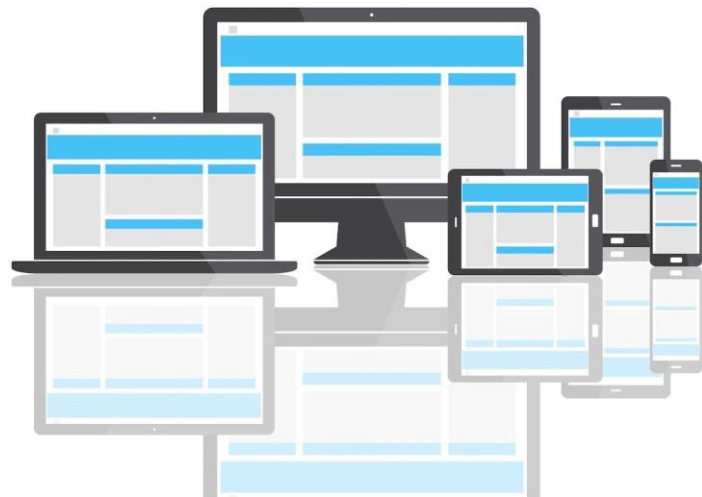




# Создание веб-приложений (основы): HTML, CSS, JS



# План занятия



1

Сайт. Страницы сайтов. Знакомство с HTML (синтаксис, правила)

3



Работа с формами. Отправка запросов на сервер разными методами (GET, POST)

2

Знакомство с CSS (синтаксис, правила). Часто используемые свойства.


4

О языке программирования JavaScript



# Определения


**HTML** - ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ документа в Интернете (не язык программирования). Большая часть всех страниц в Интернете созданы с помощью языка HTML.



**Браузер** - программное обеспечение (компьютерная программа) для просмотра веб-страниц, а также используется для управления веб-приложениями и решения других простых задач. Браузер использует протокол HTTP(s) для отправки запроса от клиента на сервер.

**Страница сайта (веб-страница)** - это текстовый документ, размеченный (сверстанный) на языке HTML. Веб-страница загружается и отображается браузером.

**Сайт (веб-сайт)** - специальное место для хранения веб-страниц и других документов или файлов, имеющие отношение к этому сайту.



# Определения

**Язык HTML** позволяет грамотно структурировать и стилистически оформить всю информацию, размещаемую на веб-странице, например: добавить изображения, ссылки на другие страницы, таблицы, видео, звук и т.п.

**Оформление веб-страницы** происходит с помощью языка **CSS**, например: сделать текст цветным, большего размера или другим шрифтом, изменить отображение списка или таблицы.

**Для добавления на веб-страницу каких-либо функциональных возможностей используется язык JavaScript**, например: отправка сообщения владельцу сайта, добавление товара в корзину и т.п.

# Веб-сайт

Из чего состоит сайт:

Интерфейс пользователя, **Front-end часть**

Совокупность элементов, средств и методов, при помощи которых пользователь взаимодействует с различными устройствами, аппаратурой, машинами и т.п.

Примеры: меню в телефоне или планшете, список каналов в ТВ, пульт управления ТВ, электронные часы, автомагнитола, приборная панель в авто, бортовой компьютер и т.д.

В разработке UI обычно используют:

- HTML
- CSS
- JavaScript (скрипты)



# Веб-сайт

## Функции сайта, **Back-end** часть

- PHP
- JavaScript (скрипты)
- Java
- Python
- Ruby
- C

Примеры функций сайта: взаимодействие с БД сайта, отправка и обработка сообщений или файлов с сайта, оформление покупки (покупка товаров), проведение оплаты, отправка отзыва за товар или услугу и т.п.

# Веб-сайт

## Этапы создания сайтов:

**Концепция (этап поиска идеи или прототипирования)** - о чем будет сайт? Кому он будет интересен? Какие требования к технической или визуальной его части? Реализация любой идеи начинается с прототипа. Прототип - описание или визуальное представление будущего сайта (схема, черновик или набросок), которое имитирует работу различных элементов страницы или функций сайтов.

**Дизайн** - это точное графическое представление будущего сайта, внешнее его оформление. И это не только красивая обертка для сайта, но и целая стратегия того, как этот сайт должен работать и взаимодействовать с пользователем. Работа **UX/UI**. UI — это интерфейс пользователя (от английского user interface). UX в переводе означает «опыт взаимодействия» (user experience).

**Верстка (создание интерфейса)** - создание структуры html-кода, размещающего элементы веб-страницы (изображения, текст и т. д.) в окне браузера, согласно разработанному макету дизайну.

**Программирование (создание функций)** - как будет работать сайт? Какие функции будут на сайте? Какие модули или библиотеки использоваться для реализации необходимых функций? Как будет осуществляться наполнение сайта? Какой будет система управления содержимым?

# Веб-сайт

## Этапы создания сайтов:

**Наполнение** - любой сайт должен "говорить" со своими посетителями. Сайт любой компании в интернете - это ее лицо, главный представитель, который виртуально общается со своими потенциальными клиентами (через текст, изображения, видео материалы и прочий контент).

**Тестирование** – проверка работоспособности сайта и его соответствия предъявляемым требованиям (back-end, front-end, usability, security, etc.)

**Размещение** - место размещение сайтов - веб-сервер - какой-то удаленный компьютер. Прежде всего он должен быть надежным, быстрым, современным - чтобы выдержал любые нагрузки и любой трафик посетителей. Скорость работы сайта влияет на поисковое продвижение.

**Продвижение и реклама** - этап, на котором сайт «заполняется» клиентами. Как правило, продвижение и реклама осуществляется отдельными специалистами.



# HTML язык

## HTML теги:

### 1. Парные теги

`<тег>Содержание тега</тег>`

### 2. Непарные теги

`<тег>`

или `<тег />`

### 3. Теги с атрибутами


`<тег атрибут="значение" />`

`<тег атрибут1="значение1" атрибут2="значение2" />`

`<тег атрибут1="значение1" атрибут2="значение2" атрибут3="значение3" атрибут4="значение4">Содержание тега</тег>`

# HTML язык


## 4. Обязательные теги:



```
<html>  
<head></head>  
<body></body>  
</html>
```

**<html>** (указывает на то, что это HTML документ),  
**<head>** (обозначает служебную шапку HTML документа),  
**<body>** ("тело" HTML документа).

Обязательные теги формируют "скелет" (структуру) веб-страницы и являются его неотъемлемой частью.



# HTML язык

## Рекомендованная структура HTML страницы:

```
<!DOCTYPE html>
```

```
<html lang="ru">
```

```
<head>
```

```
  <meta charset="UTF-8" />
```

```
  <title>Документ</title>
```

```
</head>
```

```
<body>
```

```
  <!-- Мой первый HTML документ -->
```

```
  <p>Абзац текста документа</p>
```

```
</body>
```

```
</html>
```

**<!DOCTYPE>** предназначен для указания типа текущего документа и версии языка HTML.

Необходим для понимания браузером, как следует отображать текущую веб-страницу.

**<html>** - главный контейнер, который заключает в себе всё содержимое веб-страницы, включая элементы **<head>** и **<body>**. Закрывающий тег обязателен.

**<head>** - предназначен для хранения элементов, цель которых — помочь браузеру в работе с данными. Закрывающий тег обязателен.

**<title>** - элемент определяет главный заголовок документа и отображается, как правило, на вкладке браузера или в его заголовке. Закрывающий тег обязателен.

**<body>** - предназначен для хранения всего содержимого (текст, изображения, таблицы, блоки, другие теги и т.п.) веб-страницы и является важным элементом документа. Закрывающий тег обязателен.

# HTML язык

## Другие теги:

`<p></p>` - абзац

`<strong></strong>` - выделение жирным

`<b></b>` - выделение жирным

`<i></i>` - выделение курсивом

`<em></em>` - выделение курсивом

`<a></a>` - гиперссылка

`<table></table>` - таблица

`<tr></tr>` - строка таблицы

`<td></td>` - ячейка в строке таблицы

# HTML ЯЗЫК

**<p>** - элемент определяет текстовый параграф (абзац), всегда начинается с новой строки. Каждый новый абзац должен начинаться с нового тега **<p></p>**. Закрывающий тег обязателен.

**Синтаксис:** **<p>Текст</p>**

**<strong>, <b>** - устанавливает жирное начертание шрифта. Закрывающий тег обязателен.

**Синтаксис:**

**<strong>Текст</strong>**

**<b>Текст</b>**

**<i>, <em>** - устанавливает курсивное начертание шрифта. Закрывающий тег обязателен.

**Синтаксис:**

**<i>Текст</i>**

**<em>Текст</em>**

# HTML язык

**<a>** - один из самых важных элементов в HTML и предназначен для создания ссылок. Для этого необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. Закрывающий тег обязателен. Имеет обязательные атрибуты.

## Синтаксис:

```
<a href="адрес">название</a>
```

```
<a href="http://google.com">Перейти на Google.com</a>
```

```
<a href="/contacts.html">Контакты</a>
```

## Обязательные атрибуты тега <a>:

href="адрес" - атрибут задает адрес документа или другого сайта, на который следует перейти.

## Другие атрибуты тега <a>:

target="\_blank" - атрибут, который позволяет открыть ссылку в новом окне. По умолчанию принимает значение "\_self".

Адрес ссылки может быть как абсолютным, так и относительным. Абсолютные адреса (внешние) должны начинаться с указания протокола (обычно http://) и содержать имя сайта. Относительные ссылки (внутренние) ведут отсчет от корня сайта или текущего документа.

# HTML язык

Элемент `<br>` устанавливает перевод строки в том месте, где он находится. Закрывающий тег не требуется. Является "пустым" тегом.

Элемент `<hr>` рисует горизонтальную линию. Закрывающий тег не требуется.



# HTML язык

Элемент `<img>` предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG, SVG или PNG. Адрес файла с картинкой задаётся через обязательный атрибут `src`. Имеет обязательные атрибуты.

## Синтаксис:

```
  
  

```

## Обязательные атрибуты `<img>`:

`src="адрес"` - адрес графического файла, который будет отображаться на веб-странице. Наиболее популярные файлы в формате PNG и JPEG.

## Необязательный, но важный атрибут:


`alt="текст"` - атрибут `alt` устанавливает альтернативный текст для изображений. Такой текст позволяет получить текстовую информацию о рисунке при отключенной в браузере загрузке изображений.



# HTML язык

## Другие атрибуты тега:

Для изменения размеров изображения средствами HTML предусмотрены универсальные атрибуты **height** и **width**.



```
height="размер"
```

```
width="размер"
```

```
height="50%"
```

```
height="20px"
```

```
width="80%"
```

```
width="20"
```

Допускается использовать значения в пикселях или процентах.



# HTML язык

`<!-- -->` - используется для создания комментариев в любой части документа. Все, что находится внутри `<!-- -->`, будь то элемент или текст – будет проигнорировано браузером (не будет обрабатываться и выводиться на экран).

## Синтаксис:

`<!-- Мой комментарий -->`

# HTML язык

## Таблицы

**<table>** - служит контейнером для создания таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью элементов **<tr>** и **<td>**, соответственно.

### Синтаксис:

```
<table border="1">  
  <tr>  
    <td>Ячейка 1</td>  
    <td>Ячейка 2</td>  
  </tr>  
  <tr>  
    <td>Ячейка 3</td>  
    <td>Ячейка 4</td>  
  </tr>  
</table>
```

Элемент **<td>** должен размещаться внутри контейнера **<tr>**, который в свою очередь располагается внутри **<table>**. Закрывающие теги обязательны.

# HTML язык

## Атрибуты <table>, <tr>, <td>:

`align="выравнивание"`

Задаёт выравнивание таблицы на странице, а для <tr> и <td> - выравнивание содержимого ячеек по горизонтали. Может принимать значения: left (по левому краю), right (по правому краю), center (по центру).

## Атрибуты <table>, <td>:

`width="ширина" и height="высота"`

Задают размер таблицы, а для <td> - размер ячеек (ширина и высота). Могут принимать только целые числовые значения в px или %. Обозначение "px" вместе с числом можно не указывать.

# HTML язык

## Атрибуты <table>

border="толщина"

Толщина рамки в пикселях (без px).

## Атрибуты <tr>, <td>:

valign="выравнивание"

Выравнивание содержимого ячеек по вертикали. Принимает значения: top (по верхнему краю), middle (по середине), bottom (по нижнему краю).

## Атрибуты <td>:

colspan="кол-во" и rowspan="кол-во"

Устанавливают число ячеек, которые должны быть объединены по горизонтали и вертикали, соответственно. Принимают только целые положительные числа.

# HTML язык

## Списки

При создании списков чаще всего используют теги `<ul>` (нумерованный или неупорядоченный список), `<ol>` (нумерованный или упорядоченный список) и `<li>` (пункт списка). Все эти теги парные.



```
<ul>
```

```
<li>Хлеб</li>
```

```
<li>Молоко</li>
```

```
<li>Чай</li>
```

```
</ul>
```

```
<ol>
```

```
<li>Позвонить другу</li>
```

```
<li>Сходить за покупками</li>
```

```
</ol>
```



# CSS стили

**Стилем или CSS** (Cascading Style Sheets, каскадные таблицы стилей) называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид.

**Если в языке HTML все элементы мы называем тегами или контейнерами, то в CSS - это селекторы.**

Подключение стилей к html документу возможно разными способами. Мы рассмотрим два часто используемых.



# CSS стили

## 1. Пример подключения стилей CSS с помощью тега <style> в области <head> (с комментариями и пояснениями – выделены розовым)

<head>

<style>

```
/* Общие стили для всего документа */
body {
  font-family: Arial, sans-serif; /* Шрифт для текста на странице */
  line-height: 1.6; /* Межстрочный интервал */
  margin: 20px; /* Внешние отступы вокруг содержимого страницы */
}
/* Стили для жирного текста */
b {
  font-size: 18px; /* Размер текста */
  font-weight: bold; /* Жирный текст */
  color: #333; /* Цвет текста (темно-серый) */
}
/* Стили для абзацев */
p {
  color: #666; /* Цвет текста (светло-серый) */
  margin: 10px 0; /* Внешние отступы сверху и снизу абзаца */
}
/* Стили для ссылок */
a {
  color: #007bff; /* Цвет ссылки (синий) */
  text-decoration: none; /* Отключение подчеркивания ссылки */
}
```

```
/* Стили для ссылок при наведении курсора */
a:hover {
  text-decoration: underline; /* Подчеркивание ссылки при наведении курсора */
}
/* Стили для изображений */
img {
  max-width: 20%; /* Максимальная ширина изображения (20% от ширины контейнера) */
  height: auto; /* Автоматически подстраивает высоту изображения для сохранения пропорций */
  display: block; /* Превращает изображение в блочный элемент, чтобы задать отступы */
  margin-top: 20px; /* Внешний отступ сверху изображения */
}
/* Стили для таблиц */
table {
  border-collapse: collapse; /* Объединяет границы ячеек таблицы, чтобы создать единую границу */
  width: 100%; /* Ширина таблицы (100% от ширины контейнера) */
  margin-top: 20px; /* Внешний отступ сверху таблицы */
}
/* Стили для упорядоченных списков */
ol {
  list-style-type: decimal; /* Маркер, цифра */
  margin-left: 20px; /* Отступ слева */
}
</style>
</head>
```



# CSS стили

## 2. Использование CSS внутри тегов через атрибут style="":

`<p style="font-size: 36px; color: white; background: green;">Рецепт печенья</p>`


Стиль будет применен только к содержимому этого тега!



# JavaScript


**Изначально JavaScript был создан, чтобы «сделать веб-страницы живыми».**

Программы на этом языке называются **скриптами**. Они могут встраиваться в HTML и выполняться автоматически при загрузке веб-страницы.



Скрипты распространяются и выполняются, как простой текст. Им не нужна специальная подготовка или компиляция для запуска.

В браузере для JavaScript доступно всё, что связано с манипулированием веб-страницами, взаимодействием с пользователем и веб-сервером.




# JavaScript

**Например, в браузере JavaScript может:**

- Добавлять новый HTML-код на страницу, изменять существующее содержимое, модифицировать стили.
- Реагировать на действия пользователя, щелчки мыши, перемещения указателя, нажатия клавиш.
- Отправлять сетевые запросы на удалённые сервера, скачивать и загружать файлы (технологии AJAX и COMET).
- Получать и устанавливать куки, задавать вопросы посетителю, показывать сообщения.
- Запоминать данные на стороне клиента («local storage»).

# JavaScript

Программы на JavaScript могут быть вставлены в любое место HTML-документа с помощью тега `<script>`. Тег `<script>` содержит JavaScript-код, который автоматически выполнится, когда браузер его обрабатывает.



```
1 <!DOCTYPE HTML>
2 <html>
3
4 <body>
5
6   <p>Перед скриптом...</p>
7
8   <script>
9     alert( 'Привет, мир!' );
10  </script>
11
12   <p>...После скрипта.</p>
13
14 </body>
15
16 </html>
```