

node.js

npm -- пакетный менеджер, типа как mvn

preset (базовая настройка)

запустить -- npm run serve

наше приложение будет подвешено сюда:

```
1 <!DOCTYPE html>
2 <html lang="">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width,initial-scale=1.0">
7   <link rel="icon" href="%= BASE_URL %>favicon.ico">
8   <title>%= htmlWebpackPlugin.options.title %</title>
9 </head>
10 <body>
11   <noscript>
12     <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without JavaScript .
13   </noscript>
14   <div id="app"></div>
15 </body>
16 </html>
```

```
import Vue from 'vue'
import App from './App.vue'

Vue.config.productionTip = false

new Vue({
  render: h => h(App),
}).$mount( elementOrSelector: '#app')
```

каждый vue файл это 1) кусок пользовательского интерфейса 2) логики

компоненты образуют дерево

которое незаметно для нас vue.js преобразует в DOM и подвесит к #app (см index.html)

самый корневой компонент для нас App.vue

Из чего состоит компонент:

- часть template -- на html-подобном разметка компонента
- часть script
- часть style -- можно писать css

style scoped -- применить стиль исключительно для этого компонента

```
<template>
  <div id="app">
    <Header/>
  </div>
</template>

<script>
import Header from "../components/Header";
export default {
  name: 'App',
  components: {Header}
}
</script>
```

КОРОЧЕ ключевая фишка это КОМПОНЕНТЫ

мы разбиваем фронтенд на компоненты это все облегчает

```
<script>
export default {
  name: "SidebarPost",
  props: ["post"]
}
<div class="header">
  {{ post.title }}
</div>

<Middle :posts="posts"/>
<Footer/>

<SidebarPost v-for="post in posts" :post="post" :key="post.id"/>
```

ВЫЧИСЛИМЫЕ СВОЙСТВА

```
computed: {
  viewPosts: function() {
    return Object.values(this.posts).sort(compareFn: (a, b) => b.id - a.id).slice(0, 2);
  }
},
```

```
<SidebarPost v-for="post in viewPosts" :post="post" :key="post.id"/>
```

чем отличаются данные от свойств? данные менять можно

```
<Index v-if="page === 'Index'"/>
<Enter v-if="page === 'Enter'"/>

<a href="#" @click.prevent="changePage('Index')">Register</a>
```

использование шины событий

```
changePage: function (page) {
  this.$root.$emit(event: "onChangePage", page);
}
```