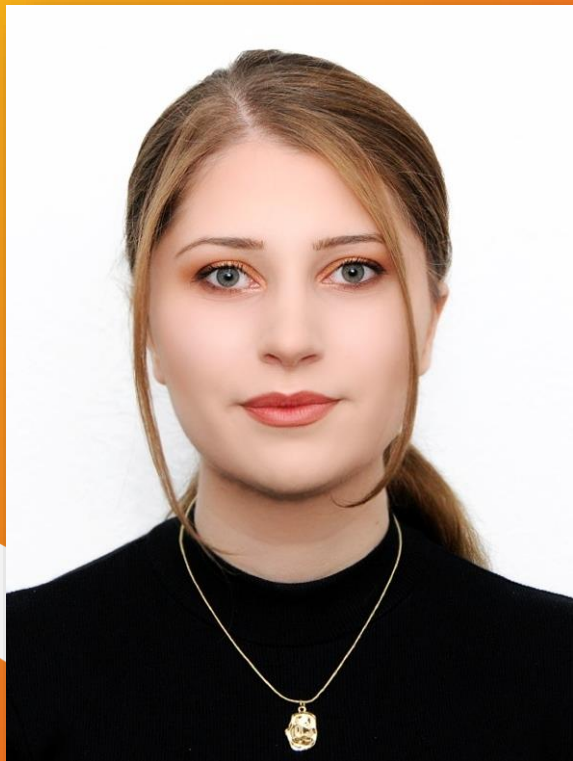


# Frontend Basic

## LESSON 1



Anna Khachaturyan



# 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 Tel-Ran

[https://t.me/anny\\_khachaturyan](https://t.me/anny_khachaturyan)



# ВАЖНО:

- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- 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

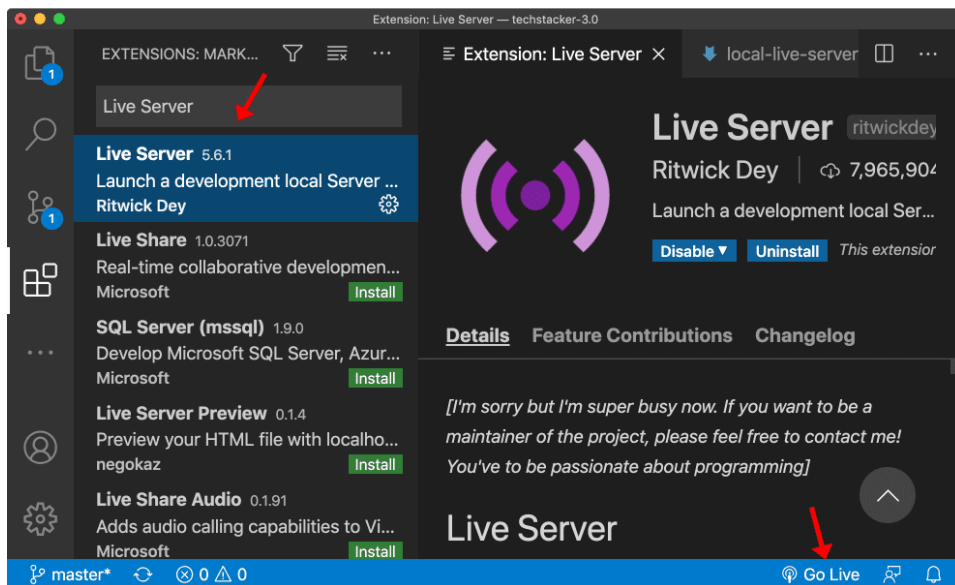
1

# 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.





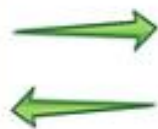
# Клиент-сервер

**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.

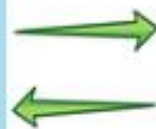
**Сервер:** Компьютер, который подключен к Интернету и также имеет IP-адрес. Сервер ожидает запросов от клиента/ов и отвечает на них. В отличие от клиента, который также имеет IP-адрес, на сервере установлено и работает специальное серверное программное обеспечение, которое подсказывает ему, как реагировать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



С ним работает  
пользователь



Тут хранится код



Тут лежат данные

# 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
- JavaScript
- 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».

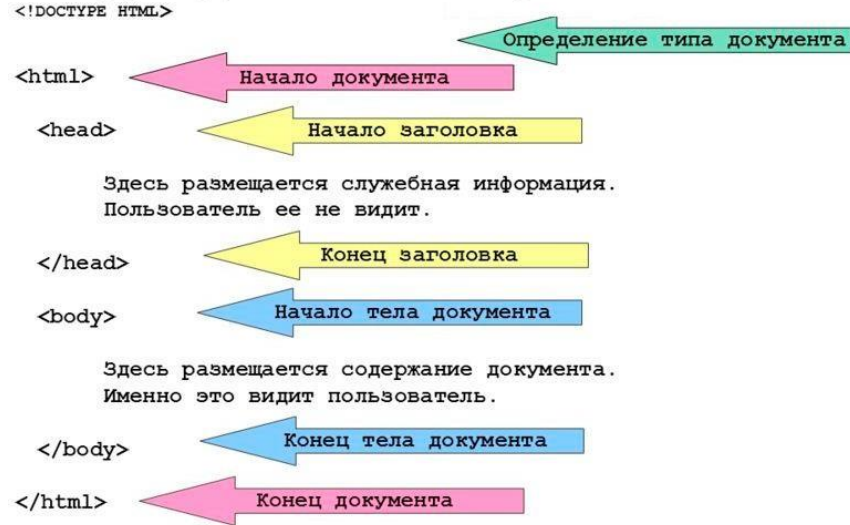
Для бэкенда вы можете использовать любые инструменты, доступные на вашем сервере (который, по сути, является просто компьютером, настроенным для ответов на сообщения). Это означает, что вы можете использовать любой универсальный язык программирования: **Ruby, PHP, Python, Java, JavaScript / Node, bash**.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

# 4

## ОСНОВЫ HTML/CSS

## 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.

## Атрибуты

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 значения, потому что у них есть только одна опция. Они называются логическими атрибутами.



## Заголовки

```
<h1>My first blog</h1>
  <h2> Level 2</h2>
  <h3> Level 3</h3>
  <h4> Level 4</h4>
  <h5> Level 5</h5>
  <h6> Level 6</h6>
```

Тег `<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 `<p>` (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 `<p>`.

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

## Еще теги для текстов

Тег `<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.

```
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Earum illum velit  
re sapiente.  
>
```

Dolor blanditiis quam laborum quos ea architecto saepe perspiciatis veritatis dolorem? Doloribus ullam necessitatibus ratione nesciunt sapiente possimus.

Тег `<span>` является базовым элементом контента для форматирования контента, который, по существу, ничего не представляет. Он может использоваться для группировки элементов в целях стилизации. Например, внутри абзаца (тега `<p>`) можно изменить цвет и размер слова, если добавить начальный и конечный тег `<span>` и определить для него стиль текста.

```
<p>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.</p>
```

## Перевод строки

Тег `<br>` устанавливает перевод строки в том месте, где этот тег находится. В отличие от тега абзаца `<p>`, использование тега `<br>` не добавляет пустой отступ перед строкой.

```
<span>Text Text Text</span>  
<br>  
<span>Text Text Text</span>
```



## CSS Свойства

В 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;  
}
```

## 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 `p` выберет все элементы `<p>`.

```
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

```
<p>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Accusantium optio voluptatibus architecto consequatur nobis quo accusamus asperiores aspernatur at voluptate soluta doloremque, corporis explicabo qui necessitatibus veniam alias, dolores dignissimos.</p>
```

```
<p class="paragraph2">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Accusantium optio voluptatibus architecto consequatur nobis quo accusamus asperiores aspernatur at voluptate soluta doloremque, corporis explicabo qui necessitatibus veniam alias, dolores dignissimos.</p>
```

```
<p class="paragraph2">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Accusantium optio voluptatibus architecto consequatur nobis quo accusamus asperiores aspernatur at voluptate soluta doloremque, corporis explicabo qui necessitatibus veniam alias, dolores dignissimos.</p>
```

## CSS

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

- 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

```
<p id="color_red">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Accusantium optio voluptatibus architecto consequatur nobis quo accusamus asperiores aspernatur at voluptate soluta doloremque, corporis explicabo qui necessitatibus veniam alias, dolores dignissimos.</p>
```

## CSS

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

**Кто молодцы?**



**Мы молодцы!**