

# Компоненты во Vue.js

---

JS  
COURSE  
ORT DNIPRO

---

ORT**DNIPRO**.ORG/JS

# Vue.js

# Vue.js – фреймворк для разработки SPA



The Progressive  
JavaScript Framework



WHY VUE.JS?

GET STARTED



GITHUB

Special Sponsor



**Autocode**

Build app-to-app workflows and connect APIs

## Approachable

Already know HTML, CSS and JavaScript? Read the guide and start building things in no time!

## Versatile

An incrementally adoptable ecosystem that scales between a library and a full-featured framework.

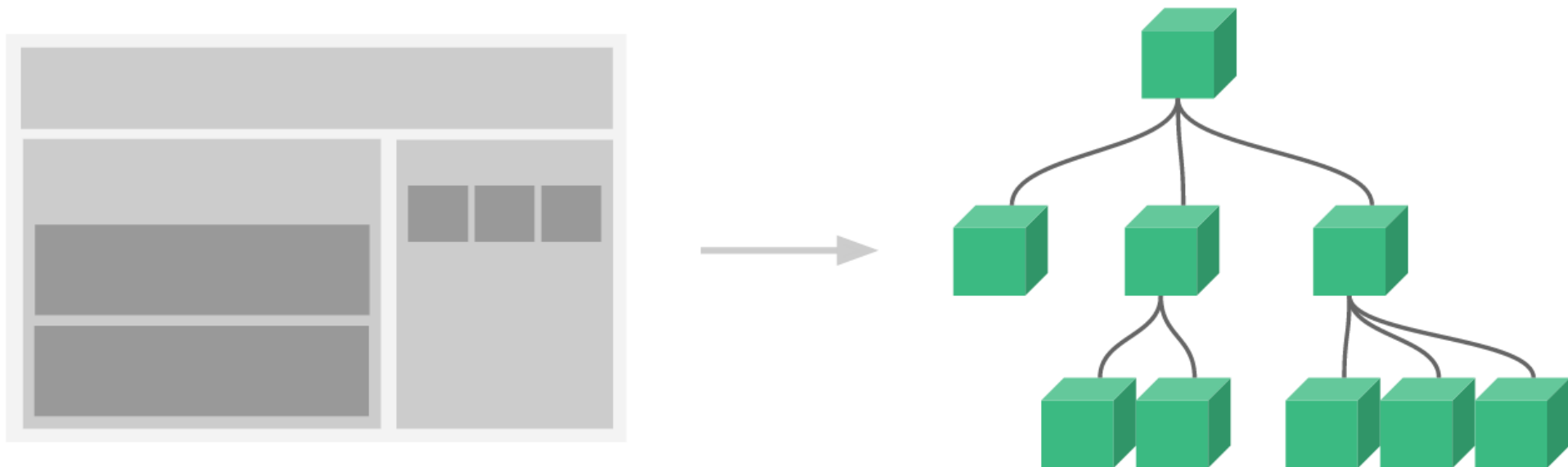
## Performant

20KB min+gzip Runtime  
Blazing Fast Virtual DOM  
Minimal Optimization Efforts

<https://v3.ru.vuejs.org/ru/>

# 1. Компоненты во Vue.js

# Компоненты – разделение приложения на блоки



Идея **компонентов Vue.js** аналогична идеи **компонентов** в библиотеке **Bootstrap** – отдельные, относительно независимые, части приложения которые можно переиспользовать.

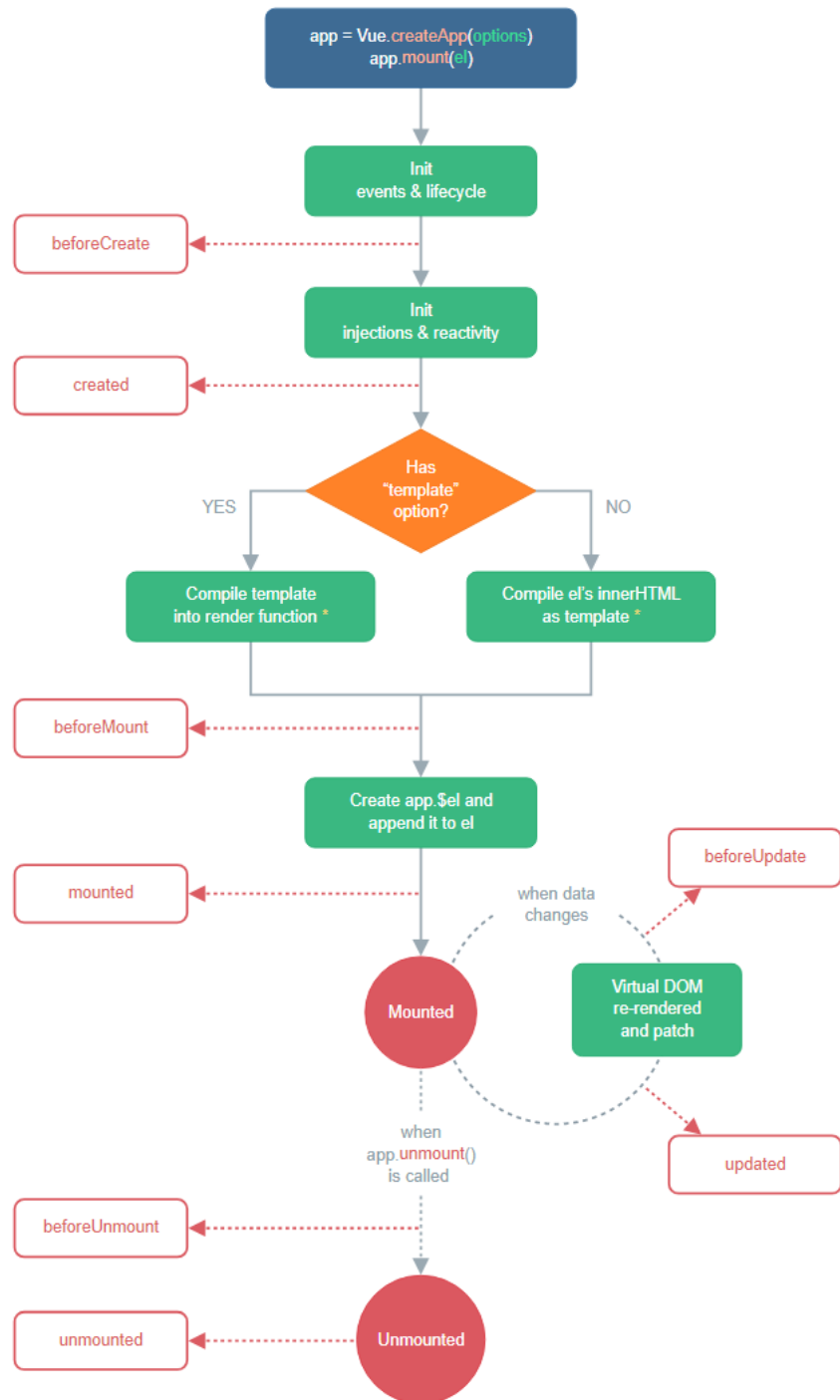
<https://v3.ru.vuejs.org/ru/guide/component-basics.html>

## **2. Жизненный цикл компонента (приложения)**

# Жизненный цикл компонента (и приложения)

Этапы создание и функционирования каждого компонента (и приложения), представляют интерес тем, что можно связать свой код с тем или иным хуком жизненного цикла. Одним из самых полезных является хук **mounted()** позволяющий запустить код когда компонент перемонтирован в документ и отображается пользователю (что полезно для запуска AJAX-запросов и прочих «долгих» действий).

<https://v3.ru.vuejs.org/ru/guide/instance.html#диаграмма-жизненного-цикла>



# 3. План на сегодња



# План на сегодня

## 1. Структура компонента

<https://v3.ru.vuejs.org/ru/guide/component-basics.html>

## 2. Передача параметров в компоненты

<https://v3.ru.vuejs.org/ru/guide/component-basics.html#организация-компонентов>

## 3. Получение информации из компонента (emit-события)

<https://v3.ru.vuejs.org/ru/guide/component-basics.html#прослушивание-событий-из-дочерних-компонентов-в-родительских-компонентах>

## 4. Файловая структура проекта с множеством компонентов

Воспользуйтесь шаблоном в репозитории занятия [./src/demo-template](#)

## 4. «Загрузка» приложения

# «Загрузка» приложения и директива **v-cloak**

```
13 <body>
14   <div id="app" v-cloak>
15     <h1>{{title}}</h1>
16   </div>
17   <div id='app-loading' >
18     <i class="fas fa-spinner fa-spin"></i>
19   </div>
20
21 </body>
```

```
1
2 [v-cloak] {
3   display: none;
4 }
5
6 #app-loading {
7   display: none;
8 }
9
10 [v-cloak] + #app-loading {
11   display: block;
12 }
13
14
```

Воспользуйтесь шаблоном в репозитории занятия [./src/app-load-imitation](#)

При помощи директивы **v-cloak** и небольшого фрагмента стилей есть возможность скрыть разметку приложения, до того момента пока оно не будет смонтировано и готово к работе. Подробнее: <https://v3.ru.vuejs.org/ru/api/directives.html#v-cloak>

**К следующему занятию...**

**Система сборки  
Vue-приложений**

**Домашнее задание**  
**/сделать**

# Домашнее задание #S.1

Воспользуйтесь разделом API <https://fakestoreapi.com/products> для формирования страницы со списком товаров. Реализуйте вывод списка товаров, а также обеспечьте работу **сортировки** по цене и **поиска** которые **должны работать одновременно**. **Изначально** (после загрузки страницы) **должен выводиться список в той же последовательности, в которой пришли данные от API**.

На каждой карточке товара должен быть **счётчик количества**. А в верхней части страницы указание **сколько суммарно единиц товаров** (всех) выбрал пользователь. И какая **суммарная стоимость** всех выбранных товаров. Всё это должно работать в динамике. **Vue и компоненты вам в помощь**.

