



vue.js

Vue.js

Vue (/vju:/, как **view**) - это прогрессивный фреймворк для создания пользовательских интерфейсов.

Что такое Vue?

- Это JavaScript фреймворк
- У него есть Virtual DOM и реактивные компоненты
- Имеет легкое ядро и хорошо работает со встроенными библиотеками для роутинга, глобального состояния и HTTP запросов.
- **Официально документирован на русском!**
- Легкий, быстрый, гибкий

В чем разница

Подробности: <https://ru.vuejs.org/v2/guide/comparison.html>

Синтаксис

```
<div id="app">
  {{ message }}
</div>
<script src="https://unpkg.com/vue/dist/vue.js"></script>
<script>
  new Vue({
    el: '#app',
    data: {
      message: 'Hello World!'
    }
  })
</script>
```

Что мы только что сделали?

- Создали экземпляр Vue который связан с элементом по id="app"
- Представили контент этого элемента в качестве шаблона (template)
- Применили текстовую интерполяцию с использованием синтаксиса Mustache (двойные фигурные скобки)

???



Установка

- `<script>`
- CDN
- npm
 - vue-cli

Content Delivery Network (CDN)

```
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
```

Начало работы

Вспоминаем что такое:

- HTML
- CSS
- JavaScript



HTML



JavaScript



CSS

DOM

```
<!DOCTYPE html>
<html>
<head>
  <title>
    DOM
  </title>
</head>
<body>
  <div>
    This is simple Document Object Model (DOM)
  </div>
</body>
</html>
```

Что нам дает Vue

Декларативная отрисовка

HTML

```
<div id="app">
  {{ message }}
</div>
```

JavaScript

```
var app = new Vue({
  el: '#app',
  data: {
    message: 'This is simple
Document Object Model (DOM)'
  }
})
```

This is simple Document Object Model (DOM)

Декларативная отрисовка 2

```
<div id="app">  
  <span v-bind:title="message">  
    Hover me for few seconds  
  </span>  
</div>
```

```
var app = new Vue({  
  el: '#app',  
  data: {  
    message: 'Page loaded at: ' +  
    new Date().toLocaleString()  
  }  
})
```

Hover me for few seconds

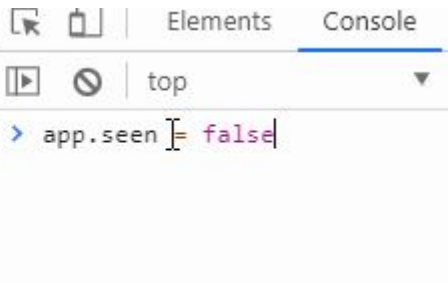


Циклы и условия

```
<div id="app">  
  <span v-if="seen">You can see me</span>  
</div>
```

```
var app = new Vue({  
  el: '#app',  
  data: {  
    seen: true  
  }  
})
```

You can see me



Пользовательский ввод

```
<div id="app">
  <p>{{ message }}</p>
  <button
v-on:click="reverseMessage">Reverse</button>
</div>
```

```
var app = new Vue({
  el: '#app',
  data: {
    message: 'Hello Vue.js!'
  },
  methods: {
    reverseMessage: function () {
      this.message =
this.message.split('').reverse().join('')
    }
  }
})
```

Hello Vue.js!

Reverse

Еще раз. Что мы можем делать

Интерполяция строк {{ propertyName }} (mustache)

Выражения {{ signedUp ? 'Cancel' : 'Sign up' }}

Биндить атрибуты (v-bind): `<div v-bind:id="superId"></div>`

Фильтры: {{ slug | capitalize }}

И ещё несколько директив: v-if, v-else, v-show, v-for, v-on:event...

Методы

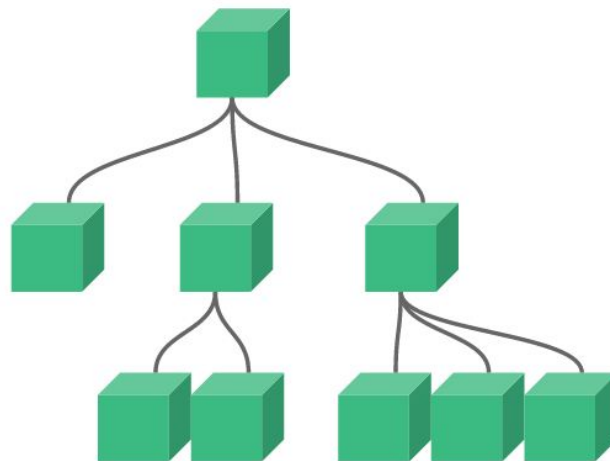
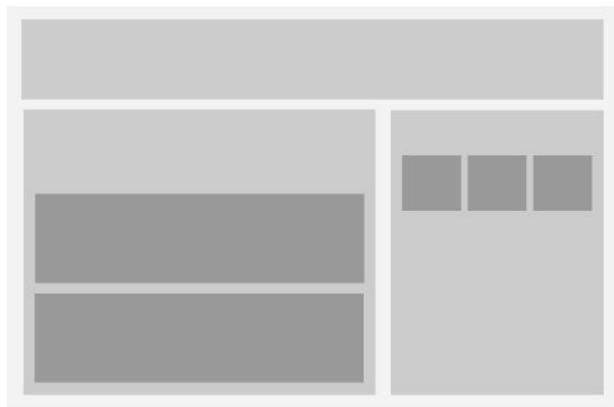
```
<div id="app">
  {{ showThing() }}
</div>
<script>
new Vue({
  el: '#app',
  data: {
    message: 'Hello World!'
  },
  methods: {
    showThing: function () {
      return `Thing shown: ${this.message}`
    }
  }
})
</script>
```

Вычисляемые свойства

```
<div id="example">
  <p>Изначальное сообщение: «{{ message }}»</p>
  <p>Сообщение задом наперёд: «{{ reversedMessage
}}»</p>
</div>
```

```
var vm = new Vue({
  el: '#example',
  data: {
    message: 'Привет'
  },
  computed: {
    // геттер вычисляемого значения
    reversedMessage: function () {
      // `this` указывает на экземпляр
      vm
      return
      this.message.split('').reverse().join('')
    }
  }
})
```

Компоненты



Создание и использование компонентов

Каждый компонент это маленький экземпляр Vue. Создадим глобальный компонент с именем “todo-item”:

```
Vue.component('todo-item', {  
  props: ['todo'],  
  template: '<li>{{ todo.text }}</li>'  
})
```

```
<ol>  
  <todo-item v-for="item in todos" v-bind:todo="item"></todo-item>  
</ol>
```

Заметка. Переменные.

Обратите внимание что при создании переменных в JavaScript используется camelCase, в HTML kebab-case.

О чем мы забыли упомянуть

Routing

Vue-Router: <https://router.vuejs.org/ru/>

State management

<https://vuex.vuejs.org/ru/>

Single-file syntax (Vueify)

```
<template>
  <div class="my-component">{{ message }}</div>
</template>
```

```
<script>
export default {
  data: {
    message: "Hello World!"
  }
};
</script>
```

```
<style>
.my-component {
  border: 1px solid ghostwhite;
}
</style>
```

QA