

# Vue.js и система сборки Vue CLI

---

JS  
COURSE  
ORT DNIPRO

---

ORTDNIPRO.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.

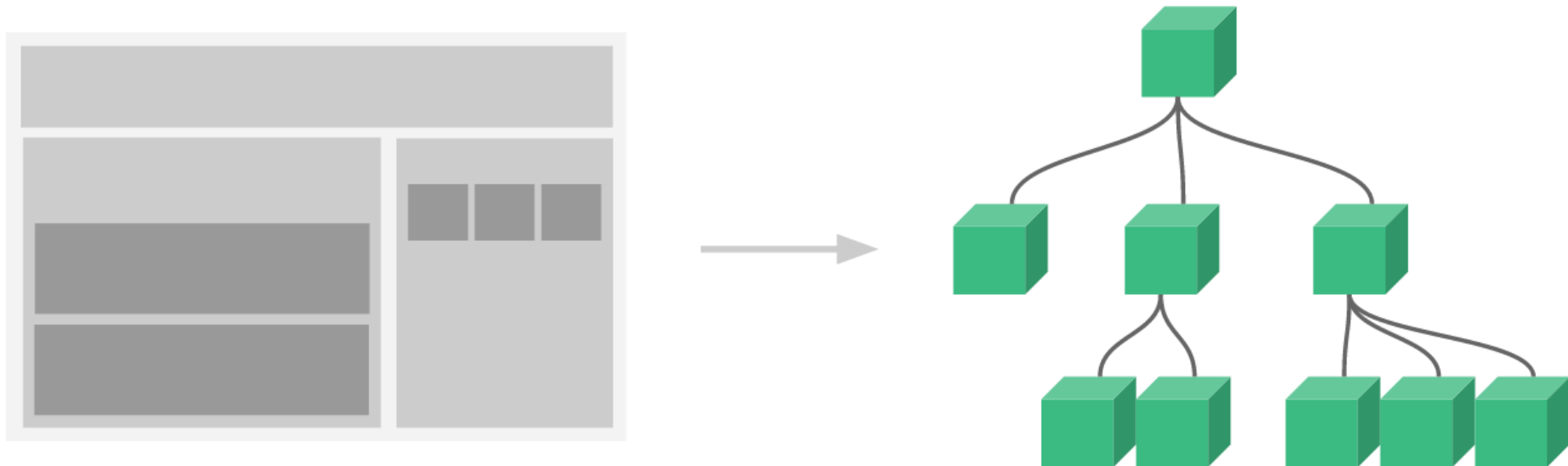
## Performant

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.

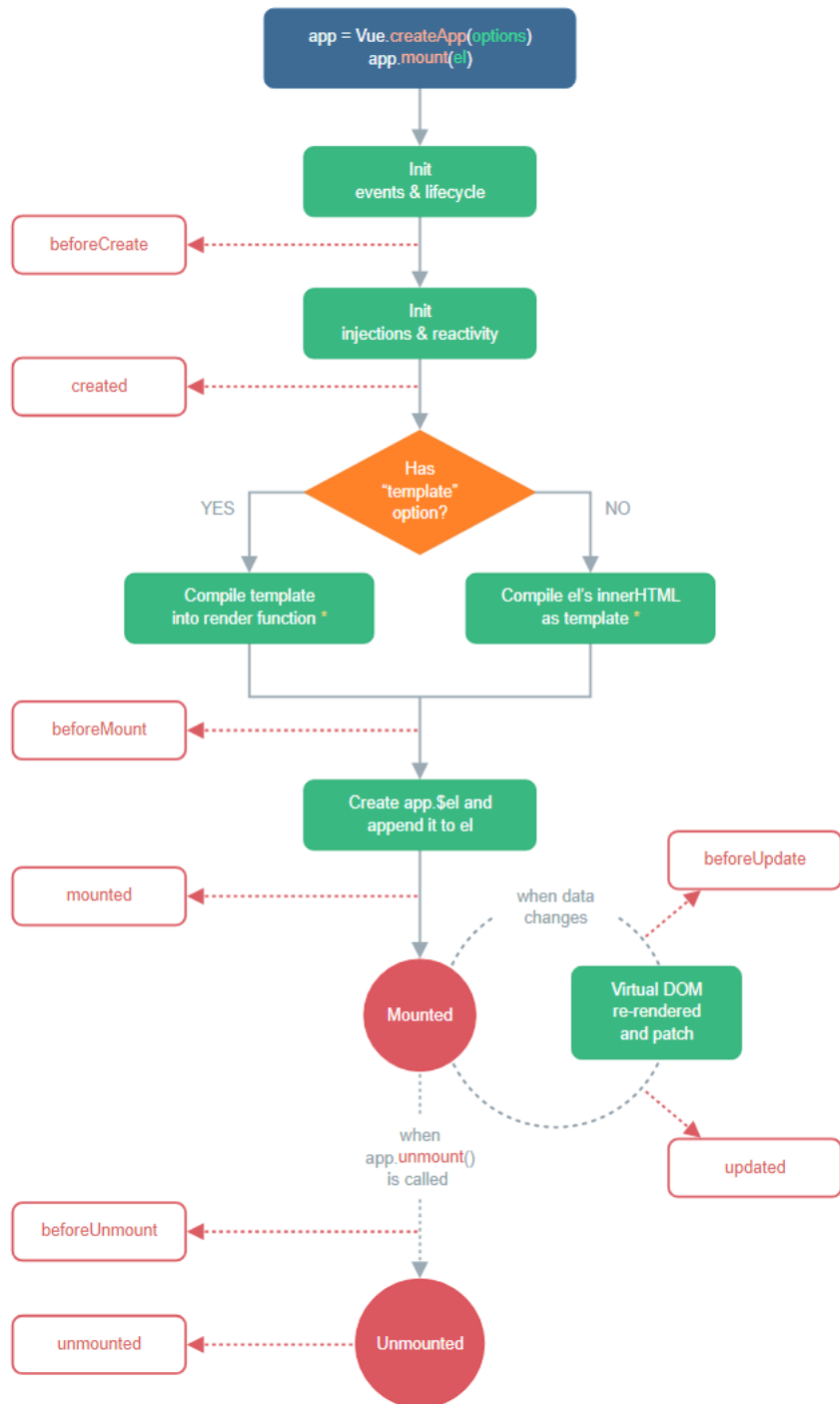
<https://v3.ru.vuejs.org/ru/guide/component-basics.html>

## **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» (событием) 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. Система сборки **Vue CLI**



# Система сборки Vue-приложений **Vue CLI**

```
npm install -g @vue/cli
```

```
vue create my-app
```

```
cd my-app
```

```
npm run serve
```

```
npm run build
```

**Vue CLI** - утилита позволяющая выполнять сборку **Vue-приложения**, содержащая локальный сервер и поддерживающая использования однофайловых компонентов.

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

# 4. Однофайловые компоненты во Vue.js

```

1  <template>
2    <h1>{{ msg }}</h1>
3
4  <p>
5    <a href="https://vitejs.dev/guide/features."
6      | Vite Documentation
7    </a>
8    |
9    <a href="https://v3.vuejs.org/" target="_bl
10  </p>
11
12  <button type="button" @click="state.count++">
13    | count is: {{ state.count }}
14  </button>
15  <p>
16    | Edit
17    | <code>components/HelloWorld.vue</code> to t
18  </p>
19  </template>
20
21  <script setup>
22    import { defineProps, reactive } from 'vue'
23
24  defineProps({
25    | msg: String
26  })
27
28  const state = reactive({ count: 0 })
29  </script>
30
31  <style scoped>
32  <a {
33    | color: #42b983;
34  }
35  </style>

```

## Однофайловый компонент


Такой компонент размещается в одном файле с расширением **.vue**. В этом файле размещены и разметка компонента и его стили и, разумеется, логика его работы (конфигурационный объект).

Конфигурационный объект экспортируется из файла (синтаксис ECMAScript-модулей). В целевом файле (например файле старшего компонента или приложения) такой однофайловый компонент необходимо импортировать.

## 5. Немного практики

# ShopApp + Vue CLI

Shop App

 (5) 53.64\$



Воспользуемся: <https://fakestoreapi.com/products/>

**На следующем занятии**

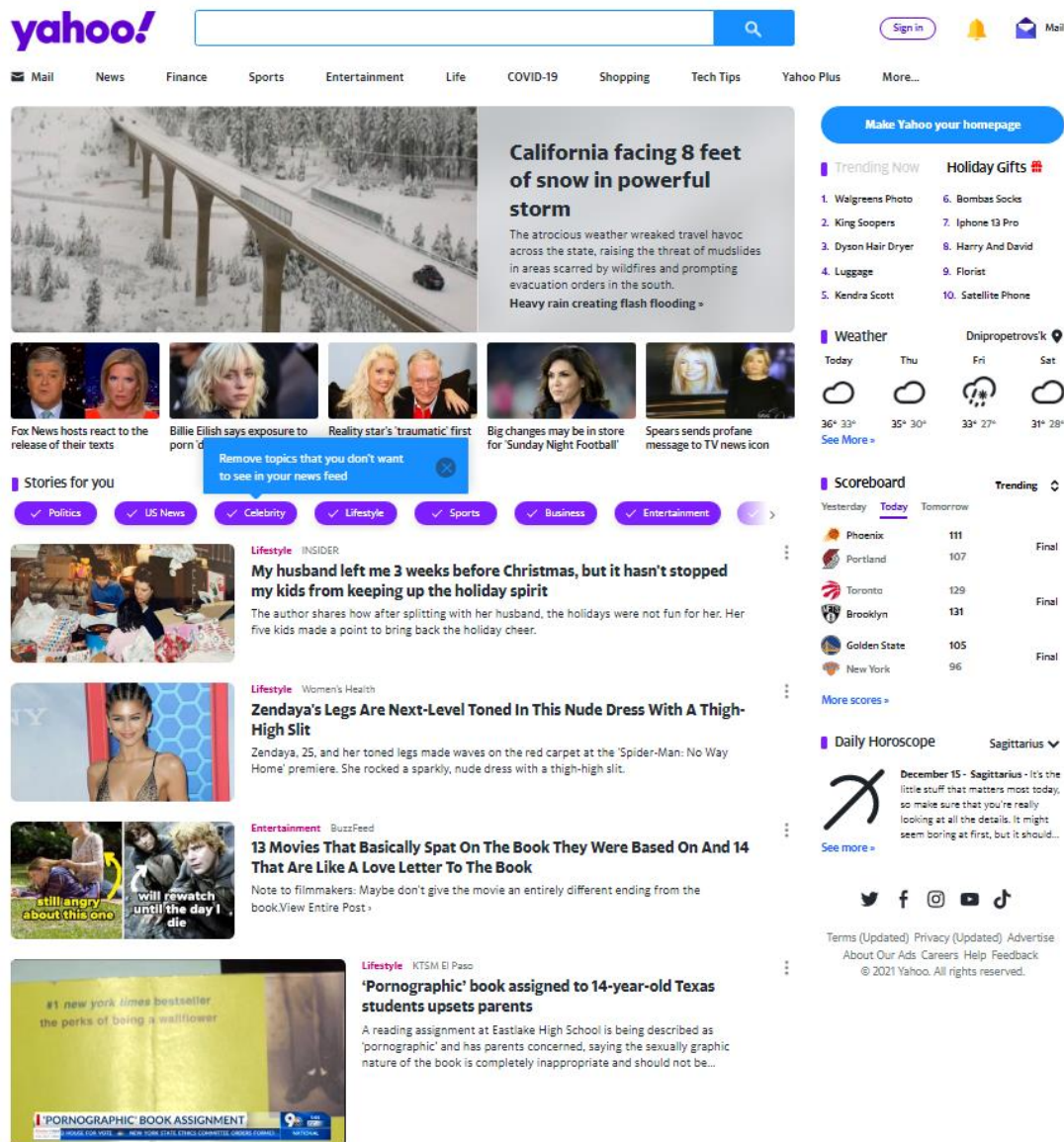
**На следующем занятии**

**Концепция `serverless` приложений**

# Финальное Домашнее задание



# Домашнее задание (в перспективу)



Создайте страницу-портал.  
На которой будут  
следующий разделы  
(блоки, виджеты):

- 1) Новостная лента;
- 2) Курсы основных валют (Приват/НБУ);
- 3) Прогноз погоды для той местности где находится пользователь;
- 4) Цену акций компаний Microsoft, Google, Apple, Facebook (Meta), Amazon, Netflix.

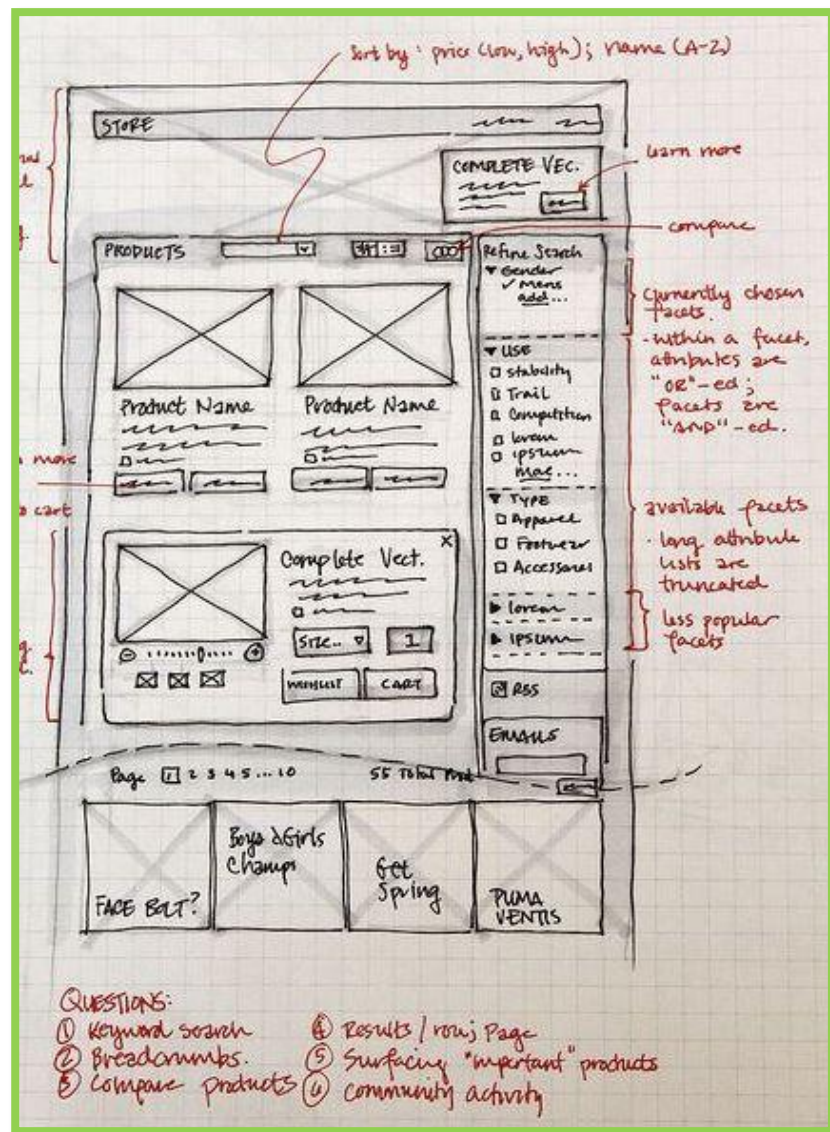
# Домашнее задание

## API сервисы которые могут пригодится

1. <https://mediastack.com/>
2. <https://weatherstack.com/>
3. <https://openweathermap.org/>
4. <https://api.privatbank.ua/>
5. <https://bank.gov.ua/ua/open-data/api-dev>
6. <https://exchangeratesapi.io/>
7. <https://marketstack.com/>

*Большая часть сервисов требует **регистрации**, и использования предоставляемого ключа в URL'е запроса. Также у многих есть **лимиты** по количеству запросов за период.*

# Домашнее задание



Не забываем

1.Планирование

2.Проектирование

3.Поэтапная разработка