!<p>
</div>

</body>
```

Теги

Теги представляют собой зарезервированные последовательности символов, начинающиеся с < (знака меньше) и заканчивающиеся > (знаком больше).

Закрытие тега отличается от открытия только наличием символа '/'

** Это мой текст **

Теги могут вкладываться друг в друга иерархически, но без пересечений, то есть допустимо вложение вида.

<teg1>
<teg2>
</teg2>
</teg1>

но не

<teg1>
<teg2>
</teg1>
</teg2>

Одиночные Теги

Парные теги обычно нужны, чтобы оформить некоторый участок текста. Благодаря паре тегов вы можете указать начало и конец этого участка. Но ведь есть теги, которые не предназначены для оформления фрагментов текста.

Например, тег для вставки изображения или тег для вставки разделительной полосы. Такие теги добавляют на страницу одиночный объект, и им не нужно для этого заключать в себя какой-то текст. Поэтому их называют одиночными.

Примеры таких тегов:

, <hr>, <img>.

Кстати, в HTML-редакторе вы увидите такие фрагменты кода:

<!-- текст -->

Они называются «комментарии», и браузер не отображает их на странице.

Атрибуты

Теги могут иметь атрибуты. Некоторые теги есть смысл использовать только с атрибутами.

Наиболее яркий пример — тег <img>, обозначающий изображение. Для него обязательно нужно указывать атрибут src, который задаёт адрес картинки (иначе браузер не сможет загрузить её).

В общем случае тег записывается следующим образом:

<имя-тега атрибут1="значение1" атрибут2="значение2" ...>

Атрибутов может быть несколько.

Заголовки

Каждый пользователь компьютера, работающий в текстовом редактором Microsoft Word знаком с понятием стиля заголовка. В HTML тоже применяется это понятие для структурирования документа и выделения важности заголовка. Всего существуют 6 стилей заголовка. Каждый из них обозначается в HTML-документе парными тегами <Hi> и </Hi> Здесь i обозначает важность стиля. H1 обозначает самый важный стиль заголовка, H2 - стиль заголовка второго уровня, а H6 - стиль заголовка самого нижнего уровня.

В подавляющем большинстве случаев для заголовков Web-страниц используют три первых уровня заголовков <H1>, <H2> и <H3>. Объясняется это тем, что размеры шрифтов оставшихся заголовуов (теги <H4> - <H5>) меньше размера обычного шрифта Web-страницы.

Вот как в документ можно добавить очень важный заголовок.

<H1>An important heading</H1>

Заголовки удобно применять для маленьких кусочков текста, а что если нам надо выделить весь текст?

Тег <font> в переводе на русский - "шрифт".

Тег <font> имеет атрибут size - размер.

Пишется и выглядит это так:

<font size="+4">

Абзацы

Понятие абзаца в HTML-документе также аналогично понятию абзаца в Microsoft Word.

Абзац обозначается в документе парными тегами <P> и </P> . Впрочем, применение закрывающего тега не является строго обязательным.

Следует помнить и о другой особенности текстовых абзацев: когда текст достигает правой границы окна Web-броузера. переход на новую строку осуществляется автоматически, независимо от расположения тега <P>.

Для отдельного абзаца можно указать тип, размер и цвет шрифта отличным от стиля остального документа.

Например:

```
<p align="center"><b><i><font face="arial" size=6 color="#c07080">
my greetings to you!
</p>
```

По умолчанию Ваш текст выравнивается браузером по левому краю, так что если Вам так и надо атрибут align="left" для параграфа можно не указывать.

Если в середине строки появилась необходимость ее разорвать - используйте одиночный тег переноса строки **
**.

Чтобы выделить текст полужирным шрифтом, воспользуйтесь тегом <b> или тегом .

Чтобы выделить текст курсивом, воспользуйтесь тегом <i> или тегом .

Абзацы

Абзац имеет атрибут **align** "выравнивание" который в свою очередь может быть равен тому ли иному значению. С помощью этого атрибута можно расположить текст по центру:

** Привет мир!!! **

По левому краю:

** Привет мир!!! **

По правому краю:

** Привет мир!!! **

Или по обоим краям документа:

** Привет мир!!! - здесь нужен текст подлиней чтобы эффект был хорошо виден при открытии документа **

Тег **<pre>**, текст заключённый в данный тег выводится браузерами на экран в том виде в котором он был набран, т.е. со всеми пробелами и переносами строк

Стили и таблицы стилей

При создании веб-страниц используются два языка: HTML и CSS. HTML отвечает за структуру и содержание, а CSS — за оформление.

Браузер объединяет HTML- и CSS-код и формирует внешний вид страницы.

CSS это аббревиатура «Cascading Style Sheets» или «Каскадные Таблицы Стилей». Обычно CSS называют просто «стили».

С помощью CSS можно задавать параметры отображения любого тега: ширину и высоту, отступы, цвет и размер шрифта, фон и так далее.

Синтаксис таких стилей очень простой:

ТЭГ {характеристика: величина}

Примеры:

```
H1 {color: white}

H1, H2, H3 {color: red}

H2 { color: blue;
font-size: 14pt;
font-family: Arial }

P {font-size: 14pt}
P {line-height: 120%}
```

Подключение таблиц стилей

Первый Способ — подключение внешнего файла с помощью тега <link>

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

Второй — использование специального тега <style> для внедрения таблицы стилей в документ

<STYLE TYPE="text/css">

B {text-align:center}

…

</STYLE>

Третий страивание в тэги документа

<P STYLE="color: green”>Этот абзац будет зеленым </P>

Задавать стили каждого тега с помощью атрибута style очень затратно и хлопотно. А ещё это приводит к засорению HTML-кода избыточными, повторяющимися кусками CSS.

Селекторы

Когда вы задаёте стили тега с помощью атрибута style, браузер сразу же понимает, к какому именно тегу применить эти стили. Но когда стили подключаются с помощью внешнего файла или через тег <style>, браузер ищет стилизуемые теги с помощью «селекторов».

Селектор находился перед фигурными скобочками в CSS-коде. В общем случае синтаксис CSS-правил выглядит так:

```
селектор {
  свойство1: значение1;
  свойство2: значение2;
  ...
}
```

Селекторы указывают на то, к каким элементам применять стили, а свойства — на то, как именно отображать элементы.

Язык селекторов очень мощный и гибкий. Простейший тип селекторов — селекторы по имени тега: p, h1 и так далее. Когда браузер видит такой селектор, он применяет стили из правила ко всем подходящим тегам.

Классы

Класс — это всего лишь один из атрибутов HTML-тегов, например:

...
...

В CSS можно задавать стили для элементов с определённым классом. Для этого используется селектор по классу, который пишется так .имя-класса, например:

.important { color: red; } — выберет все теги с классом "important"
.help { color: green; } — выберет все теги с классом "help"

Классы гибкие, их можно создавать много и называть понятными именами.

Параметр ID

Задает стилевой идентификатор — уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты. Идентификатор в коде документа должен быть в единственном экземпляре, иными словами, встречаться только один раз.

```
<style>
   #john { letter-spacing: 1em; }
   H1#bob { color: red; background-color: blue}
</style>
```

```
<p id="john"> Разрежение </p>
<h1 id="bob" > Красный </h1>
```

Приоритеты таблиц стилей

Приоритетность правил имеет следующий вид:

-связанная таблица стилей;

-импортируемая таблица стилей;

-правило с элементом HTML в качестве селектора;

-правило с параметром class в качестве селектора;

-правило с параметром ID в качестве селектора;

-встроенное в тэг HTML правило.

-!important

Свойства и значения CSS

Существует огромное количество CSS-свойств, которые влияют практически на все аспекты отображения элементов. Причём каждому свойству соответствует определённый набор значений.

Некоторые значения задаются с помощью текстовых констант, например red, bold, другие с помощью цифровых значений: 12px, 120% и так далее.

Мощь стилей заключается в том, что вы можете быстро и гибко менять внешний вид нужных элементов, особенно когда используете классы.

Свойства шрифтов

Свойство font-family задает список шрифтов.

H3 {font-family: "Times New", serif}

Свойство font-style определяет стиль шрифта: нормальный (normal), курсивный (italic) или наклонный (oblique).

Свойство font-variant определяет шрифт в виде «малых прописных» букв.

Свойство font-weight определяет жирность шрифта.

Свойства цвета и фона

Для установки цвета текста элемента существует свойство **color**.
Цвет фона определяется значением свойства **background-color**.
Фоновое изображение задается свойством **background-image**.
Пример:

```
BODY {background-image: url(serg.gif);}
P {background-image: none}
```

Свойство **background-repeat** определяет повторяемость фона.
Свойство **background-attachment** определяет, будет ли фон документа оставаться неподвижным при прокрутке окна браузера.

Один и тот же цвет можно задать двумя способами:
используя шестнадцатеричное значение цвета RGB - например #008000 - зелёный цвет, либо используя константы цвета - green

Мы уже знакомы с тегом **** у него есть еще один атрибут - color.

Привет мир!!! - То цвет шрифта станет красным.
Привет мир!!! - Будет тоже самое..

Есть еще один способ изменить цвет текста. Тег **<body></body>** "тело" - имеет атрибут text - "текст" если присудить этому атрибуту любой цвет из доступной палитры то весь текст в нашей странице окрасится, кроме тех мест, где мы "принудительно" указали другой цвет.

В строчке где стоит открывающий тег <body> пишем так:

<body text="#ff8c40 ">

Ссылки на другие документы

Тег <a> (от anchor- якорь), в него можно заключить текст или рисунок, которые станут ссылкой на те или иные документы.

Атрибут тега <a> href задаёт имя и путь к документу на который указывает ссылка

<A href="serg.html"> Домашняя страничка Сергея </A>

<A href="serg.html"><IMG SRC="photo.gif"> </A>

<A href=”mailto:serg@mail.ru"> Пишите! </A>

Внутренняя ссылка:

<A href="#abzac"> Форматирование абзацев </A>

Списки

Маркированный список

```
<ul>
    <li>Понедельник
    <li>Вторник
    <li>Среда
</ul>
```

Нумерованный список

```
<ol type="A" start="5">
    <li>Понедельник
    <li>Вторник
    <li>Среда
</ol>
```

Под элементом списка может подразумеваться не только элемент в виде текста, а, вообще, говоря всё, что угодно. Например, вместо текста может быть ссылка или даже изображение. Просто само форматирование будет в виде списка.

Изображение

<img src="/img/2.png" alt="Текст"
align="left" border="1" width="100" >

тег <img> не требует закрывающего тега!

Блочная верстка

Любая веб-страница состоит из расположенных на ней элементов, и практически всегда за их размещение отвечает блочная верстка div. Конечно, существует еще и табличная верстка с использованием тегов <table>, <tr>, <td>, и существуют даже споры о том, какую лучше использовать систему — блочную или табличную.

Однако в действительности в в настоящее время вы не встретите ни одного из современных, популярных и удобных сайтов, использующих только табличную верстку.

В лучшем случае она используется только для того, для чего она и предназначена — то есть для создания таблиц, но никак не для формирования самой структуры сайта.

Основа блочной системы — это тег , который является контейнером для контента. Внутри него также могут содержаться другие контейнеры

Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут class или id с именем селектора.

.

Блочная верстка

Как правило, стандартная структура сайта формируется следующим образом: существует основной контейнер (часто ему присваивается класс с названием wrapper, container, main и т.д.). Внутри этого контейнера располагаются блоки меню, контентной части, сайдбара.

По умолчанию, каждый новый блок располагается с новой строки. Для того, чтобы расположить блок слева или справа от другого (например, чтобы расположить сайдбар справа), используется свойство float. По умолчанию оно имеет значение «none», но можно также выставлять значения «left» и «right».

Рассмотрим это свойство на примере с двумя блоками.

```
<div style="float:left;background:#FF00DC; width:300px;">Блок для контента</div>
<div style="float:left;background:#C0C0C0; width:100px;">Блок для сайдбара</div>
```

Важно учитывать, что свойство float распространяется не только на сам блок, в котором оно прописано, но и на последующий элемент, который будет идти за этим блоком. То есть если мы к вышеописанным двум блокам добавим еще один блок, не указывая ему никаких свойств, то он расположится не с новой строки, а начнется справа от второго блока.

Для того, чтобы избежать этого, блочная верстка div использует свойство clear, которое должно быть задано для того блока, который мы хотим расположить с новой строки. Чаще всего для этого ему задается значение «both», но можно также задать значения «left» или «right», если мы хотим не просто расположить блок на новой строке, но и задать ему выравнивание.

```
<div style="clear:both;background:#FFD800; width:400px;">Новый блок, расположенный снизу</div>
```

Отступы в блочной верстке

Помимо расположения блоков, немаловажным является задание отступов как между блоками, так и внутри них. Для этого, соответственно, используются свойства margin и padding.

Отступы задаются отдельно для верхней, правой, нижней и левой частей элемента. Их можно задать одной строкой путем перечисления четырех значений:

margin: 20px 10px 0 40px

Блок с такими параметрами будет располагаться на 20 пикселей ниже вышестоящего элемента, на десять пикселей от правостоящего элемента, будет иметь нулевой отступ снизу и будет иметь отступ размером в 40 пикселей слева.

Если все те же самые показатели указать в свойстве **padding**, то это будут внутренние отступы для контента по отношению к границам блока, в которых он расположен.

Можно также задавать отдельные свойства для отдельных граней с помощью **margin-top, margin-bottom, margin-left, margin-right** (и аналогично для padding). В таком случае, если какая-то из граней не будет указана, то отступ с ее стороны будет равен нулю или будет определен общими свойствами css, заданными для блоков на странице.

Позиционирование элементов на Web-странице

Свойство **left** (x-координата) используется для задания в пикселях расстояния элемента от левого края окна.

Свойство **top** (y-координата) задает расстояние в пикселях до элемента от верхнего края окна.

Свойство **z-index** указывает, в каком порядке элементы будут перекрывать друг друга. Элемент с более высоким z-индексом будет появляться над элементами с более низким индексом

Свойство **position** определяет способ позиционирования элемента на странице: *статический, относительный* или *абсолютный.*

Значения свойства – static, relative, absolute.

Пример:

```
<div style="position:absolute; top:25px; left:20px; width:400px">
<H3> текст </H3>
<img src="/img/2.png" style="position:absolute; top:30px; left:30px;
    z-index:-1">
</div>
<div style="border-radius: 5px; width: 70%; margin-left: 14%; margin-right:
    14%; border: solid 1px #000; height: 50px; float: left">
```

Свойства видимости и переполнения

Свойство visibility управляет отображением элемента. Если его значение равно visible (значение по умолчанию), то элемент отображается, если оно установлено равным hidden, то элемент не отображается

Свойство display может иметь значение block и none.

Свойство opacity определяет дрозрачность блока.

Свойство overflow определяет поведение элемента, когда его размеры не соответствуют размерам блока.

Значения свойства:

-visible

-hidden

-auto

-scroll

Таблицы

```
<table style="border: solid 1px #ccc">

    <tr style="border: solid 1px #ccc">
        <th style="border: solid 1px #ccc">№</th>
        <th>Фамилия </th>
        <th>Имя</th>
        <th>Група</th>
    </tr>
    <tr>
         <td>1</td>
        <td>Шевченко</td>
        <td>Андрей</td>
        <td>2-МП</td>
    </tr>
</table>
```

Диалоговые формы

Тег <input> (с англ. ввод данных) позволяет создавать элементы интерфейса: кнопки, текстовые поля, переключатели и флажки.

Основной метод получения информации от пользователя (читателя) базируется на этом теге. В зависимости от атрибута type, отображается в виде различных элементов управления.

```
<input type="checkbox" checked="checked"/> флаг
<input type="radio"/> переключатель
<input type="file" value="имя файла"/>
<input type="text" value="Логин"/>
<input type="password" value="пароль"/>
<input type="button" value="Обычная кнопка"/>
```

Пример:

```
<form method="post" action="/cgi-bin/data" >
    <input type=text name="Name1" value="Имя" size=25 maxlength=45>
    <input type=password name="Name2" size=25 >
    <input type=checkbox name="Name3" value="5" checked>
    <select name="var-name">
        <option value=”red”>red</option>
        <option selected="selected">green</option>
    </select>
    <textarea name="Name4" rows=5 cols=10></textarea>
</form>
```