

# Frontend Basic LESSON 1







## Анна Хачатурян

#### Front End/Gen Tech Teacher

- Since 2018 in IT
- Full Stack Developer at Web Magnat
- QA Engineer/Web Developer at Central Bank of RA
- Lecturer at Plekhanov Russian University of Economics
- TA at Picsart Academy
- Teacher at Starta Institute/Tel-Ran



### ВАЖНО:

начался блок вопросов.

Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас,

• Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.

также можно задать вопрос в чате или когда преподаватель скажет, что

- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.





## ПЛАН ЗАНЯТИЯ

- 1. Установка VS code
- 2. Клиент-серверное приложение
- 3. Front-end VS Back-end
- 4. Основы HTML/CSS
- 5. Практика







# 1

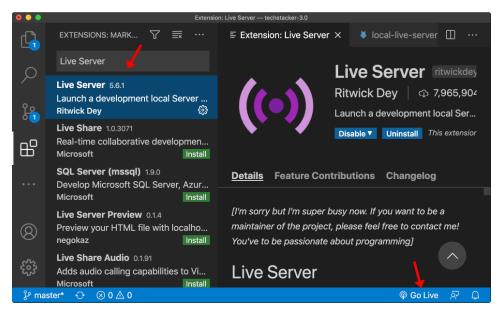
# Visual Studio code

### Установка VS code



https://code.visualstudio.com/

#### Плагин Live Server для VS Code







# 2

# КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ

## Что такое WEB?



Веб или вэб (англ. web — паутина, сеть) — «всемирная паутина/сеть», интернет-пространство:

Веб — система доступа к связанным между собой документам на различных компьютерах, подключённых к Интернету.





## Клиент-сервер



**Клиент**: Приложение, например, Chrome, Safari или Firefox, которое запущено на компьютере и подключено к Интернету. Его основная роль состоит в том, чтобы принимать пользовательские команды и преобразовывать их в запросы к другому компьютеру, называемому веб-сервером.

**Сервер**: Компьютер, который подключен к Интернету и также имеет IP-адрес. Сервер ожидает запросов от клиента/ов и отвечает на них. В отличие от клиента, который также имеет IP-адрес, на сервере установлено и работает специальное серверное программное обеспечение, которое подсказывает ему, как реагировать на входящие запросы от вашего браузера. Основной функцией веб-сервера является хранение, обработка и доставка веб-страниц клиентам.





# 3

# FRONT-END VS BACK-END



# Определение понятия front-end

**Front-end** — все, что браузер может читать, выводить на экран и / или запускать. То есть это **HTML**, **CSS** и **JavaScript**.

**HTML** говорит браузеру, каково содержание страницы, например, «заголовок», «параграф» и т.д.

**CSS** говорит браузеру, как отображать элементы, например, «после первого параграфа отступ в 20 пикселей».

**JavaScript** говорит браузеру, как реагировать на некоторые взаимодействия, используя легкий язык программирования.



- HTML5
- CSS3
- JavaScrypt
- jQuery

# Определение понятия back-end



- SQL
- C#
- ASP.NET
- PHP

**Back-end** — все, что работает на сервере, то есть «не в браузере» или «на компьютере, подсоединенном к сети (обычно к Интернету), который отвечает на сообщения от других компьютеров».

Для бэкенда вы можете использовать любые инструменты, доступные на вашем сервере (который, по сути, является просто компьютером, настроенным для ответов на сообщения). Это означает, что вы можете использовать любой универсальный язык программирования: Ruby, PHP, Python, Java, JavaScript / Node, bash. Это также означает, что вы можете использовать СУБД, такие как MySQL, PostgreSQL, MongoDB, Cassandra, Redis, Memc ached.



# 4

# Основы HTML/CSS

# Новый проект

Для создания нового проекта необходимо создать папку проекта.

#### В папке будет:

- файл index.html
- папка style с файлом style.css

index.html

Название главного HTMLдокумента в проекте style.css

Название css-документа, нужен для описания стилей элементов в HTMLдокументе



# Новый проект

**Базовая разметка** - дерево HTML-документов, которое включает в себя теги, которые есть в любом HTML-файле.

Эти теги и служебная информация нужны браузеру для корректного отображения информации.

Создается автоматически с помощью! + enter.

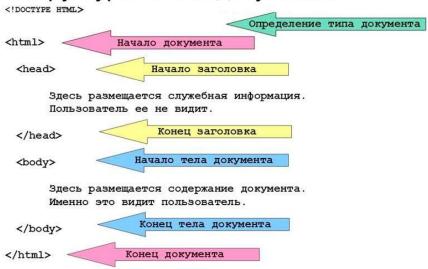
```
✓ style# style.css◇ index.html
```

Структура проекта



#### HTML

### Структура HTML документа





#### Теги

Основной и единственный структурный элемент языка HTML – это тег. Разберем их основные виды.

#### <имя тега> Содержимое </имя тега>

Такие теги называются парными, так как для них указаны маркеры начала и конца тега.

У тега могут быть **атрибуты**, которые указываются в его открывающей части и отделяются друг от друга пробелами.

Также существуют одиночные теги, для которых закрывающая часть отдельно не указывается.

#### Атрубуты

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

Атрибут имеет **имя** и **значение**. Имена атрибутов должны состоять из одного или нескольких символов, кроме управляющих, таких как пробел, ", ', >, / и =. Имена и значения атрибутов не чувствительны к регистру, но, тем не менее, рекомендуется набирать их в нижнем регистре.

Некоторые атрибуты не требуют значение, потому что у них есть только одна опция. Они называются логическими атрибутами.

#### Заголовки

Ter <h1> — это обязательный заголовок для всех страниц. Это может быть название страницы, статьи или товара, которое несет конкретную информацию, о чем будет читать пользователь.

Заголовок <h2> является подзаголовком <h1> и имеет меньшее значение, чем <h1>.

Теги <h3>-<h6> включают в себя уточняющую информацию. Использование всех заголовков встречается крайне редко. Это должна быть огромная статья с выстроенной иерархией.

#### Параграф

Для разметки параграфов предназначен тег (от английского «paragraph»).

По умолчанию абзацы начинаются с новой строки и отделяются от остального контента отступами сверху и снизу. Так что, если нужно отделить один блок текста от другого, верным решением будет заключить их в теги .

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Recusandae, voluptatum dignissimos alias sapiente sed necessitatibus enim quam, quidem optio saepe animi nesciunt, inventore officiis eveniet tenetur placeat expedita id? Sapiente!



#### Еще теги для текстов

Ter pre> представляет собой предварительно отформатированный текст, который должен быть представлен точно так, как написано в HTML-файле. Текст обычно отображается с использованием непропорционального ("monospace") шрифта. Пробелы внутри этого элемента отображаются как записанные.

Ter **<span>** является основным строковым контейнером для фразового контента, который, по существу, ничего не представляет. Он может использоваться для группировки элементов в целях стилизации. Например, внутри абзаца (тега **сроку на сроку на** 

Lorem ipsum dolor <span>sit</span> amet adipisicing elit. Iste, dolore quo! Nobis ullam ea blanditiis? Iste, fugiat magnam dolorem fuga amet omnis officiis ullam delectus unde ea libero aliquam quia.



#### Ссылка

Ter <a> является одним из важных элементов HTML и предназначен для создания ссылок. Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. В качестве значения атрибута href используется адрес документа (URL, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход.

Есть еще один важный атрибут для <a> тега, это **target**. По умолчанию, при переходе по ссылке документ открывается в текущем окне. При необходимости, это условие может быть изменено атрибутом target тега <a>.

\_blank - Загружает страницу в новое окно браузера. \_self - Загружает страницу в текущее окно. По умолчанию

```
<a href="https://www.google.com/" target="_blank">Google</a>
Click <a href="https://github.com/" target="_blank">here</a>
```

#### Картинка

Ter <img> предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Адрес файла с картинкой задаётся через атрибут src. Из главных атрибутов является alt — альтернативный текст. Такой текст позволяет получить текстовую информацию о рисунке при отключенной в браузере загрузке изображений.

Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег <img> в контейнер <a>.

### Отличия блочных и строчных элементов

Выделяют две основные категории HTML-элементов, которые соответствуют типам их содержимого и поведению в структуре веб-страницы — блочные и строчные элементы.

#### Блочные элементы

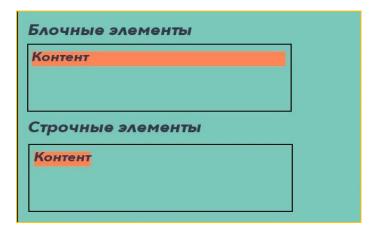
Блочные элементы являются основой, которая используется для верстки веб-страниц. Такой элемент представляет собой прямоугольник, который по умолчанию занимает всю доступную ширину страницы (если иное значение не указано в CSS), а длина элемента зависит от его содержимого. Такой элемент всегда начинается с новой строки, то есть, располагается под предыдущим элементом. Блочный элемент может содержать в себе другие блочные и строчные элементы.

Примеры блочных элементов: , <h1>-<h6> , <div> и т. д.

#### Строчные элементы

В отличие от блочного, строчный элемент не переносится на новую строку, а располагается на той же строке, что и предыдущий элемент. Такие элементы, как правило, находятся внутри блочных элементов, и их ширина зависит лишь от содержимого и настроек CSS. Еще одно отличие строчного элемента от блочного заключается к том, что в нем может находиться только контент и другие строчные элементы. Блочные элементы в строчные вкладывать нельзя.

Примеры строчных элементов: <a>, <span>, <img> и т. д.



## Отличия блочных и строчных элементов

Даже если мы изменим ширину блочного элемента, то все равно с правой стороны ничего располагаться не будет.





## Работа со стилями

Подключаем css-файл к html-документу внутри тега <head></head> c помощью тега <link>.

В атрибуте href надо прописать путь к файлу css.



```
Селектор

p {
    color: orange;
    font-size: 16px;
}
```

#### CSS Свойства

B CSS **цвет текста** задается с помощью свойства **color**, а его значение можно записать несколькими способами – в шестнадцатеричном (hex) виде, в формате RGB или HSL либо указав ключевое слово.

Размер шрифта определяется свойством font-size. Можно задать размер шрифта в пикселях.

Text-decoration добавляет оформление текста в виде его подчеркивания, перечеркивания, линии над текстом и мигания. Одновременно можно применить более одного стиля, перечисляя значения через пробел.

```
span{
   color: red;
   font-size: 20px;
}
```

```
.p3{
color: red;
}
```

#### CSS-селекторы

Селектор определяет, к какому элементу применять то или иное CSS-правило.

• Селекторы по типу элемента (название тега)

Этот базовый селектор выбирает тип элементов, к которым будет применяться правило.

Синтаксис: название тега

Пример: селектор р выберет все элементы .

```
p{
   font-size: 20px;
   color: red;
}
```

#### • Селекторы по классу

Этот базовый селектор выбирает элементы, основываясь на значении их атрибута class.

Синтаксис: .имяКласса

Пример: селектор .paragraph2 выберет все элементы с соответствующим классом (который был определён в атрибуте class=" paragraph2").





#### HTML

#### **CSS**

```
p{
    font-size: 20px;
    color: red;
}
.paragraph2{
    color: blue;
    font-size: 20px;
}
```



#### • Селекторы по идентификатору

Этот базовый селектор выбирает элементы, основываясь на значении их id атрибута. Не забывайте, что идентификатор должен быть уникальным, т. е. использоваться только для одного элемента в HTML-документе.

Синтаксис: #имяИдентификатора

Пример: селектор #color\_red выберет элемент с идентификатором color\_red (который был определён в атрибуте id="color\_red").

#### HTML

#### CSS

```
#color_red{
    color: red;
    font-size: 18px;
}
```



## Полезные ссылки

- Клиент-серверное приложение
- Ссылка для загрузки vs code
- <u>Многие ученики испытывают трудности с немецкой клавиатурой, поэтому скидываем тренажер для тренировки</u>
- Шпаргалка по emmet vs code

