

Вводная лекция

Что такое веб приложение,
его структура, JS-фреймворки

Что такое Web?

Веб

Материал из Википедии — свободной энциклопедии

Веб или **вэб** (англ. *web* — паутина) — интернет-пространство:

- **Веб** — система доступа к связанным между собой документам на различных компьютерах, подключённых к Интернету.
 - Веб 1.0
 - Веб 2.0
- Веб- — часть составных слов, относящихся к «всемирной паутине». [Список статей, названия которых начинаются с «Веб-».](#)
- **Доктор Веб** — семейство антивирусов.

Что такое Web?



Что такое Web?

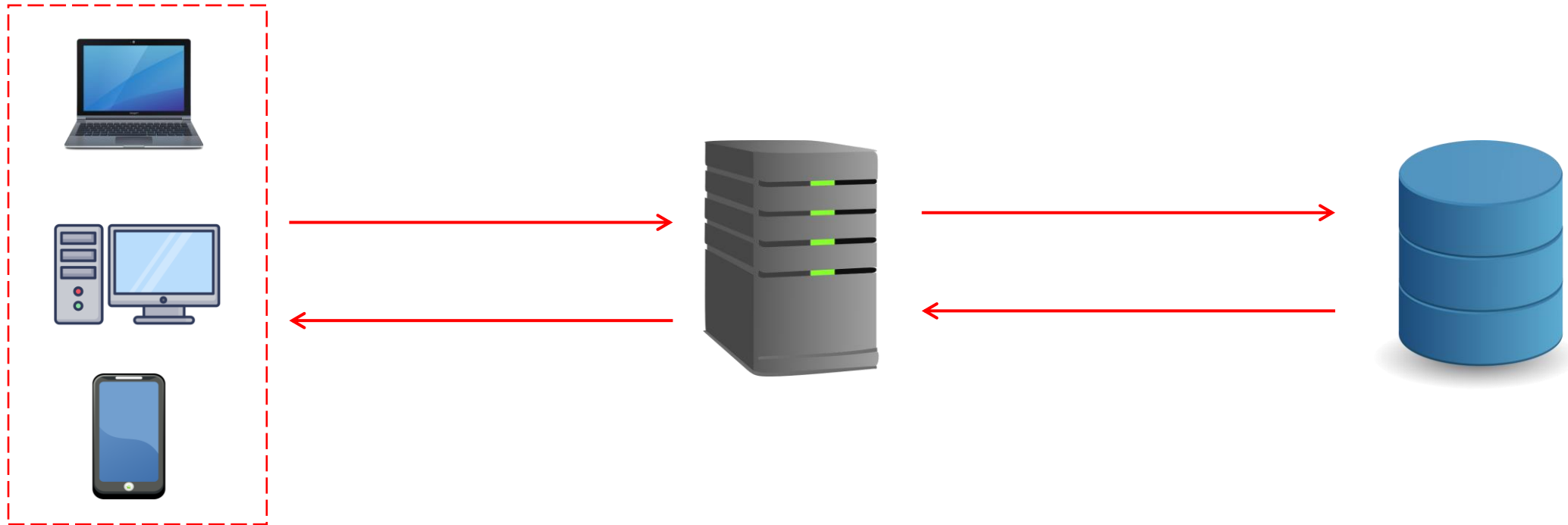


Из чего состоит web приложение

Клиент

Сервер

База данных



Из чего состоит web приложение

Клиент

Сервер

База данных



Клиентский код
(Frontend)



Серверный код
(Backend)

Из чего состоит web приложение



Клиентский код
(Frontend)



Серверный код
(Backend)



Из чего состоит web приложение



Frontend
разработчик

Клиентский код
(Frontend)

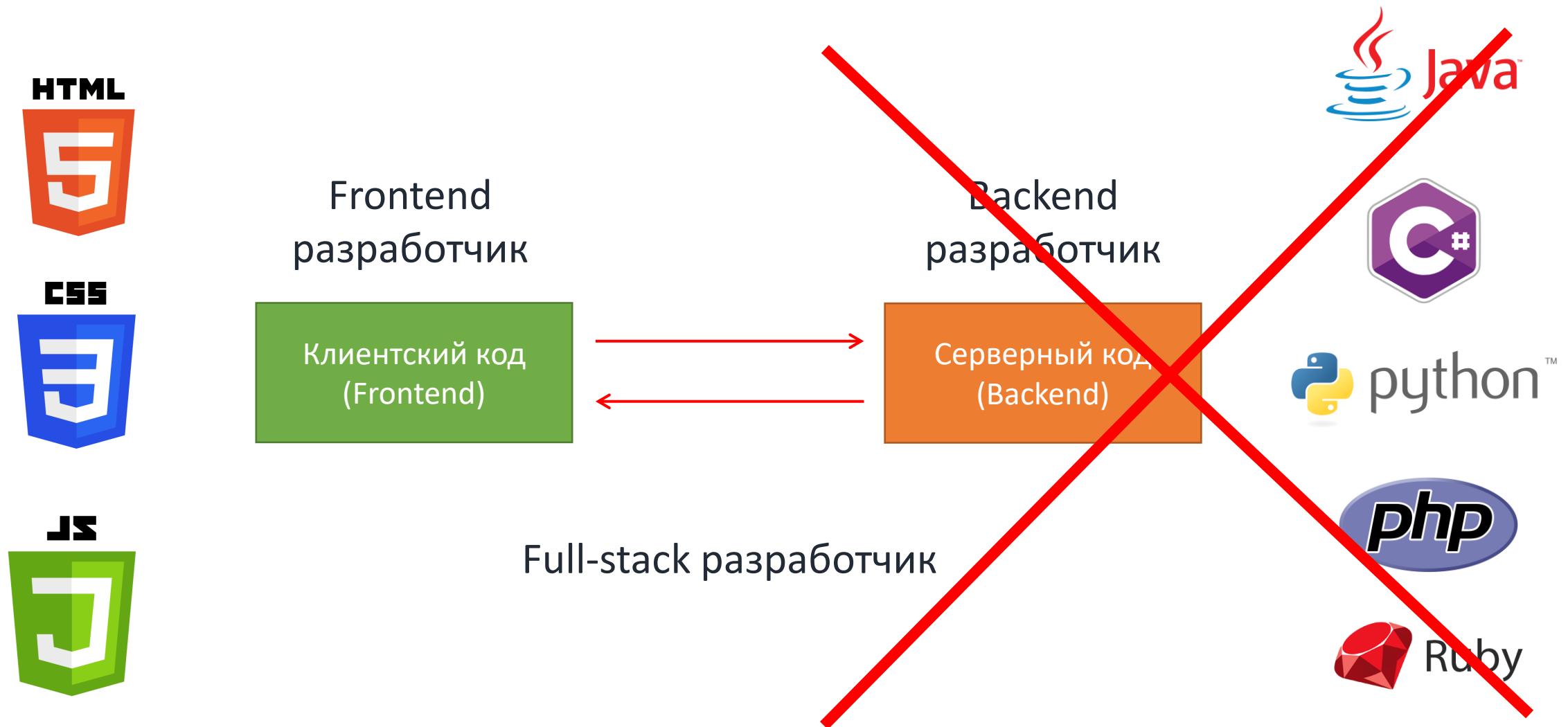
Backend
разработчик

Серверный код
(Backend)

Full-stack разработчик



Из чего состоит web приложение



Что такое Frontend

HTML



CSS



JS



Что такое Frontend



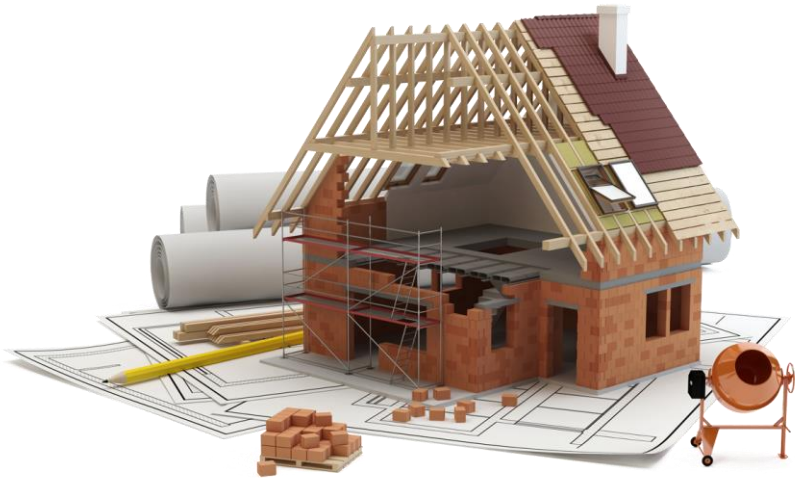
Что такое Frontend



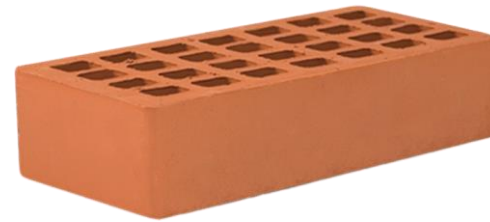
HTML

Hyper**T**ext **M**arkup **L**anguage

Язык гипертекстовой разметки



Web-страница



HTML тег (элемент)

HTML — это текстовый файл с расширением *.html.

HTML

HTML – не является языком программирования

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <title>Пример веб-страницы</title>
6    </head>
7    <body>
8      <h1>Заголовок</h1>
9      <!-- Комментарий -->
10     <p>Первый абзац.</p>
11     <p>Второй абзац.</p>
12   </body>
13 </html>
```

- тег **p** – параграф
`<p> ... </p>`
- тег **img** – изображение
``
- тег **div** – блочный элемент
`<div> ... </div>`
- тег **input** – текстовое поле
`<input type=...>`
- тег **button** – кнопка
`<button> ... </button>`
- тег **link** – связывает документ с другим документом
`<link rel=... href=...>`
- тег **script** – описание JavaScript кода
`<script type=...> ... </script>`
- тег **title** – заголовок документа
`<title>...</title>`
- тег **table** – таблица
`<table> ... </table>`

HTML

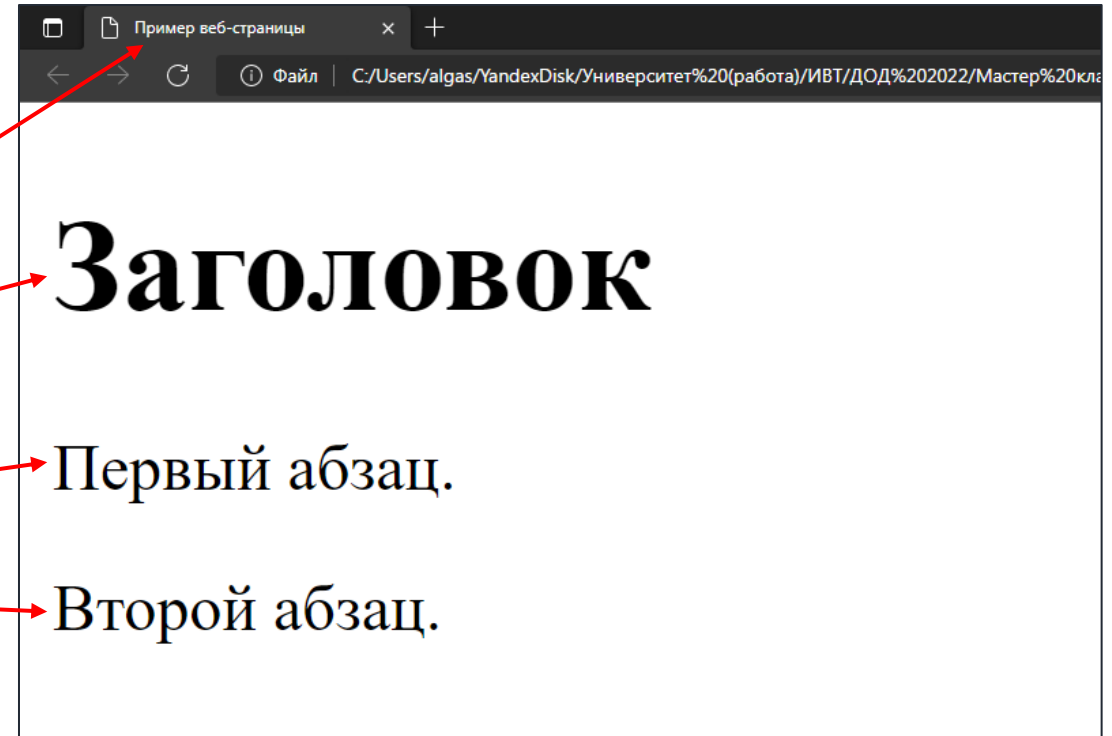
```
<div class="timer-input-wrapper">
  <input
    id="seconds"
    type="number"
    min="0"
    max="60"
    maxlength="2"
    value="0"
    class="timer-input"
    onkeypress="return false"
    onpaste="return false;"
  />
</div>
```

- атрибут **class** – задает стилевой CSS класс
- атрибут **id** – задает идентификатор (уникальное имя элемента) по которому к нему можно обращаться
- атрибут **type** – указывает к какому типу относится элемент формы
- атрибут **min** – устанавливает нижнее значение для ввода чисел
- атрибут **max** – устанавливает верхнее значение для ввода чисел
- атрибут **maxLength** – устанавливает максимальное число символов, которое можно ввести
- атрибут **value** – устанавливает значение
- событие **onkeypress** – срабатывает, когда клавиша на клавиатуре нажата и отпущена
- событие **onpaste** – срабатывает, когда пользователи пытаются вставить скопированное значение

Структура HTML документа

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <title>Пример веб-страницы</title>
6    </head>
7    <body>
8      <h1>Заголовок</h1>
9      <!-- Комментарий -->
10     <p>Первый абзац.</p>
11     <p>Второй абзац.</p>
12   </body>
13 </html>
```

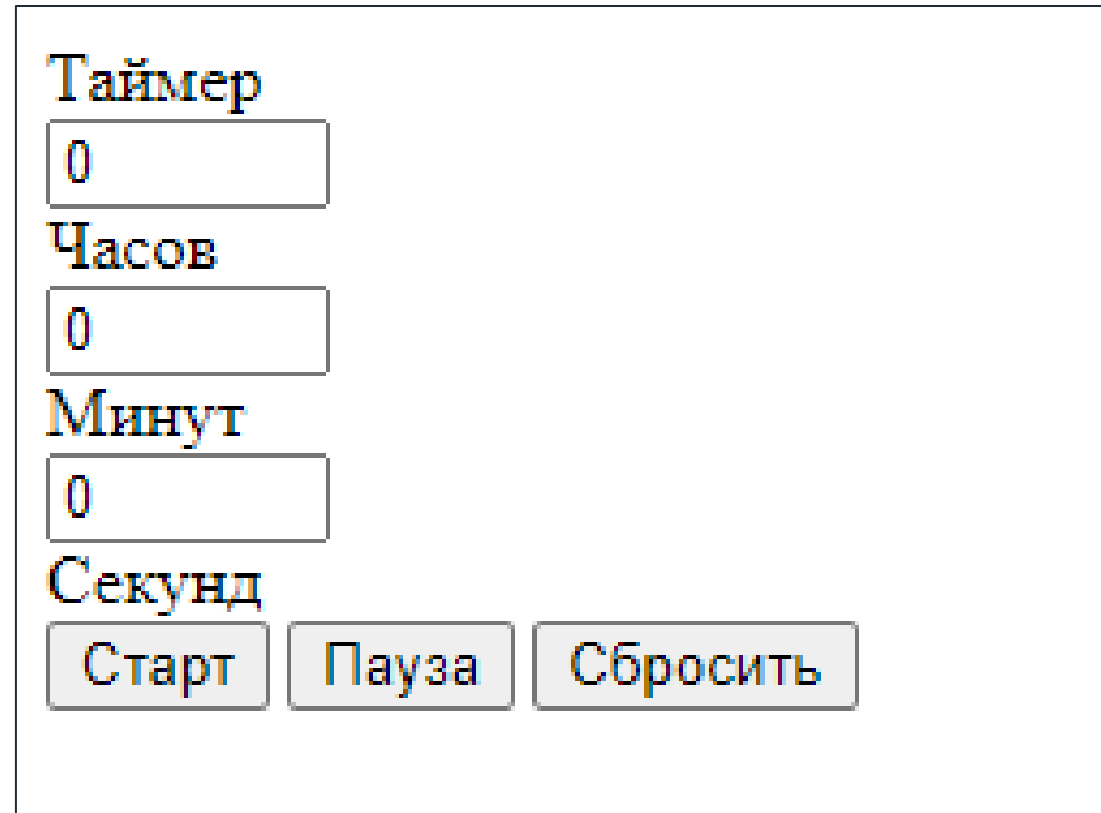
Исходный код страницы



Результат выполнения кода

HTML

```
index.html x style.css scripts.js
index.html > ...
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <title>Таймер</title>
6 <link href="style.css" rel="stylesheet" />
7 <link
8 href="https://fonts.googleapis.com/css?family=Inter"
9 rel="stylesheet"
10 />
11 <script src="scripts.js" defer></script>
12 </head>
13 <body>
14 <div class="timer-wrapper">
15 <div class="timer-block">
16 <div class="timer-title">Таймер</div>
17 <div class="timer-inputs-wrapper">
18 <div class="timer-input-block">
19 <div class="timer-input-wrapper">...
20 </div>
21 <div class="timer-input-label">Часов</div>
22 </div>
23 <div class="timer-input-block">...
24 </div>
25 <div class="timer-input-block">
26 <div class="timer-input-wrapper">
27 <input
28 id="seconds"
29 type="number"
30 min="0"
31 max="60"
32 maxLength="2"
33 value="0"
34 class="timer-input"
35 onkeypress="return false"
36 onpaste="return false;"
37 />
38 </div>
39 <div class="timer-input-label">Секунд</div>
40 </div>
41 <div>
42 <div class="timer-btns-wrapper">
43 <button id="startBtn" class="timer-btn timer-btn-start">Старт</button>
44 <button id="pauseBtn" class="timer-btn timer-btn-pause">Пауза</button>
45 <button id="resetBtn" class="timer-btn timer-btn-reset">
46 Сбросить
47 </button>
48 </div>
49 </div>
50 </div>
51 </body>
52 </html>
```



Результат выполнения кода

CSS

Cascading Style Sheets

Каскадные таблицы стилей



HTML

+



CSS

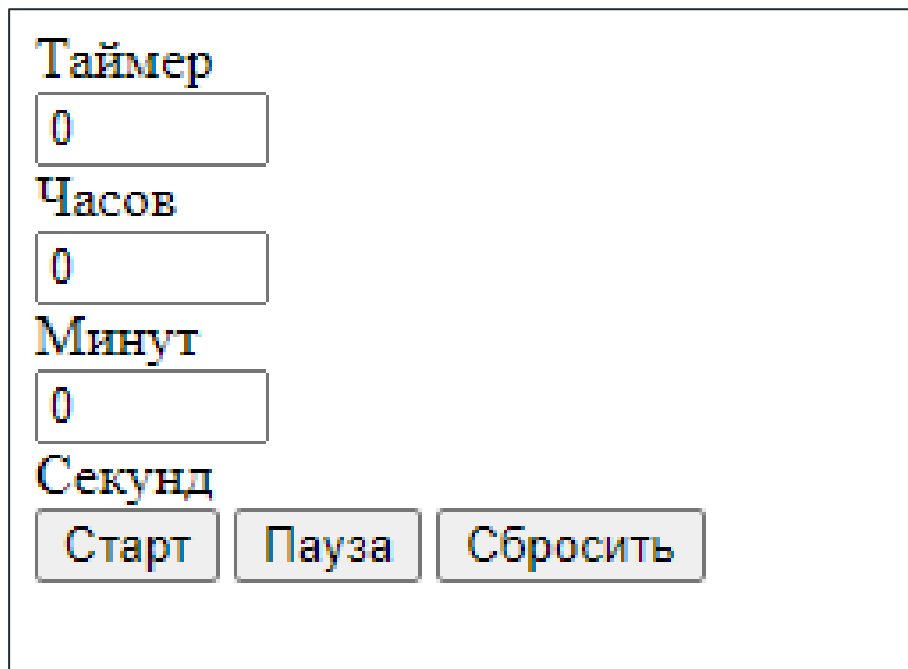
=



Web-страница

CSS

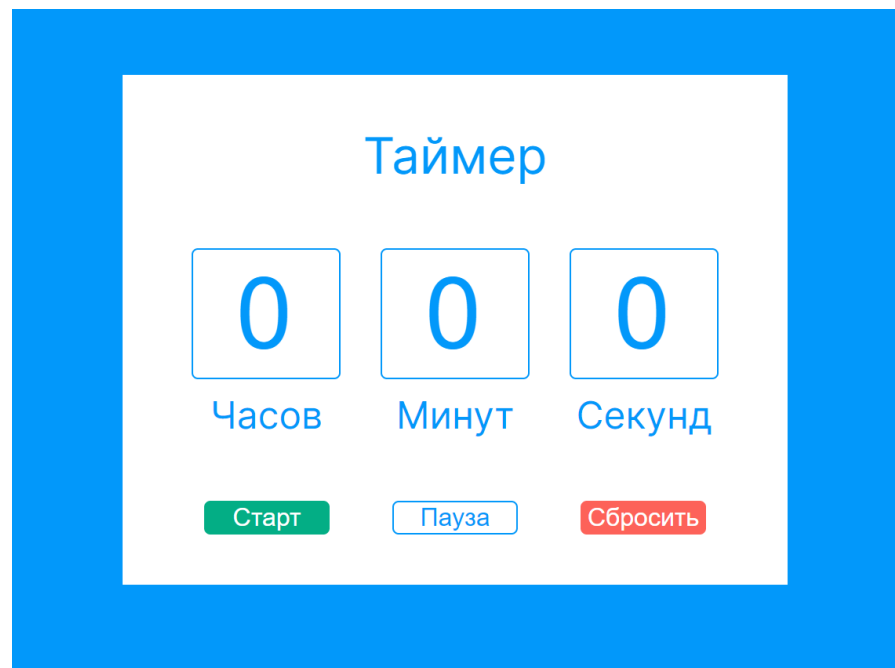
CSS – не является языком программирования



The image shows a web page for a timer without any CSS styling. The text is in a default monospaced font. The layout is as follows:

- Title: Таймер
- Input field: 0
- Label: Часов
- Input field: 0
- Label: Минут
- Input field: 0
- Label: Секунд
- Buttons: Старт, Пауза, Сбросить

Страница без CSS



The image shows the same timer web page but with CSS styling. The page has a blue border. The text is in a blue serif font. The layout is as follows:

- Title: Таймер
- Three large input fields, each containing the number 0, representing hours, minutes, and seconds.
- Labels: Часов, Минут, Секунд
- Buttons: Старт (green), Пауза (blue), Сбросить (red)

Страница с CSS

CSS – это текстовый файл с расширением *.css

Синтаксис CSS

селектор { **свойство**: **значение**; }

Селектор - конструкция, которая позволяет выбрать отдельные или однотипные элементы (тег) на странице, чтобы их стилизовать.

Виды селектор:

- **по идентификатору (id):** `<div id="title">...</div>`. id – должен быть уникальным. На странице не должно быть тегов с двумя одинаковыми id.
- **по классу (class):** `<div class="title">...</div>`. Классы могут повторяться. Несколько тегов могут иметь одинаковые классы.
- **по тегу:** `<div>...</div>`. Стили будут применяться ко всем тегам на странице.

Синтаксис CSS

селектор { свойство: значение; }

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <title>Пример веб-страницы</title>
6    </head>
7    <body>
8      <div id="title">Title</div>
9      <div class="title">Title</div>
10     <div>Title</div>
11   </body>
12 </html>
```

index.html

```
1  ∨ #title {
2    |     background:  red;
3    |
4    | }
5  ∨ .title {
6    |     background:  red;
7    |
8    | }
9  ∨ div {
10   |     background:  red;
11   |
12   | }
```

style.css

Синтаксис CSS

селектор { свойство: значение; }

```
43 .timer-input, .timer-input:focus {
44     width: 10rem;
45     padding: 0.5rem;
46     padding-left: 18px;
47     border: 2px solid #0298FA;
48     border-radius: 0.5rem;
49     font-size: 8rem;
50     color: #0298FA;
51     text-align: center;
52     outline: none;
53     caret-color: transparent;
54 }
```

- **width** – ширина элемента
- **height** – высота элемента
- **margin** – внешний отступ
- **padding** – внутренний отступ
- **font-family** – шрифт
- **font-size** – размер шрифта
- **display** – как расположить элемент
- **background** – задний фон
- **text-align** – как выравнять текст
- **border** – контур
- **border-radius** – радиус контура

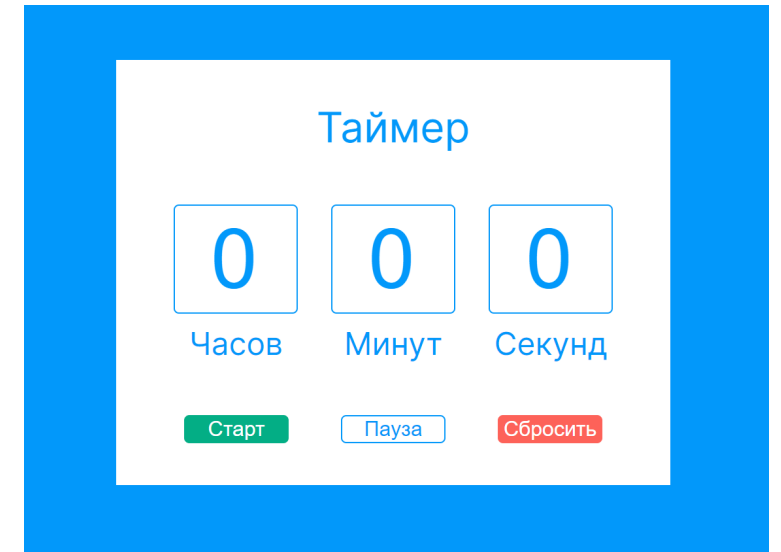
Синтаксис CSS

```
index.html > ...
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Таймер</title>
6     <link href="style.css" rel="stylesheet" />
7     <link
8       href="https://fonts.googleapis.com/css?family=Inter"
9       rel="stylesheet"
10    />
11     <script src="scripts.js" defer></script>
12   </head>
13   <body>
14     <div class="timer-wrapper">
15       <div class="timer-block">
16         <div class="timer-title">Таймер</div>
17         <div class="timer-inputs-wrapper">
18           <div class="timer-input-block">
19             <div class="timer-input-wrapper">...
31           </div>
32           <div class="timer-input-label">Часов</div>
33         </div>
34       </div>
35       <div class="timer-input-block">...
50     </div>
51     <div class="timer-input-block">
52       <div class="timer-input-wrapper">
53         <input
54           id="seconds"
55           type="number"
56           min="0"
57           max="60"
58           maxLength="2"
59           value="0"
60           class="timer-input"
61           onkeypress="return false"
62           onpaste="return false;"
63         />
64       </div>
65       <div class="timer-input-label">Секунд</div>
66     </div>
67     <div class="timer-input-block">
68       <div class="timer-input-wrapper">
69         <div class="timer-input-label">Секунд</div>
70       </div>
71       <div class="timer-input-label">Секунд</div>
72     </div>
73     <div class="timer-input-block">
74       <div class="timer-input-wrapper">
75         <div class="timer-input-label">Секунд</div>
76       </div>
77       <div class="timer-input-label">Секунд</div>
78     </div>
79   </body>
80 </html>
```

index.html

```
style.css > .timer-btn
1 html, body {
2   width: 100%;
3   height: 100%;
4   margin: 0;
5   font-family: 'Inter';
6 }
7
8 .timer-wrapper {
9   display: flex;
10  width: 100%;
11  height: 100%;
12  background: #0298FA;
13 }
14
15 .timer-block {
16   margin: auto;
17   background: #FFFFFF;
18   padding: 4rem;
19 }
20
21 .timer-title {
22   font-size: 4rem;
23   color: #0298FA;
24   text-align: center;
25 }
26
27 .timer-inputs-wrapper {
28   display: flex;
29   justify-content: space-between;
30   margin-top: 5rem;
31   margin-bottom: 5rem;
32 }
33
34 .timer-input-block {
35   margin-left: 1.5rem;
36   margin-right: 1.5rem;
37 }
38
39 .timer-input-wrapper {
40   margin-bottom: 1rem;
41 }
```

style.css



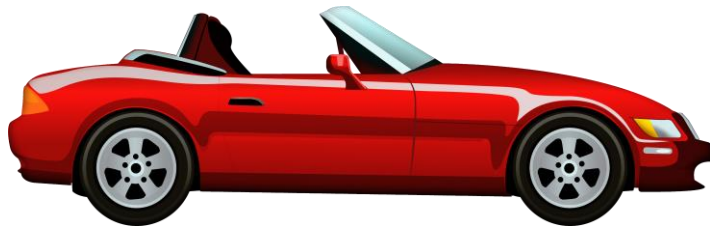
Результат выполнения кода

JavaScript

JavaScript – язык программирования.

JavaScript – это не Java.

JavaScript был создан, чтобы «сделать веб-страницы живыми».



HTML + CSS

+



JavaScript

=



Web-страница

JavaScript – это текстовый файл с расширением *.js

JavaScript

JavaScript можно учить вечно.

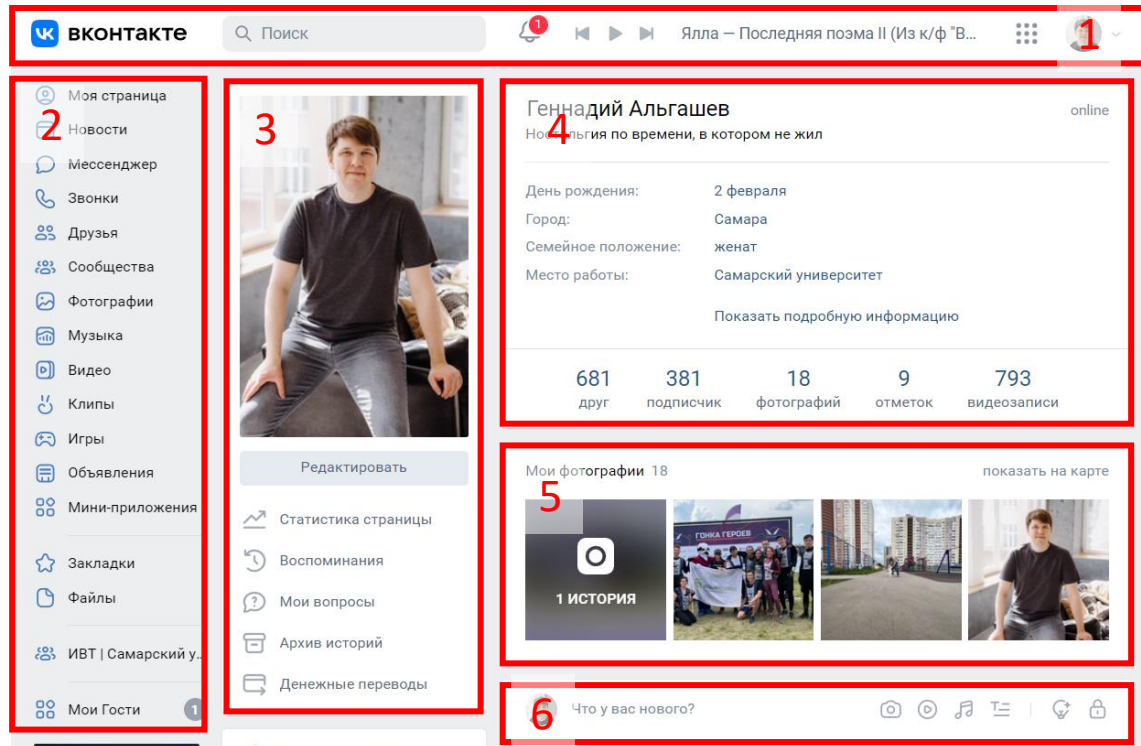


JavaScript

```
scripts.js > resetBtn.addEventListener("click") callback
1  let hours = 0;
2  let minutes = 0;
3  let seconds = 0;
4  let allSeconds = 0;
5  let interval;
6
7  const hoursEl = document.getElementById("hours");
8  const minutesEl = document.getElementById("minutes");
9  const secondsEl = document.getElementById("seconds");
10
11  const startBtn = document.getElementById("startBtn");
12  const pauseBtn = document.getElementById("pauseBtn");
13  const resetBtn = document.getElementById("resetBtn");
14
15  secondsEl.addEventListener("change", (event) => {
16    console.log("secondsEl change", event.target.value);
17    seconds = parseInt(event.target.value);
18  });
19
20  minutesEl.addEventListener("change", (event) => {
21    console.log("minutesEl change", event.target.value);
22    minutes = parseInt(event.target.value);
23  });
24
25  hoursEl.addEventListener("change", (event) => {
26    console.log("hoursEl change", event.target.value);
27    hours = parseInt(event.target.value);
28  });
29
30  startBtn.addEventListener("click", () => {
31    allSeconds = seconds + minutes * 60 + hours * 60 * 60;
32    clearInterval(interval);
33    interval = setInterval(() => {
34      seconds = seconds - 1;
35      if (seconds < 0 && minutes >= 0 && hours >= 0) {
36        seconds = 59;
37        minutes = minutes - 1;
38        if (minutes < 0 && hours >= 0) {
39          minutes = 59;
40        }
41        minutesEl.value = minutes;
42      } else if (seconds < 0 && minutes < 0 && hours === 0) {
43        clearInterval(interval);
44        seconds = 0;
45      }
46
47      secondsEl.value = seconds;
48    }, 1000);
49    console.log(allSeconds);
50  });
51
52  pauseBtn.addEventListener("click", () => {
53    clearInterval(interval);
54  });
55
```

Что такое современный Frontend

Компонентный подход



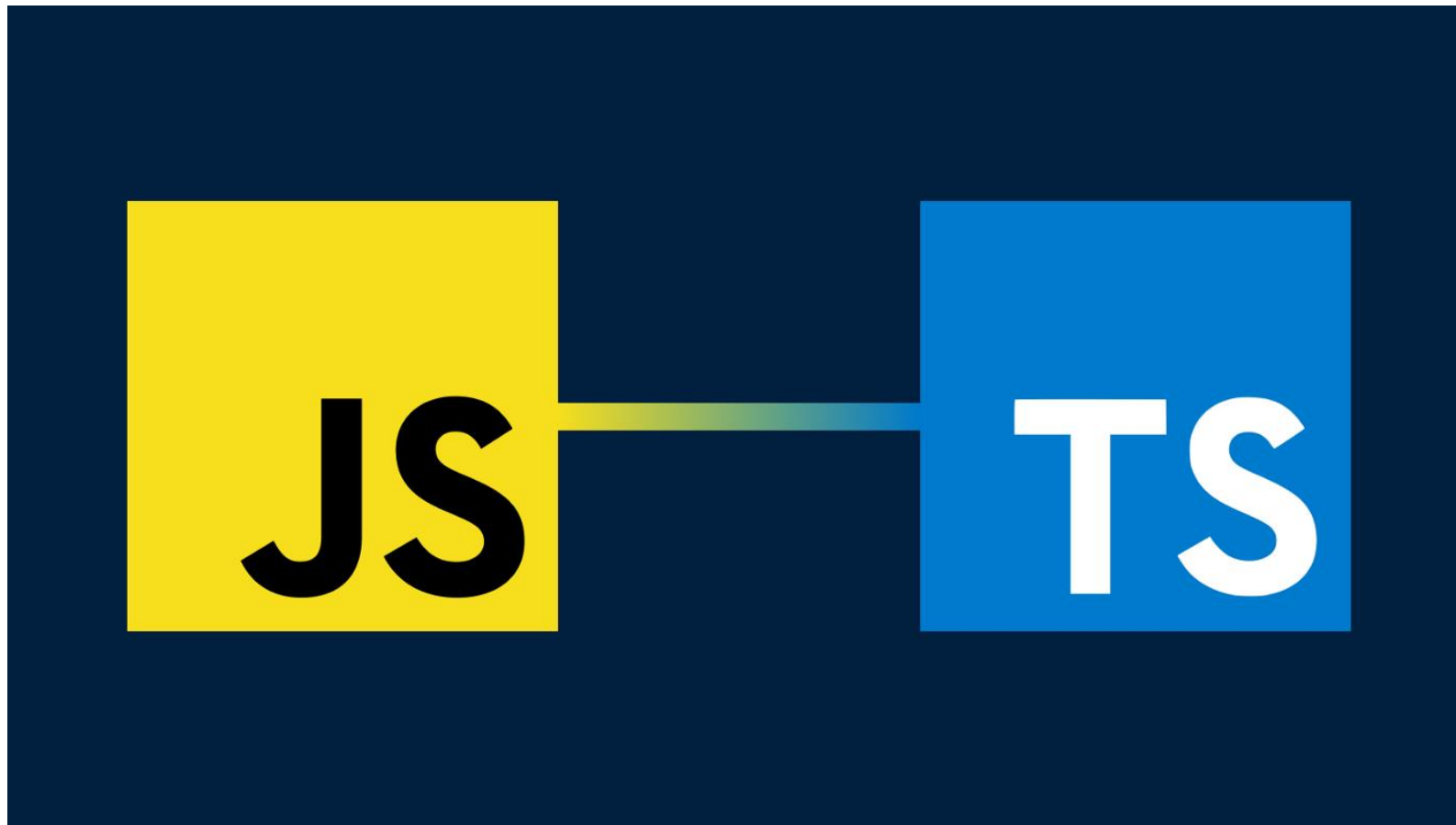
1. Хедер. Шапка .Верхняя панель навигации.
2. Сайдбар. Боковое меню.
3. Карточка пользователя
4. Информация о пользователе
5. Блок фотографий
6. Блок создания новой записи.

Что такое современный Frontend

CSS библиотеки



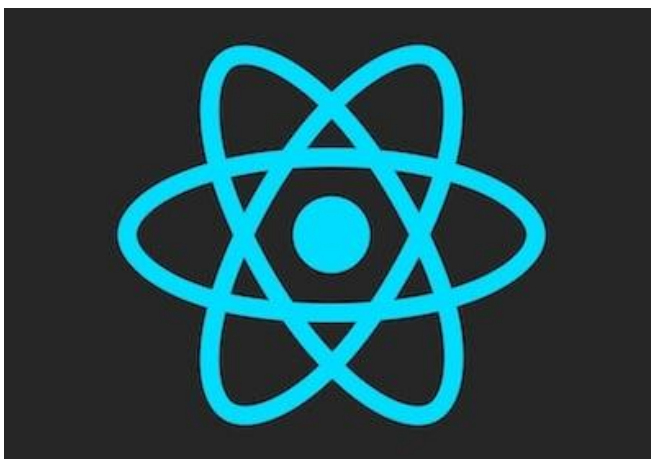
Что такое современный Frontend



Что такое современный Frontend



Angular

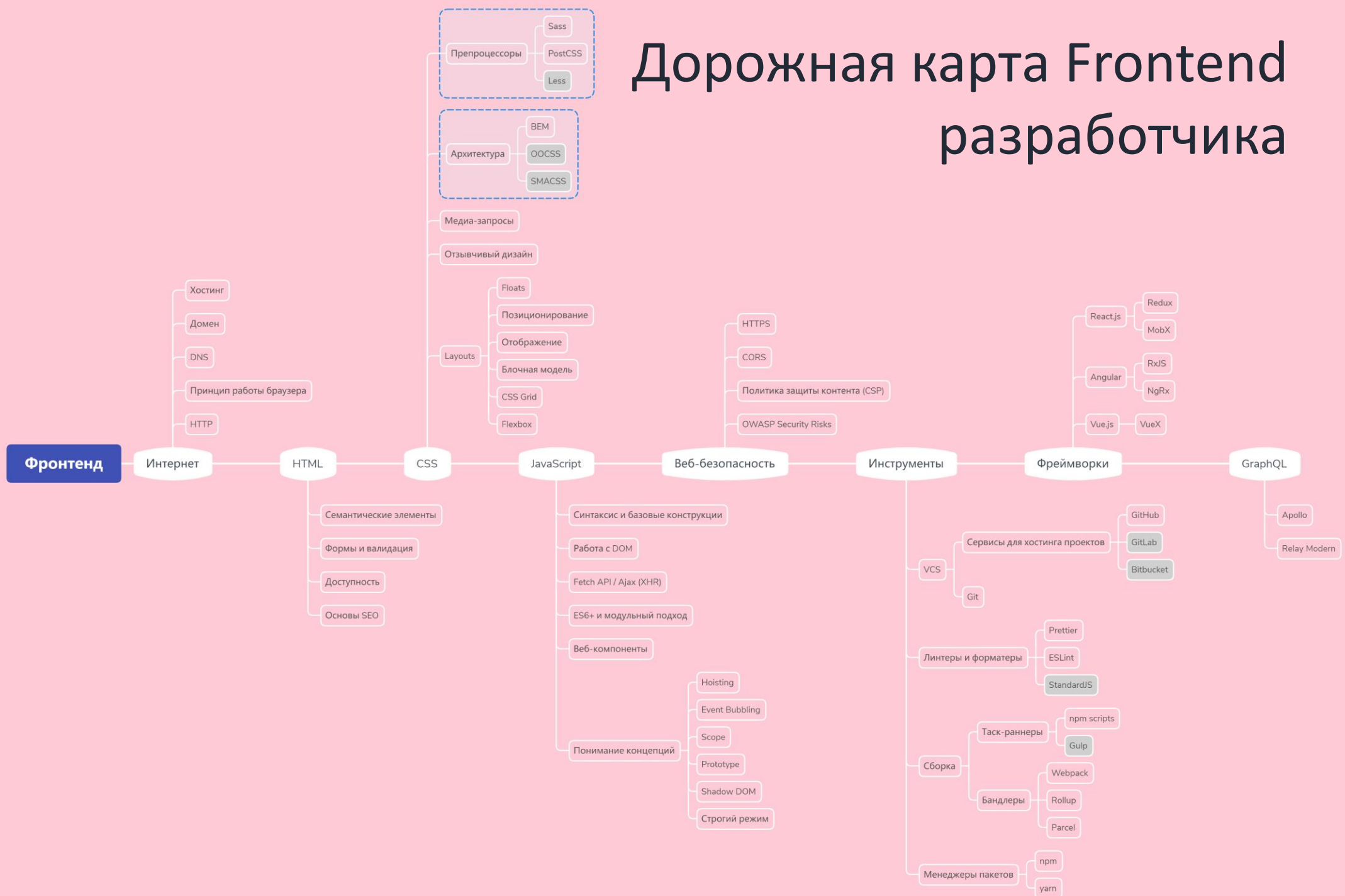


React

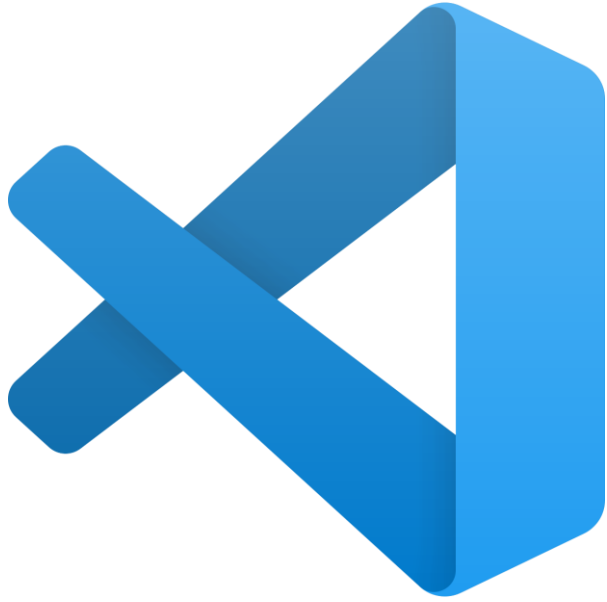


Vue

Дорожная карта Frontend разработчика



Что понадобится для работы

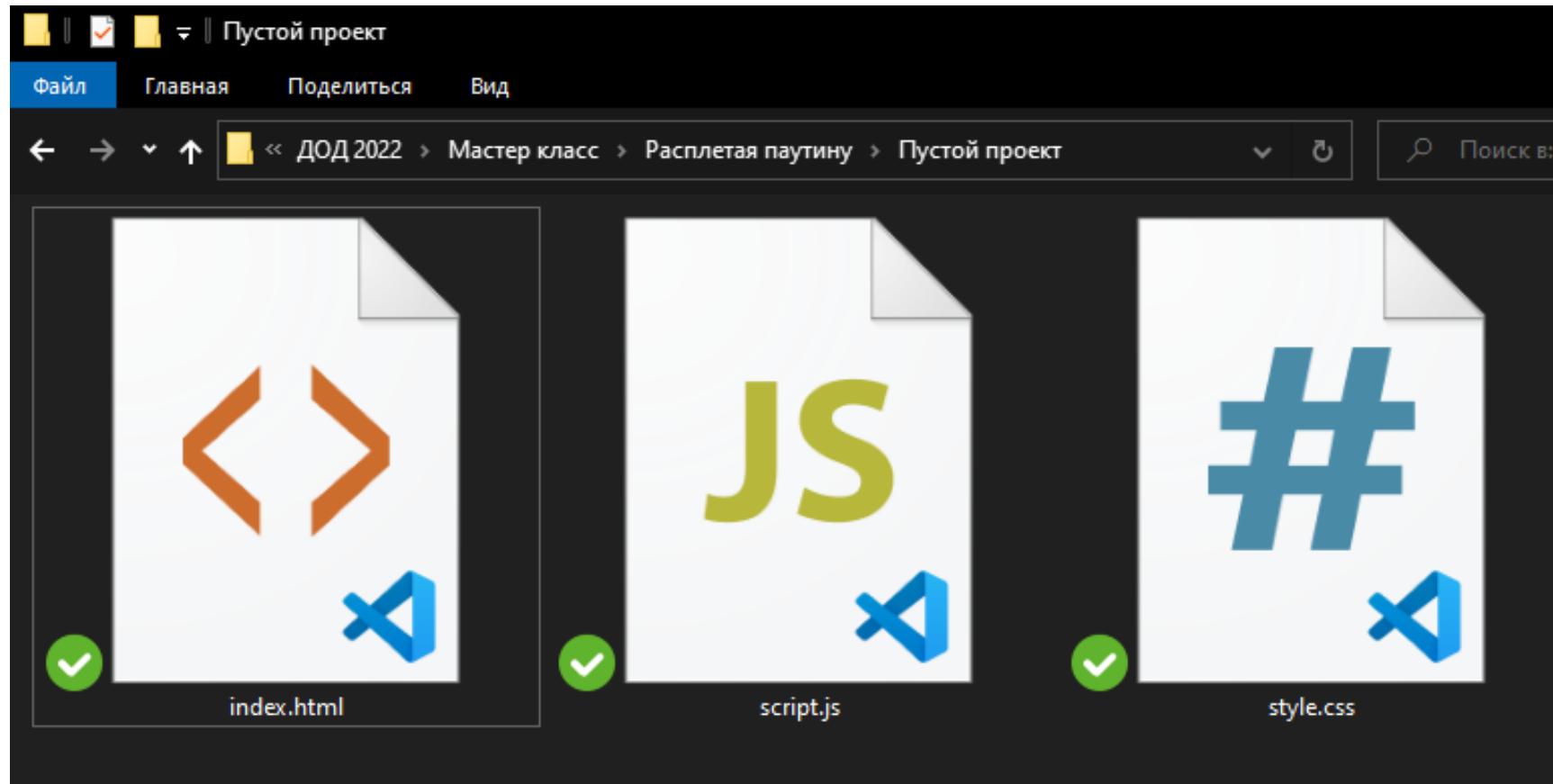


Visual Studio Code



Google Chrome

Структура проекта



Спасибо за внимание