МШП 20.35 УНИВЕРСИТЕТ КОД БУДУЩЕГО Минцифры России НАЦИОНАЛЬНЫЕ ПРОЕКТЫ РОССИИ ЦИФРОВАЯ ЭКОНОМИКА

Fullstack-разработка

Vue
Введение

МШП

Цели урока:

1. Создадим проект на Vue
2. Используем 3 кита Javascripta внутри компонента

МШП

Теория

Vue

МШП

Vue – это **фреймворк** для создания пользовательских интерфейсов

- Простой
- Ускоряет разработку
- Берёт работу с DOM на себя

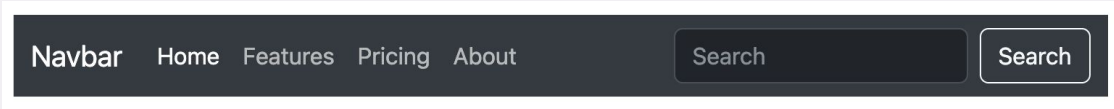
МШП

Фреймворк предлагает определённый способ писать код, что облегчает и ускоряет разработку.

Это как купить путёвку вместо планирования путешествия самостоятельно.

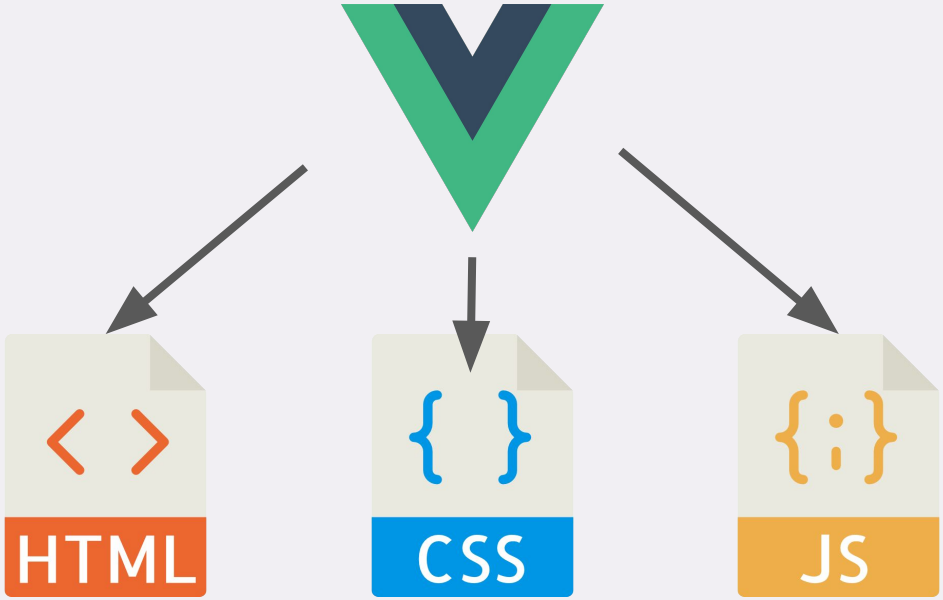
МШП

В основе Vue лежат **компоненты** – элементы интерфейса, которые можно **переиспользовать.**



МШП

Каждый компонент хранится в отдельном файле с расширением .vue



МШП

HelloWorld.vue

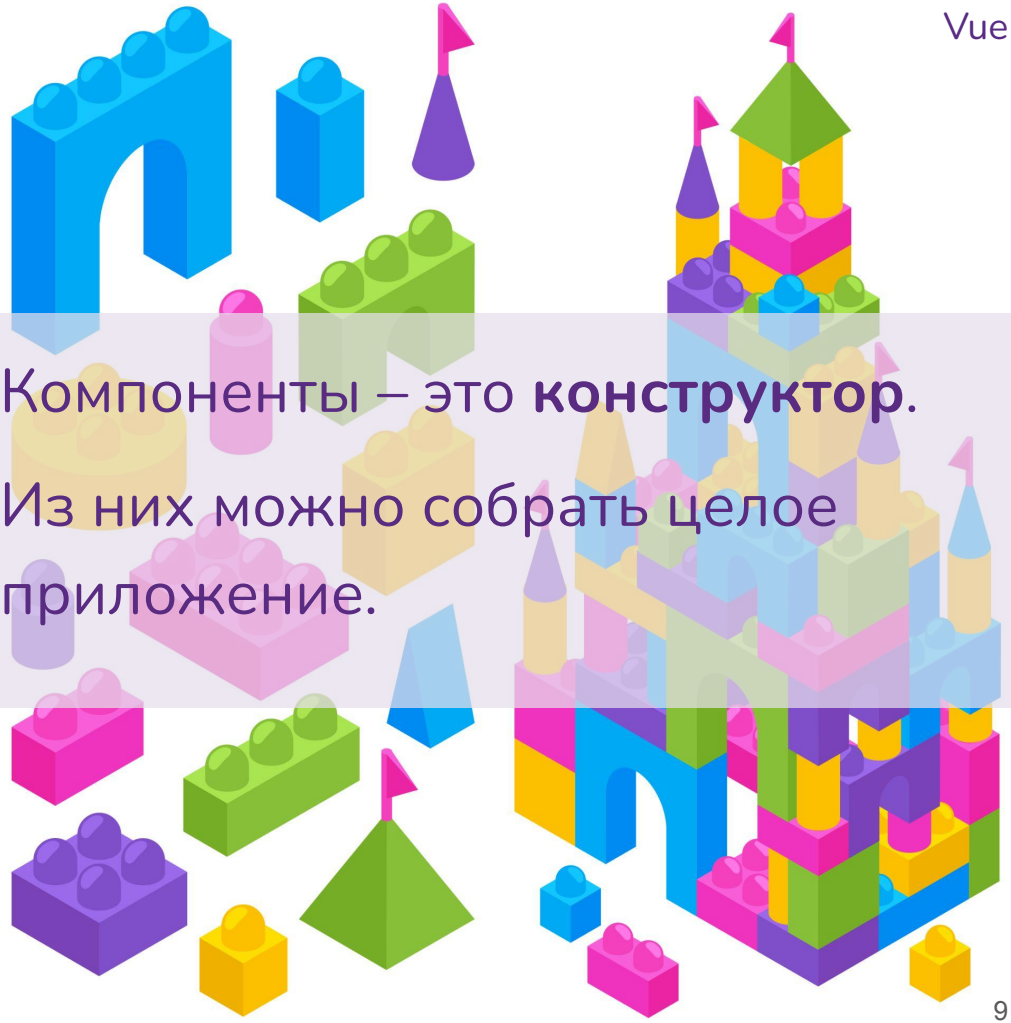
Каждый компонент состоит из трёх секций-тегов:

```
<template>
    <h1>Привет, мир!</h1>
</template>

<style>
h1 { font-size: 40px; }
</style>

<script>
...
</script>
```

МШП



Компоненты – это **конструктор**.

Из них можно собрать целое приложение.

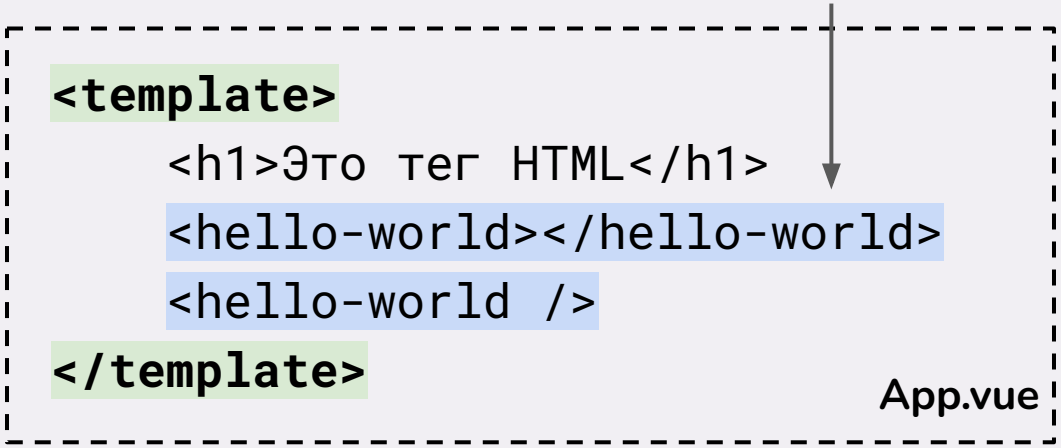
Компонент **App.vue** – как платформа для конструктора компонентов

МШП

App.vue

В него можно подключить другие компоненты и обращаться к ним как к тегам

HelloWorld.vue

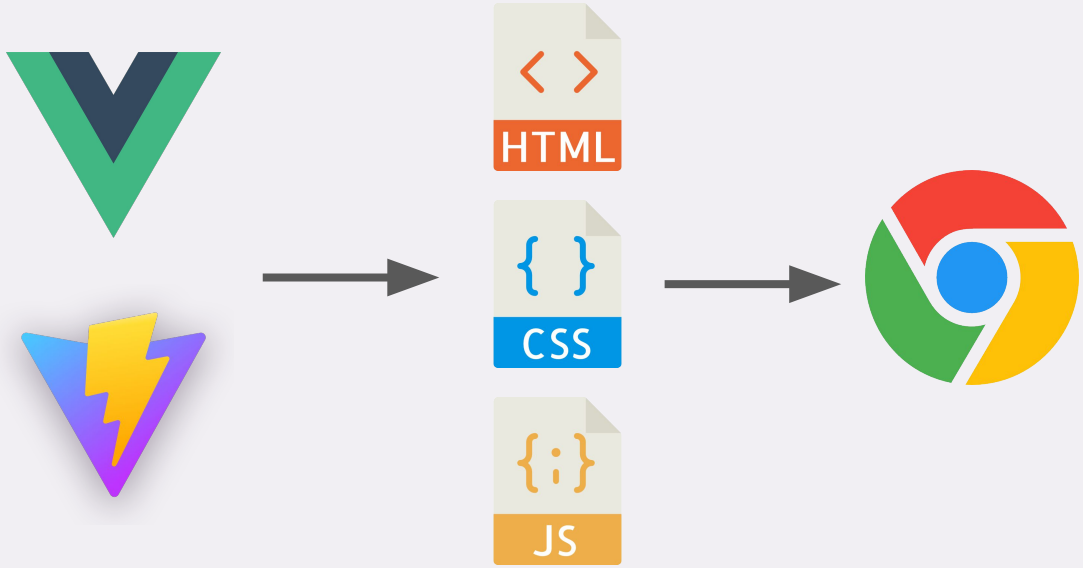


Когда пользователь откроет наш сайт, он увидит именно **App.vue**

МШП

У Vue свой уникальный синтаксис, который не понятен браузеру.

Поэтому нам нужен **сборщик Vite.**



МШП

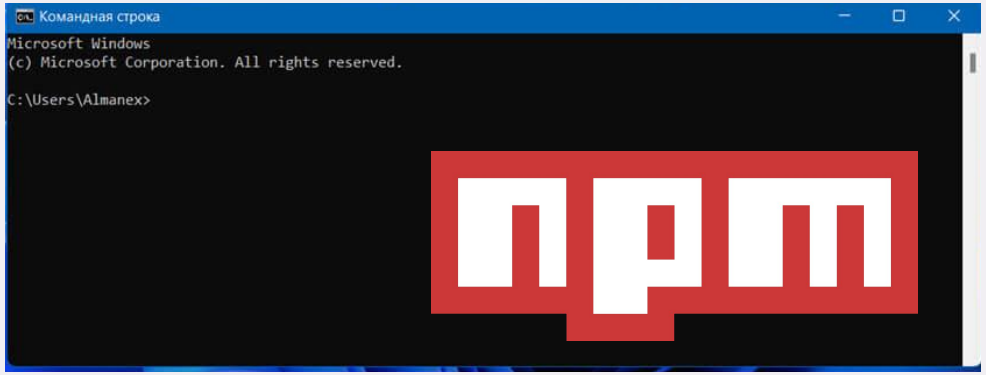
Задача сборщика – перевести компоненты Vue на понятный браузеру HTML, CSS и Javascript.

Он также запускает локальный сервер, который делает файлы доступными на localhost.

МШП

Vite и **Vue** это библиотеки.

Так что нам понадобится...



МШП

Теория и Практика

Создание проекта

МШП

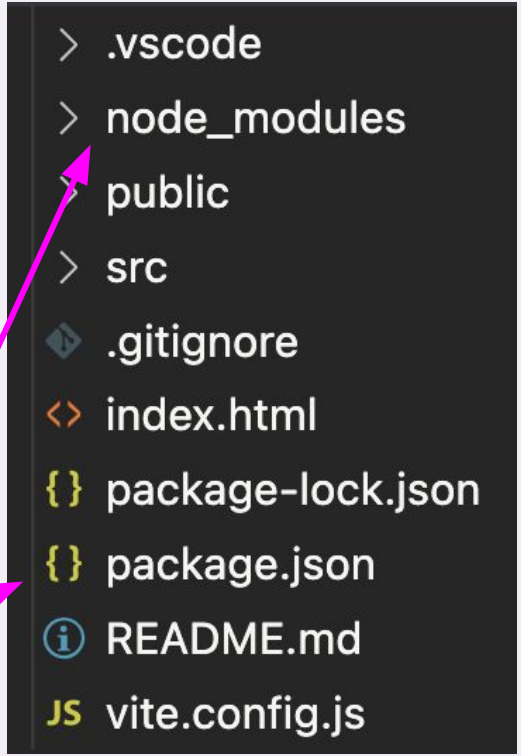
Создание проекта

В папке со всеми проектами выполни команду:

```
npm create vue@3 название
```

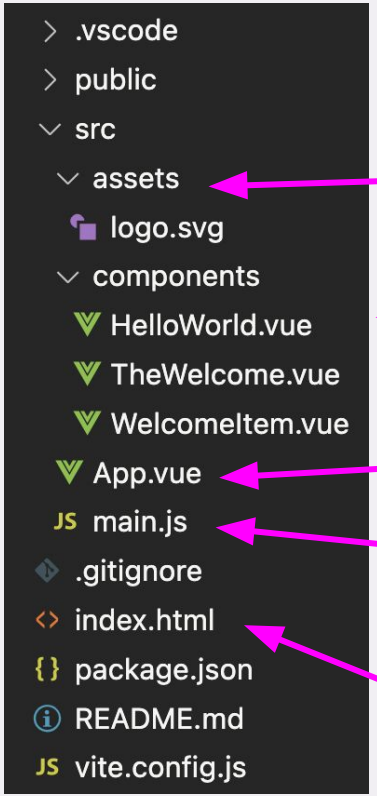
Открой проект папку проекта в VSCode.
Установи пакеты:

```
npm install
```



МШП

Файлы проекта



Изображения

Компоненты (с расширением ***.vue**)

То, что видит пользователь

Здесь код, который подставляет **App.vue** в **index.html**

Обычный index.html

МШП

Монтирование приложения

```
import { createApp } from 'vue';
import App from './App.vue';

createApp(App).mount('#app');
```

```
<body>
  <div id="app">
    <!-- Здесь будет App.vue -->
  </div>

  <script type="module" src="/src/main.js">
  </script>
</body>
```

main.js

index.html

МШП

Теория и Практика

Шапка сайта и подвал

МШП

Подключение компонента

Импорт

```
<script>
  import ComponentOne from './components/ComponentOne.vue';
  import ComponentTwo from './components/ComponentTwo.vue';

  export default {
      components: {
          ComponentOne,
          ComponentTwo
      }
  }
</script>
```

Использование

```
<template>
    <component-one> </component-one>
    <component-two />
</template>
```

МШП

Название компонентов

1. Состоит из двух слов
2. Записано в CamelCase
3. Не может повторять HTML-теги

МШП

Теория

Обработчики события

МШП

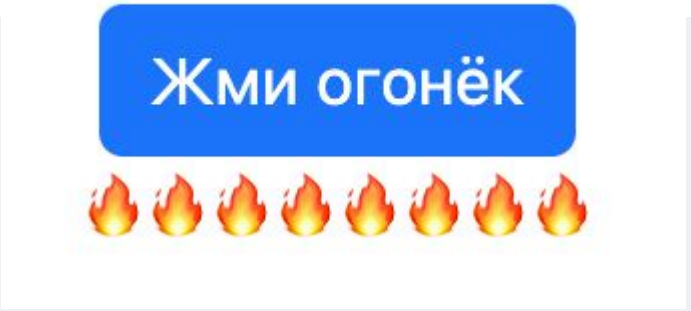
3 кита JavaScripta

1. **Найти элемент**
   document.querySelector
2. **Изменить элемент**
   node.innerHTML
3. **Реагировать на событие Клик**
   button.addEventListener

МШП

Больше мы этого не увидим)

```
1  console.log("Hello, world!");
2
3  let outputNode = document.querySelector(`#output`);
4  let buttonNode = document.querySelector(`#click`);
5
6  buttonNode.addEventListener(`click`, function () {
7    outputNode.innerHTML += `🔥`;
8  });
9
```



МШП

FireButton.vue

Создадим новый компонент с вёрсткой кнопки.

```
<template>
    <button class="btn btn-primary">
        Жми огонёк
    </button>
</template>
```

МШП

Событие

События в Vue это **директива** – специальный **атрибут**, который можно добавить любому тегу

@событие

- @click
- @input
- @keydown

МШП

Метод

Методы компонента описывается в объекте **methods**.

```
<script>
export default {
    methods: {
        fire() {
            console.log('огонёк!');
        },
        hello() {
            console.log('привет');
        }
    }
}
</script>
```

МШП

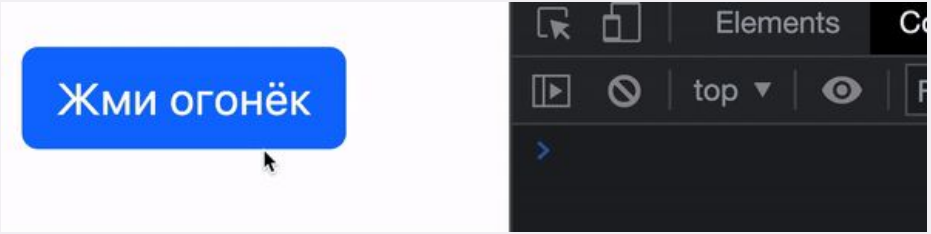
Подписка на событие

Событие и **метод** легко соединяются внутри шаблона компонента.

```
<template>
    <button @click="fire"
            class="btn btn-primary">
        Жми огонёк
    </button>
</template>
```

МШП

Через директиву **@click и метод** компонента устанавливается связь между событием и действием.



МШП

data

Кроме методов у компонента есть **данные.**

Это **объект** со свойствами для шаблона.

```
<script>
export default {
    data() {
        return {
            username: 'Никита',
            age: 15
        }
    }
}
</script>
```

МШП

data

Данные можно выводить в шаблон через наши любимые **фигурные скобки**.

```
<template>
    <h1>
        Меня зовут {{ username }},
        мне {{ age }} лет.
    </h1>
</template>
```

МШП

FireButton.vue

Допустим, у кнопки есть **data-свойство** text.

```
<template>
    <button class="btn btn-primary">
        Жми огонёк
    </button>
    <p> {{ text }} </p>
</template>
```

МШП

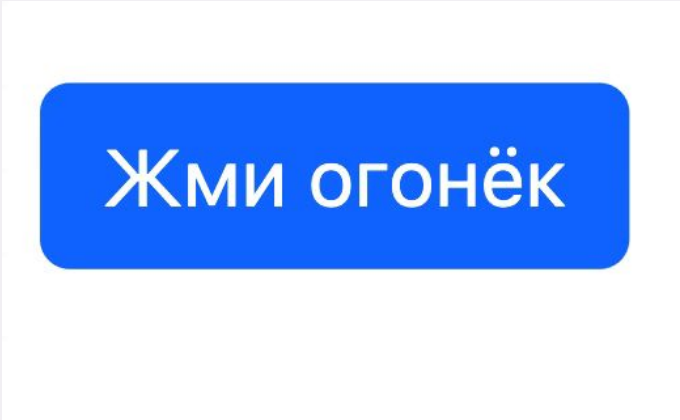
data

С помощью метода можно обратиться к свойству через **this** и изменить его.

```
<script>
export default {
    methods: {
        fire() {
            this.text += `🔥`
        }
    }
}
</script>
```

Изменение свойства **data** автоматически приводит к перерисовке компонента.

Это называется **реактивность.**



4 кита Vue

1. **Объявить data**
2. **Вывести data в шаблон**
   {{ text }}
3. **Изменить data в методе**
   fire() { this.text += "🔥"; }
4. **Связать метод и событие**
   @click="fire"

МШП

Практика

Вопрос, счётчик и галерея

МШП

Цели урока:

- ✓ Создали проект на Vue
- ✓ Использовали 3 кита Javascripta внутри компонента

МШП

Домашнее задание