Разработка программного обеспечения на языке Python

Лекция 1. Концепции web-приложения

Посмотрите видеоуроки и ответьте на контрольные вопросы после лекции

Основы html

Рассмотрим основы языков разработки фронтенд части web приложений.

HTML (HyperText Markup Language) – это язык разметки, который используется для создания веб-страниц. Он позволяет описывать структуру и содержание веб-страницы, а также управлять их внешним видом с помощью стилей CSS.

CSS (Cascading Style Sheets) – это язык описания внешнего вида веб-станиц. С помощью CSS можно управлять цветами, шрифтами, размерами и многими другими аспектами внешнего вида веб-страниц, чтобы сайты выглядели эстетично и привлекательно. Кроме того, CSS позволяет управлять адаптивностью веб-сайтов, то есть их совместимостью с различными устройствами, такими как компьютеры, смартфоны и планшеты.

Основы HTML

HTML (HyperText Markup Language)_представляет язык разметки гипертекста, используемый преимущественно для создания документов в сети интернет. HTML начали разрабатывать в начале 90-х годов как язык для создания веб-страниц. В настоящее время большинство сайтов так или иначе используют HTML.

В 2014 году официально была завершена работа над стандартом HTML5, который привнес в HTML много нового, по сравнению с предыдущими версиями. Развитием HTML5 занимается World Wide Web Consortium (сокращенно W3C - Консорциум Всемирной Паутины) - независимая международная организация, которая определяет стандарт HTML5 в виде спецификаций. Текущую полную спецификацию на английском языке можно посмотреть по адресу https://www.w3.org/TR/html5/. При этом HTML5 продолжает развиваться, соответственно выпускаются обновления к спецификации.

Для работы с HTML5 в первую очередь требуется текстовый редактор, чтобы набирать текст веб-страниц на html. На данный момент одним из самых простых и наиболее популярных текстовых редакторов является Notepad++, который можно найти по адресу http://notepad-plus-plus.org/. Кроме того, html страницы можно создавать используя различные среды разработки, например, Visual Studio Code или pyCharm. И также потребуется веб-браузер для запуска и проверки написанных веб-страниц. В качестве веб-браузера можно взять последнюю версию любого из распространенных браузеров - Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.

Особенности HTML

- Это язык разметки, обеспечивающий гибкость при разработке веб-страниц с текстом.
- Он прост в использовании и обучении.
- HTML не зависит от операционной системы и может использоваться в Windows, Linux, Macintosh и т. д.
- HTML позволяет программистам добавлять изображения, видео и аудио на веб-страницу, чтобы сделать ее более интерактивной.
- HTML позволяет программистам добавлять ссылки на веб-страницы, помогая читателям просматривать интересующую их информацию.
- HTML нечувствителен к регистру. Вы можете использовать теги как в нижнем, так и в верхнем регистре.

Документы HTML

Все документы в HTML должны начинаться с обьявления типа документа с помощью элемента <!DOCTYPE html>. Сам HTML документ начинается с <html> и заканчивается с</html>. Видимая часть документа находится между открывающим тегом <body> и закрывающим </body>.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
</body>
</html>
```

Элемент <!DOCTYPE>

В начале документа пишется <!DOCTYPE>. Таким образом объявляется тип документа, что помогает браузерам правильно отображать веб-страницы. Элемент <!DOCTYPE> не чувствителен к регистру.

Тег

Язык HTML отвечает за структуру и содержание страницы. Составными элементами для построения структуры документа и наполнения его содержимым являются теги.

Тег – это основная конструкция языка HTML. Благодаря тегам любой браузер понимает, каким смыслом вы наделяете тот или иной элемент на веб-странице. Тег состоит из имени, заключённого между знаками «<» и «>». Примеры тегов: <h2>, <a>, <ul>. В качестве имени тега должны использоваться определенные, закрепленные в HTML имена.

Все теги можно разделить **на две категории**:

- парные;
- одиночные.

Например, тег <img> – одиночный: <img src="https://google.com">

У парных тегов, в отличие от одиночных, есть закрывающий тег. В закрывающих тегах перед именем ставится символ / («слеш»). Пример парного тега: <h1>Текст</h1>

В этом примере <h1> открывающий тег, а </h1> – закрывающий.

На основании применения тегов формируется содержимое страниц. В зависимости от того, какие теги применены, может быть добавлена на страницу картинка, заголовок, таблица, список и т.д. Таким образом, на основании тегов создается любой элемент HTML.

HTML-элементы

Элемент HTML – это набор из **тега** и его содержания. Чаще всего он состоит из открывающего тега, содержимого и закрывающего тега. Примеры:

```
<h1>Заголовок первого уровня</h1>

<p>Параграф</p>
```

В данных примерах теги <h1> и являются открывающими, текст внутри тега – это содержимое элемента, теги </h1> и – закрывающие.

Некоторые элементы не имеют содержимого (например, элемент
). Эти элементы называются пустыми. Пустые элементы не имеют закрывающего тега.

Вложенные элементы HTML, структура документа

Любой HTML-документ будет представлять собой набор из вложенных друг в друга элементов HTML. Далее рассмотрим структуру документа:

Рассмотрим простой пример того, как может выглядеть HTML-документ:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
  <title>Заголовок страницы</title>
</head>
<body>
  <h1>Заголовок</h1>
  <p>Абзац</p>
</body>
</html>
```

Элемент **<html>**является корневым и определяет весь HTML-документ. У него есть начальный тег <html>и конечный тег </html>.

Внутри <html>элемента находятся элементы <head> и <body>:

Элемент**<head>**предназначен для хранения служебной информации о странице. Он располагается первым в элементе<html> , сразу перед <body> . Этот элемент нужно закрыть тегом </head> сразу после перечисления его содержимого. Так, например, в элементе <head> хранится информация о теге <title>, который отображает заголовок страницы в браузере и поисковой выдаче и является важным элементом для ранжирования сайта в поисковых системах.

Элемент **<body>**представляет собой контент (содержимое) документа HTML. В документе может быть только один элемент <body>. У него есть начальный тег <body>и конечный тег </body>. Этот элемент должен быть вторым в элементе <html>, т.е. располагаться после <head>. Внутри <body>элемента располагается основное содержимое страницы. В рассматриваемом примере <h1>и : Элемент <h1> – это HTML-заголовок, который чаще всего используется для разметки заголовка веб-страницы. Он имеет начальный тег <h1>и конечный тег </h1>. Пример: <h1>Заголовок</h1>

Элемент определяет абзац. Элемент является блочным, а значит его содержимое всегда начинается с новой строки и занимает всю доступную ширину. Он имеет начальный тег и конечный тег . Пример: Абзац

Атрибуты HTML-элементов

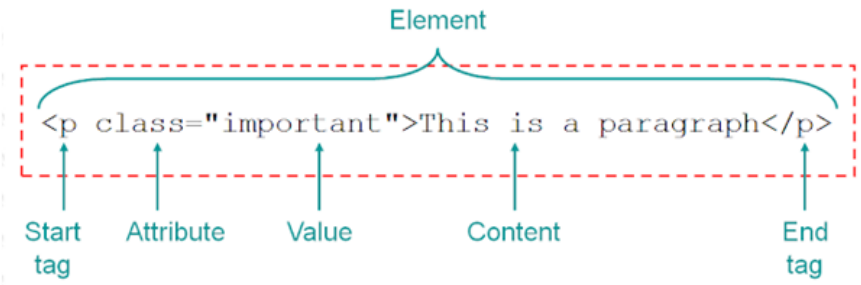
У каждого тега есть свое назначение: какие-то нужны для отображения заголовка, какие-то – для создания таблицы. Назначение определяется по имени тега. На основании тега создаются элементы в HTML-коде. Кроме указания тега можно задать дополнительные свойства каждому элементу при помощи атрибутов.

HTML-атрибуты это специальные свойства, которые управляют поведением HTML-элемента. Они добавляют дополнительную функциональность, либо меняют поведение элемента по умолчанию. Атрибуты элемента выражаются внутри начального тега элемента.

Атрибуты задаются в открывающемся теге и состоят из пары имя атрибута = "значение". В общем случае тег с атрибутами записывается следующим образом:

```
<имя-тега атрибут1="значение1" атрибут2="значение2" …> Содержимое </имя-тега>
```

В качестве примера рассмотрим атрибут для элемента «абзац» (тег p) с именем class и его значение равно important



Далее кратко рассмотрим некоторые атрибуты.

Атрибут href. Для перехода с одной страницы на другую на веб-сайтах используют ссылки. За ссылки отвечает тег a. Но чтобы переход состоялся, необходимо указать адрес назначения. Именно для этого в теге a используется атрибут href. Если создать ссылку без атрибута href, то нажатие по ссылке не приведет к переходу на другую

страницу.

Атрибут src. Для отображения картинок в HTML используется тег img. Но чтобы картинка появилась на экране, нужно указать ее источник (source). Для этого используется атрибут src. Без указания источника тег img будет отображать пустую рамку, поэтому указание атрибута src для тега img является обязательным.

Для одного элемента можно указать **несколько атрибутов.**

Пример. Добавим к рисунку дополнительные атрибуты width и height для указания ширины и высоты картинки в пискелях.

```
<!DOCTYPE html>
```

```
<html lang="en">
 <head>
     <meta charset="UTF-8">
     <title>Title</title>
 </head>
 <body>
 <img src="f1.jpg" width="1000" height="500">
 </body>
 </html>
```

Атрибут style. При помощи style него можно изменить стиль отображения любого элемента. Можно влиять на размер, цвет, прозрачность, месторасположение и т.д. Подробнее о стилях мы поговорим в дальнейшем.

Атрибут title. Данный атрибут редко используется, при помощи него можно указать подсказку для элемента. При наведении указателя мыши на элемент через некоторое время появится текст, который вы укажете в качестве значения атрибута title.

Основные элементы html

HTML-заголовки

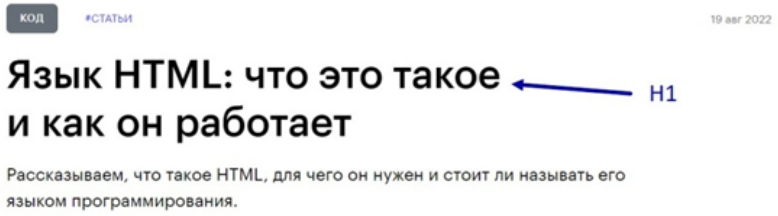
Для создания заголовков на сайте используется 6 парных тегов:

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>,

где <h1> – заголовок первого уровня, самый важный и описывающий главную тему текста, а <h6> – заголовок самого низшего уровня. На практике в текстах редко встречаются подзаголовки ниже третьего уровня. Поэтому чаще всего используются теги <h1>, <h2> и <h3>.

Заголовки выполняют важные функции на веб-странице.

1. Показывают важность раздела, к которому относятся. Чем больше заголовок, тем более он значимый. На сайтах тегом <h1> обычно обозначают название раздела или статьи.



2. Регулирует размер текста. Чем выше уровень заголовка, тем больше размер шрифта. Самым верхним уровнем является уровень 1 (<h1>), а самым нижним – уровень 6 (<h6>).

3. Имеет значение при ранжировании сайта в поисковой выдаче. Поисковые системы учитывают содержание h1, чтобы понять, насколько контент страницы соответствует поисковому запросу.

HTML-параграфы

Помимо заголовков, с помощью которых мы создаем основную структуру текста, важным элементом HTML является параграф (или абзац). Параграфы помогают разделять большой текст на небольшие смысловые блоки и таким образом делают текст более понятным и комфортным для чтения. Эти блоки браузер автоматически отделит друг от друга отступами. Параграфы создаются с помощью парного тега , в тело которого помещается небольшая часть текста. Например: Текст параграфа

По умолчанию абзацы начинаются с новой строки и отделяются от остального контента отступами сверху и снизу. Но если вы хотите добавить в сам параграф дополнительные пробелы между словами или переносы строк, браузер их проигнорирует. Несмотря на то, что в вашем HTML-коде все эти символы присутствуют, на самой странице в браузере их не будет.

Теги <hr> и

При помощи данных тегов можно добавлять переносы в любом месте HTML-кода.

Тег <hr> определяет тематический разрыв на HTML странице и чаще всего отображается в виде горизонтальной линии. Элемент <hr> используется для разделения содержимого (или определения изменения) на HTML-странице. Тег <hr> – это пустой тег, что означает, что у него нет конечного тега.

HTML-элемент
 определяет разрыв строк. Если требуется начать текст с новой строки, не начиная новый параграф, можно использовать тег
.

Теги форматирования текста

Теги форматирования позволяют изменять оформление выделенного текста. Они делятся на две категории: теги физической разметки, которые отвечают за стилевое оформление (жирное начертание, курсив, шрифт и т.д.), и теги логической разметки, которые несут смысловую нагрузку (например, дают понять поисковым системам, по каким словам необходимо ранжировать веб-страницу).

Далее рассмотрим некоторые из тегов.

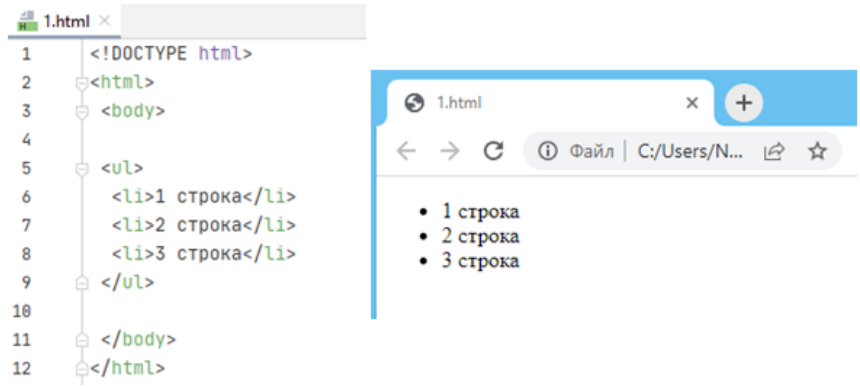
теги	описание
Теги <b> и	задают полужирное начертание шрифта. Разница между ними заключается в том, что тег <b> является тегом физической разметки и выделяет текст без акцента на его важность. Тег определяет текст, которому придают важность (например, для поисковиков)
Теги <i> и	задают курсивное начертание шрифта. Тег <i> просто изменяет оформление текста и не воспринимается, как важный, браузерами и поисковыми машинами. Тег предназначен для выделения текста, на который стоит обратить внимание.
Тег <pre>	используется для включения в HTML-документ предварительно отформатированного текста. Браузер сохраняет и отображает все пробелы и переносы, которые есть внутри тега <pre>.
Тег <small>	определяет размер шрифта текста на один размер меньше, чем у родительского элемента.
Теги <del> и <s>	Тег <del> выделяет часть текста, которая была удалена из документа. Тег <s>используется для определения текста, который больше не актуален.
Теги <sub> и <sup>	Тег <sub> используется для определения текста с нижним индексом. Тег выравнивает элемент, как подстрочный. Тег <sup> используется для определения текста в верхнем индексе.

HTML-списки

Списки – один из способов представления контента на странице. С их помощью легко группировать небольшие связанные фрагменты. **Виды списков:**

1. Неупорядоченный (маркированный) список
2. Упорядоченный (нумерованный) список
3. Список определений

Неупорядоченный список (*Unordered List).* В HTML тег <ul> отвечает за создание списка. Каждый элемент списка должен находиться внутри тега <li>. Пример.



Перед каждым из элементов будет проставлен маркер, поэтому неупорядоченный список еще называют **маркированным.** Браузеры по умолчанию в качестве маркера элемента списка добавляют метку – закрашенный круг. Но есть возможность изменить тип маркера. Для маркированного списка доступны **3 типа маркеров** по умолчанию: *disc*, *square* и *circle*. Задать тип маркера можно при помощи свойства list-style-type. Тип маркера может быть задан как для списка в целом (свойство применяется к <ul>), так и для конкретного элемента (свойство применяется к <li>).

Упорядоченный список (*Ordered List),* HTML-тег, отвечающий за создание упорядоченного списка, называется <ol>. Каждый элемент списка должен находиться внутри тега <li>. По умолчанию применяется **нумерованный** список с арабскими числами. Пример.

```
<!DOCTYPE html>
 <html>
  <body>
   <ol>
    <li>1 строка</li>
    <li>2 строка</li>
    <li>3 строка</li>
   </ol>
  </body>
 </html>
```

При этом доступны различные виды нумерации, которые задаются для тега <ol>. Для указания типа нумерованного списка применяется атрибут **type** тега <ol> (например, "1" – арабские цифры. Используется по умолчанию; "A" – прописные латинские буквы; "I" – прописные римские числа). Атрибут **reversed** – задает отображение списка в обратном порядке; Атрибут **start** – задает начальное значение, от которого пойдет отсчет нумерации, например, конструкция <ol start="10"> первому пункту присвоит порядковый номер «10»).

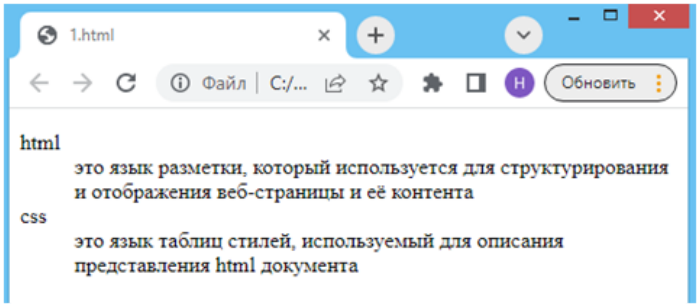
Список определений (definition list) представляет такой список, который содержит термин и определение к нему. Для создания списка определений применяется парный тег <dl>. Внутрь этих тегов помещаются элементы списка.

Каждый элемент списка состоит из термина и определения. Термин помещается в парный тег <dt> (сокращение от "definition term"), а определение – в парный тег <dd> (сокращение от "definition description").
Теги <dt> и <dd> пишутся парами внутри <dl>.

Пример. Определения для html, css

```
<!DOCTYPE html>
 <html>
     <body>
         <dl>
           <dt>html</dt>
           <dd>это язык разметки, который используется для
               структурирования и отображения веб-страницы и её контента</dd>
           <dt>css</dt>
           <dd>это язык таблиц стилей, используемый для
               описания представления html документа</dd>
         </dl>
     </body>
 </html>
```

Отображение данного примера в браузере:



HTML-ссылки

Ссылки позволяют переходить с одного сайта на другой или из одного раздела страницы к другому. Для создания ссылки необходимо воспользоваться парным тегом <a>. Все, что будет находиться между открывающим тегом <a> и закрывающим будет считаться ссылкой, и пользователь сможет кликнуть по этому содержимому. Общий синтаксис создания ссылок следующий: текст ссылки

Обязательный атрибут элемента <a> – href, указывает на пункт назначения ссылки. Текст ссылки – это та часть, которая будет видна читателю. Нажав на текст ссылки, читатель отправится на указанный URL-адрес. **Пример.** Ссылка на Википедию: Html5

Абсолютные и относительные ссылки. Существует два типа ссылок:

- абсолютные;
- относительные.

Абсолютные ссылки - предполагает указание полного адреса страницы, на которую вы ссылаетесь. Обычно они начинаются с протокола http:// или https://, далее идет название домена сайта и сама ссылка. Пример абсолютной ссылки:

```
<a href=" https://ru.wikipedia.org/wiki/HTML5/">
```

Относительные ссылки. Абсолютные ссылки помогают добавить переходы на сторонние ресурсы. Но часто трбуется сделать переходы на другие страницы нашего сайта. Такие переходы можно назвать внутренними, и с помощью их вы можете перейти с отображения одного вашего HTML документа на другой. Для этого также требуется указать адрес. Только теперь ресурсами будут файлы, которые находятся на вашем компьютере. И здесь самое важное в их месторасположении – где они хранятся относительно вашего текущего документа.

Пример. Рассмотрим пример, где указан список ссылок на источники с информацией о web._Каждая такая ссылка введет на файлы HTML, которые будут содержать информацию по каждой теме. Создадим в этой же папке два файла: html_info.html и css_info.html. Относительно файла index.html созданные файлы лежат в той же папке, поэтому мы можем просто указать их названия в атрибуте href.

```
<!DOCTYPE html>
 <html>
  <body>
   <p>Информация о web:</p>
   <ul>
       <li><a href="html_info.html">
        html5</a></li>
       <li><a href="css_info.html">
         css</a></li>
   </ul>
  </body>
 </html>
```

Рисунки

Для добавления элемента изображения на страницу необходимо использовать одиночный тег <img>. У тега <img> нет содержимого, поэтому он является пустым. Содержать он может только атрибуты, причем обязательным является атрибут src – путь к изображению, добавляемому на страницу.

```
<img src="url: путь к файлу с изображением">
```

Остальные атрибуты являются необязательными. Ниже представлены некоторые важные атрибуты тега <img>:

- **alt** – данный атрибут задает альтернативное текстовое описание изображения;
- **width** – ширина изображения в пикселях;
- **height** – высота изображения в пикселях.

Чтобы сделать изображение ссылкой, необходимо поместить элемент <img> внутрь элемента <a>.

При рассмотрении некоторых следующих элементов необходимо знать их основные свойства, которые могут быть заданы как атрибуты. Рассмотрим атрибут, связанных с форматированием внешнего вида элементов.

Таблицы

Таблица – это структурированный набор данных, состоящий из строк и столбцов (**табличных данных**). Таблицы – это очень простой и наглядный способ структурирования информации.

Таблица создаётся при помощи парного тега <table>, который является контейнером для элементов таблицы, и все содержимое таблицы должно находиться внутри него. С точки зрения HTML таблицу проще представлять не как набор строк и колонок, а как набор строк и содержащихся внутри строки ячеек. Каждая строка таблицы определяется с помощью тега <tr> (от английского table row). Каждая ячейка определяется с помощью тега <td>(от английского table data).

Пример. Список студентов, столбцы таблицы – фамилия, имя и группа.

```
<!DOCTYPE html>
 <html>
 <body>
 <table style="width:100%">
     <tr>
         <td>Фамилия</td>
         <td>Имя</td>
         <td>Группа</td>
     </tr>
     <tr>
         <td>Иванов</td>
         <td>Иван</td>
         <td>КИ21-02</td>
     </tr>
     <tr>
         <td>Петров</td>
         <td>Николай</td>
         <td>УБ20-01</td>
     </tr>
 </table>
 </body>
 </html>
```

Заголовок таблицы. Как правило, в таблицах первый ряд отводится под название колонок и является заголовком таблицы. Его названия должны выделяться на фоне остальных ячеек. Для создания ячейки заголовка нужно воспользоваться тегом <th> (от английского table header).

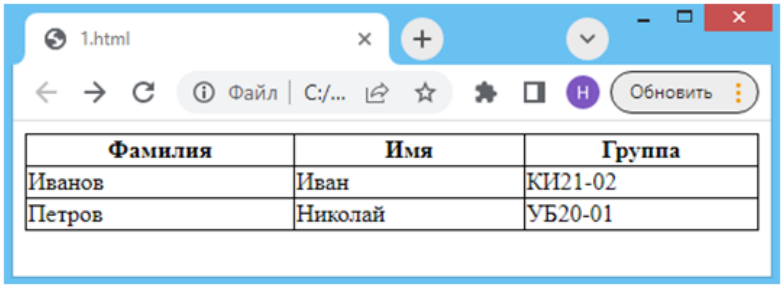
Добавление подписи. Чтобы добавить подпись к таблице, используется тег <caption>.

Стилизация таблицы. Чтобы отображение данных было максимально похоже на таблицу, можно добавить границы для ячеек. Для этого нужно воспользоваться CSS свойством border и задать его для каждой ячейки и самой таблицы. Свойство border-collapse устанавливает, как отображать границы вокруг ячеек таблицы. Это свойство нужно, когда установлена рамка, тогда в месте стыка ячеек получится линия двойной толщины. Значение collapse оставляет в качестве границ одну линию.

Пример. Добавим границы к таблице.

```
<!DOCTYPE html>
 <html>
 <head>
     <style>
         table, th, td {
             border: 1px solid black;
             border-collapse: collapse;
         }
     </style>
 </head>
 <body>
 <table style="width:100%">
     <tr>
         <th>Фамилия</th>
         <th>Имя</th>
         <th>Группа</th>
     </tr>
     <tr>
         <td>Иванов</td>
         <td>Иван</td>
         <td>КИ21-02</td>
     </tr>
     <tr>
         <td>Петров</td>
         <td>Николай</td>
         <td>УБ20-01</td>
     </tr>
 </table>
 </body>
 </html>
```

Отображение данной таблицы в баузере.



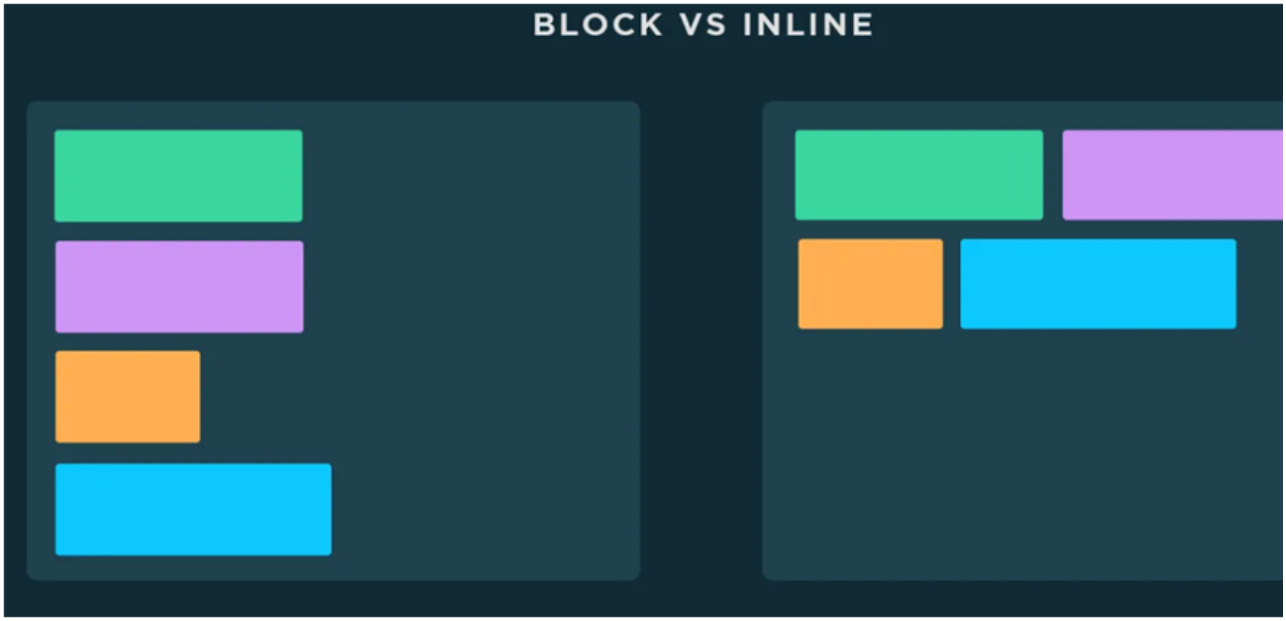
Блочные и строчные элементы

Каждый элемент HTML по способу отображения можно отнести к одной из двух групп: блочные (block) элементы, строчные (inline) элементы

Блочные элементы – отдельные структурные единицы, которые используются для создания структуры веб-страницы. Такие блоки имеют форму прямоугольника и занимают всю ширину экрана. Элементы, для которых это значение задано по умолчанию <h1>⋯<h6>, <p>, <ol>, <ul>, и др.

Строчными называются такие элементы, которые являются непосредственной частью строки. Элементы, для которых это значение задано по умолчанию, – , <a>, <q>, <code> и др., в основном они используются для изменения вида текста или его смыслового выделения.

На рисунке ниже показан пример расположения блочных элементов (слева) и строчных (справа)



Далее рассмотрим примеры блочного и строчного элемента.

Тег

Элемент используется как контейнер для других элементов HTML. Тег не имеет никакого другого назначения, кроме как быть хранилищем для других элементов. Он используется для группирования в одном блоке логически связанных между собой элементов с целью разметки и стилизации. HTML-тег является блочным, его содержимое всегда начинается с новой строки и занимает всю доступную ширину, но отступов от других элементов он не имеет, то есть элементы, расположенные до и после него, будут вплотную прилегать сверху и снизу. Отступы легко можно отрегулировать с помощью CSS. Тег может содержать вложенные элементы в неограниченном количестве. Содержимым тега может быть любой другой элемент HTML: например, абзац, ссылка, таблица, другой блок с контентом (т.е. другой div вместе со всем своим содержимым). Тег – это самый универсальный и широко используемый HTML-элемент.

Тег

HTML-тег является строчным элементом и представляет из себя контейнер для текста. Он используется в ситуациях, когда нужно изменить стиль для части текста, не помещая его при этом в блочный элемент. Сам по себе он, как и тег , не имеет какого-либо визуального представления и семантического значения, но при использовании совместно со стилями CSS он является инструментом для форматирования текста в браузере. Для этого нужно поместить внутри него содержимое, которое вы хотите стилистически изменить, и определить стиль при помощи CSS

flask

ПРЕДЫДУЩИЙ ЭЛЕМЕНТ КУРСА

◄ Запись синхронных занятий по курсу "Веб-программирование на Python"

Перейти на...

СЛЕДУЮЩИЙ ЭЛЕМЕНТ КУРСА

Задание 1. Дизайн страницы приложения ►