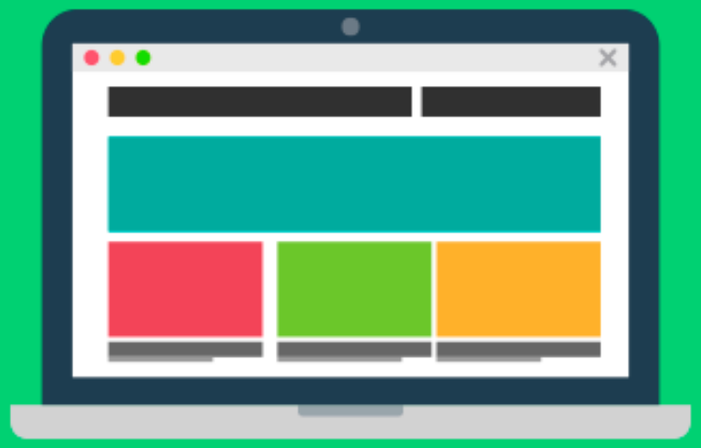
ReactJS

https://t.me/reactjsHSE2020autumn

Преподаватель: Статинов Валерий



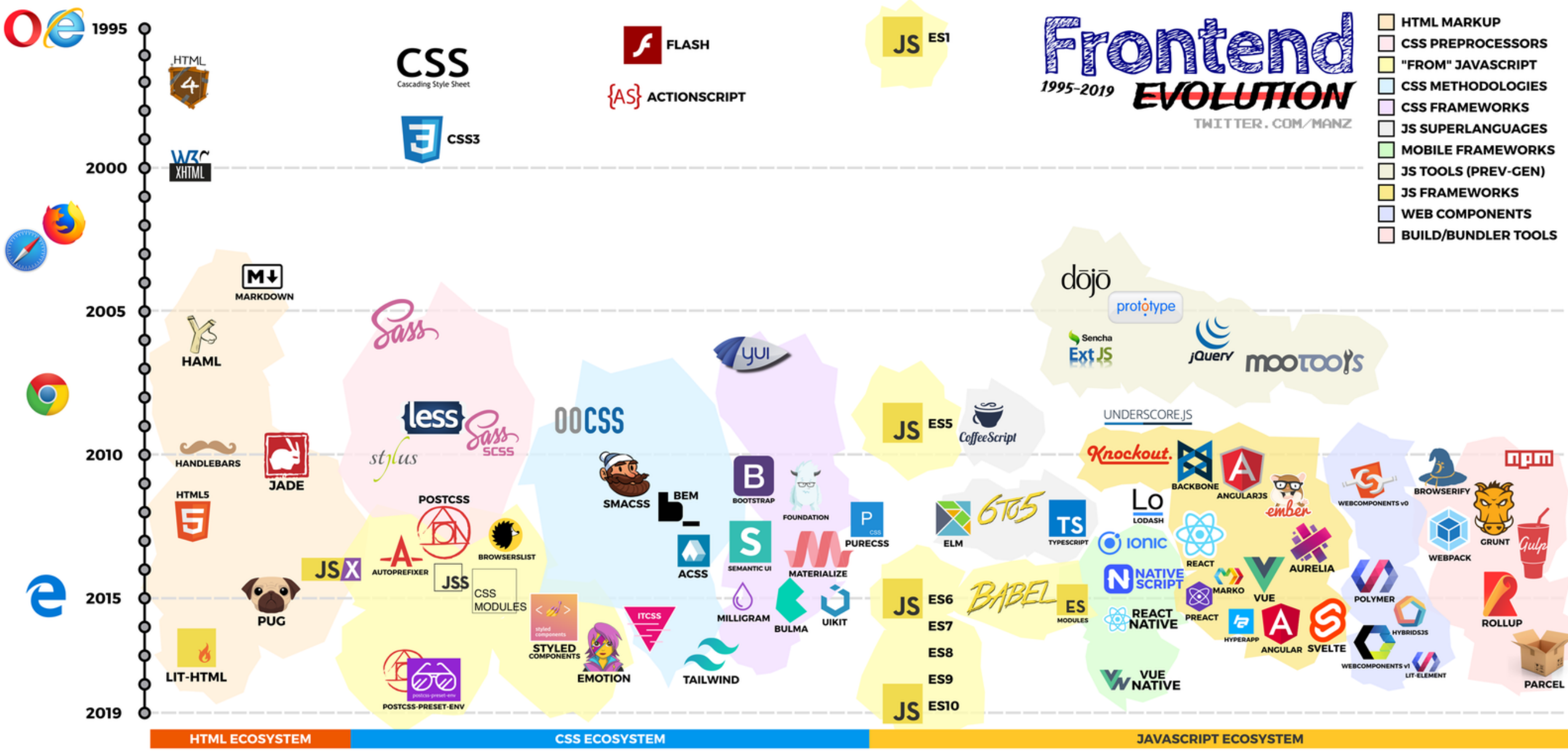
FRONTEND

BACKEND

Frontend

Все, что происходит в браузере

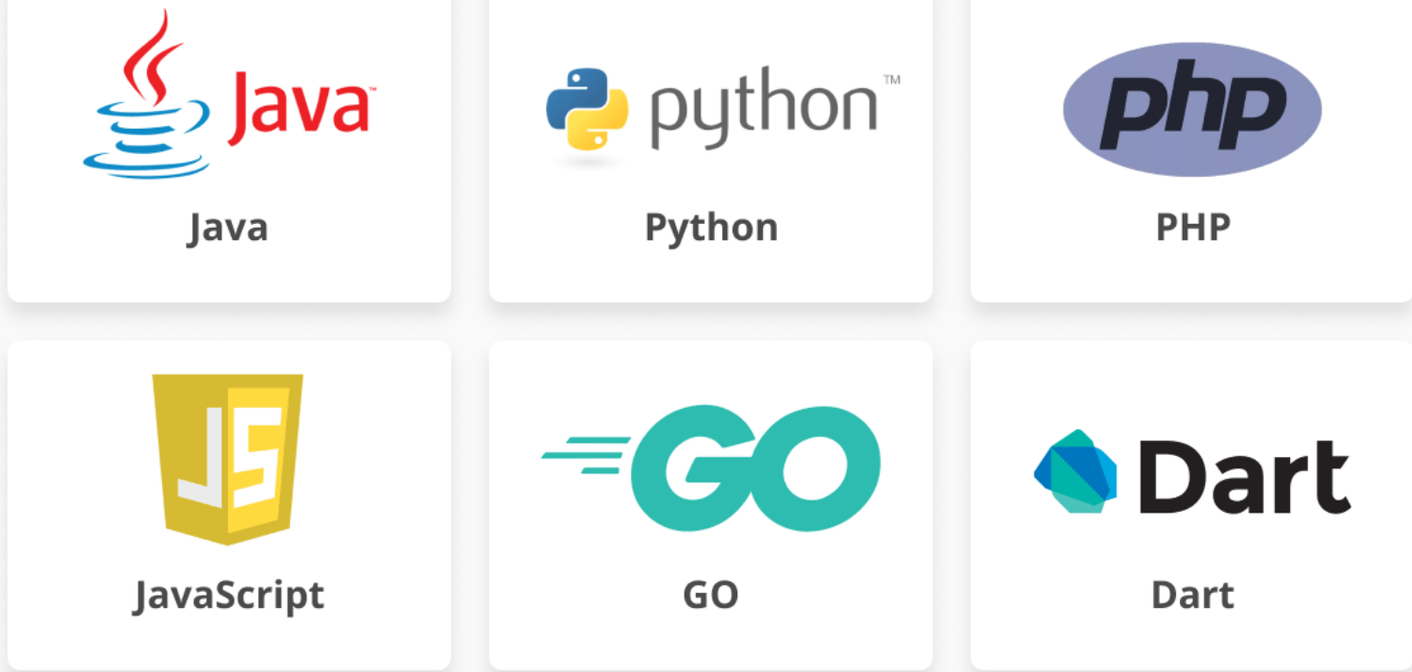
Frontend



Все, что происходит в браузере

Backend

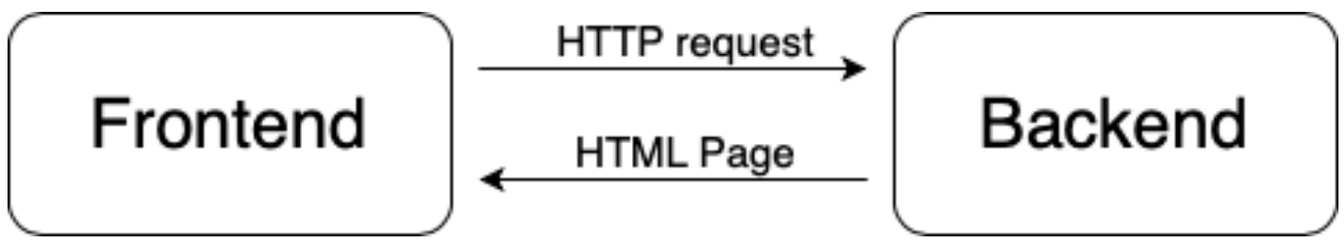
Most Popular Backend Languages



Все, что происходит на сервере

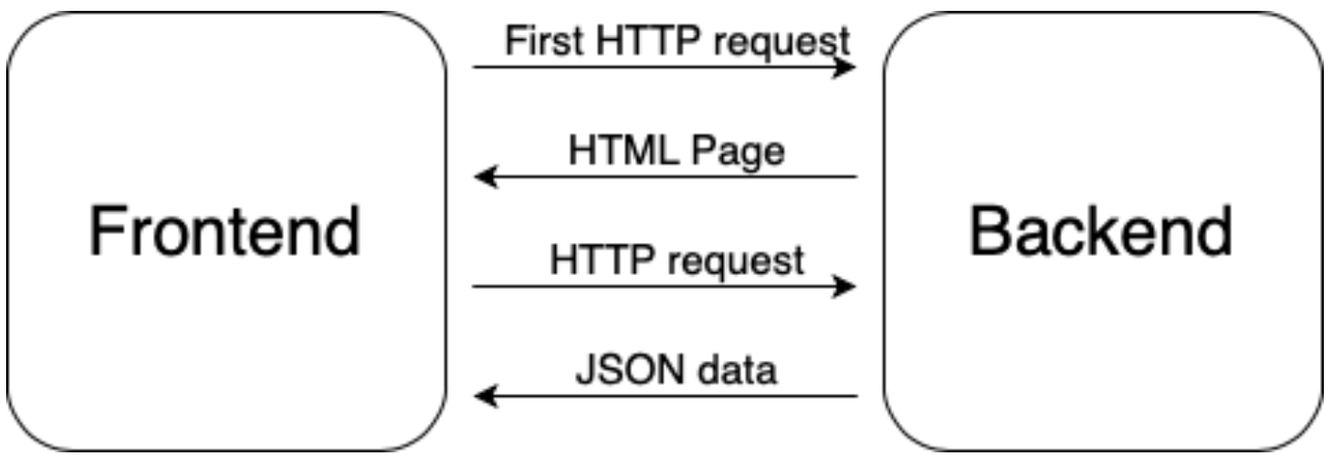
Варианты взаимодействия

- Клиент делает запрос на сервер
- Сервер отвечает статичной HTML страницей
- Для каждой страницы сайта нужно запросить новый HTML документ



Варианты взаимодействия

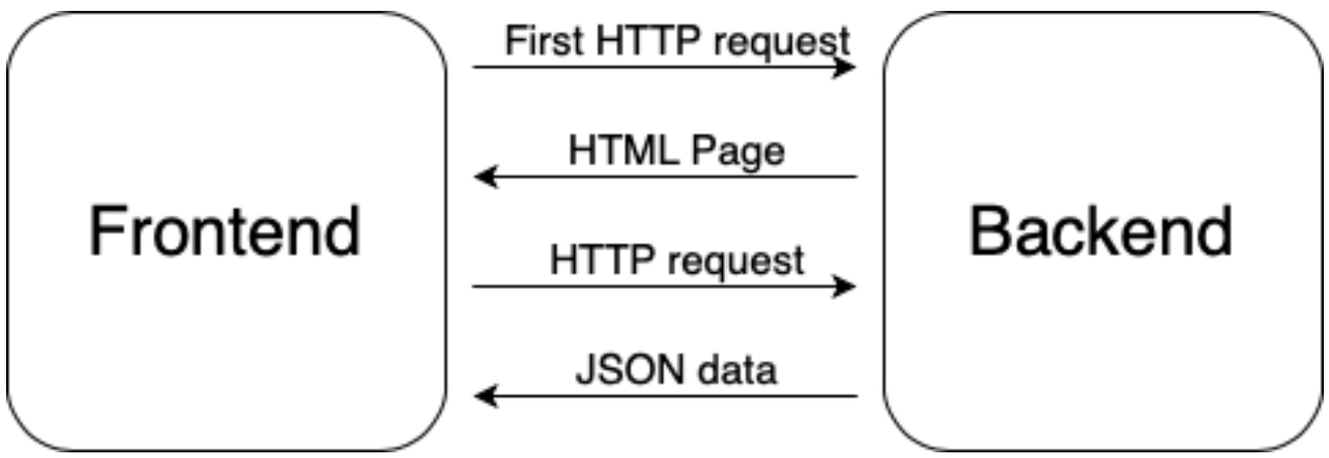
- При первом запросе клиента на сервер ему возвращается **готовый** HTML документ
- При взаимодействии пользователя со страницей данные подгружаются динамически
- При переходе на другую страницу запрашивается новый HTML документ



Варианты взаимодействия

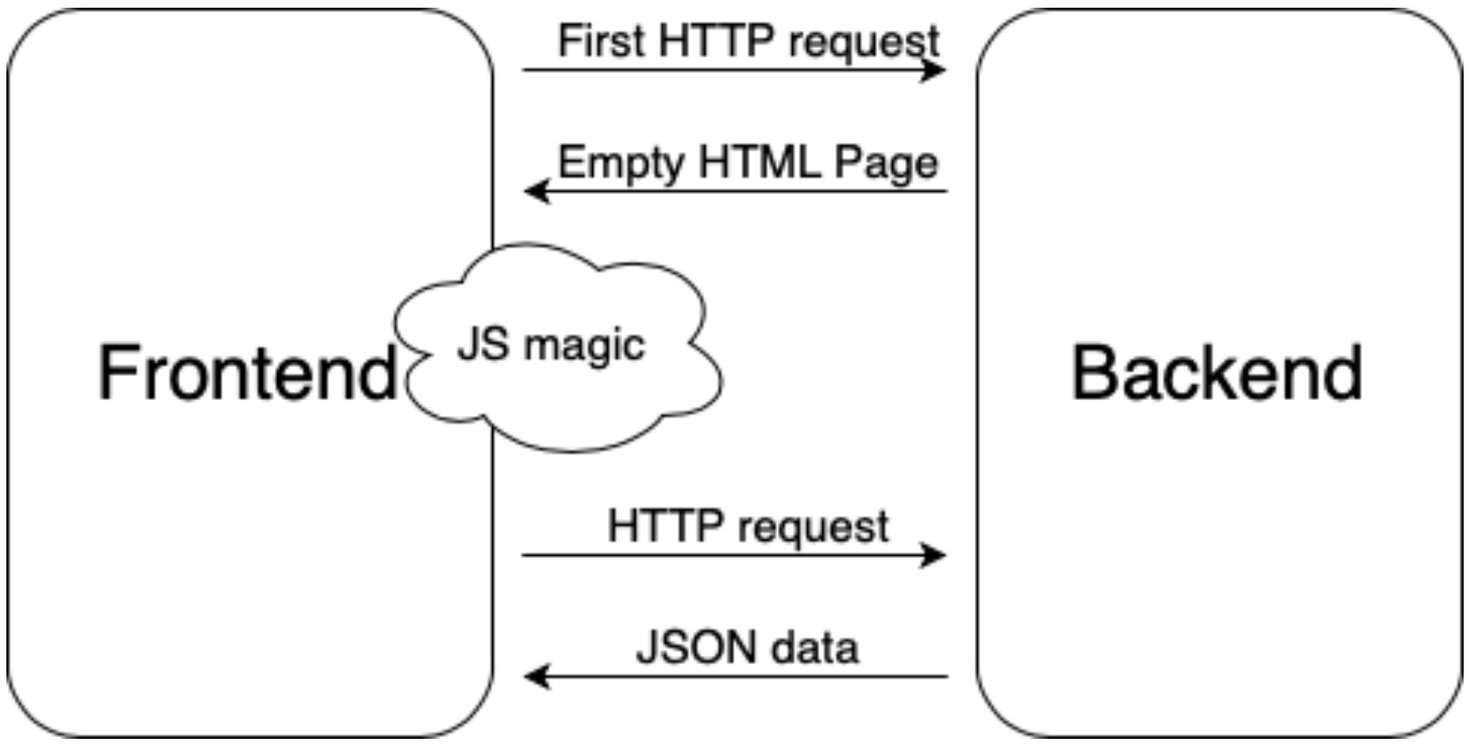
AJAX - async JS and XML

- При первом запросе клиента на сервер ему возвращается **готовый** HTML документ
- При взаимодействии пользователя со страницей данные подгружаются динамически
- При переходе на другую страницу запрашивается новый HTML документ



Варианты взаимодействия

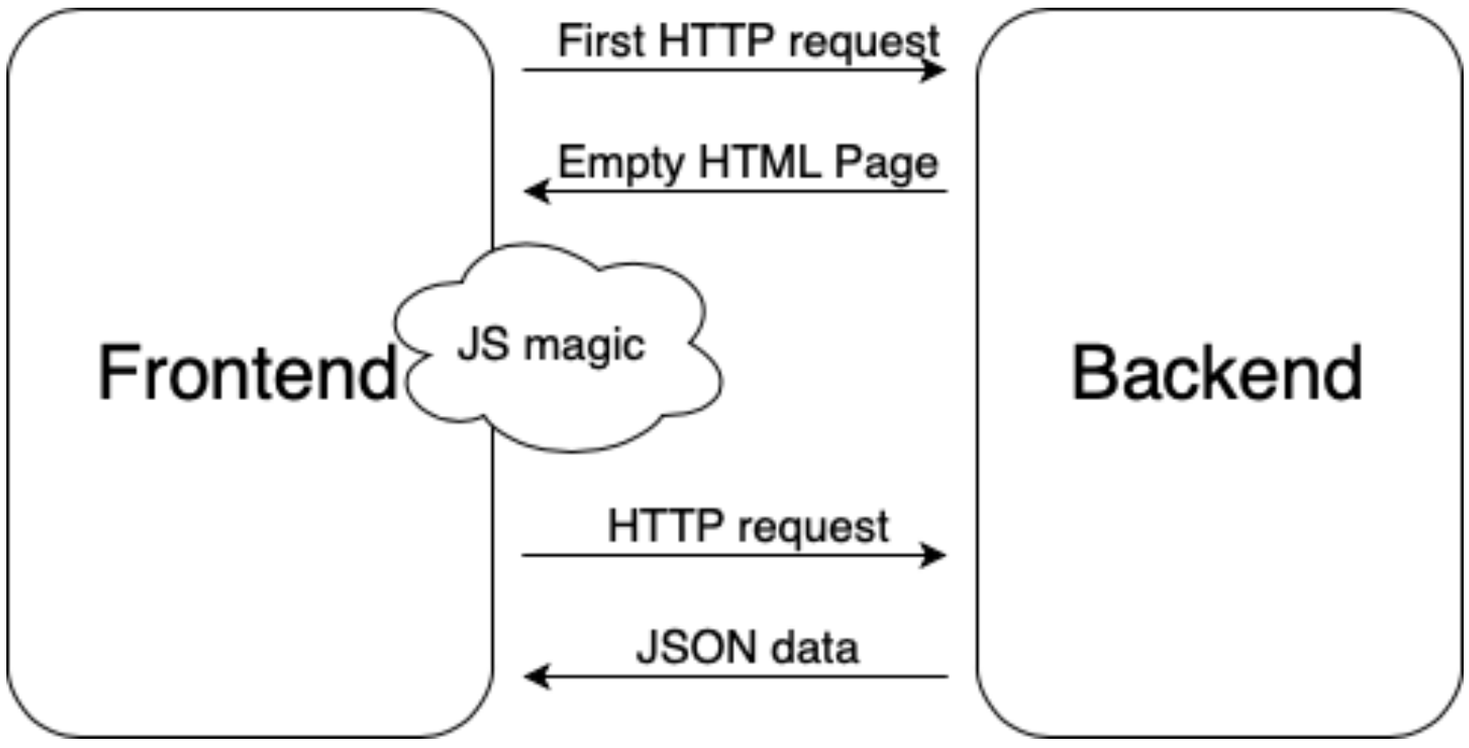
- При первом запросе клиента на сервер ему возвращается **пустой** HTML документ
- JS «отрисовывает» контент
- Новые данные подгружаются динамически
- При переходе по страницам не запрашивается новый HTML документ



Варианты взаимодействия

SPA - single page application

- При первом запросе клиента на сервер ему возвращается **пустой** HTML документ
- JS «отрисовывает» контент
- Новые данные подгружаются динамически
- При переходе по страницам не запрашивается новый HTML документ

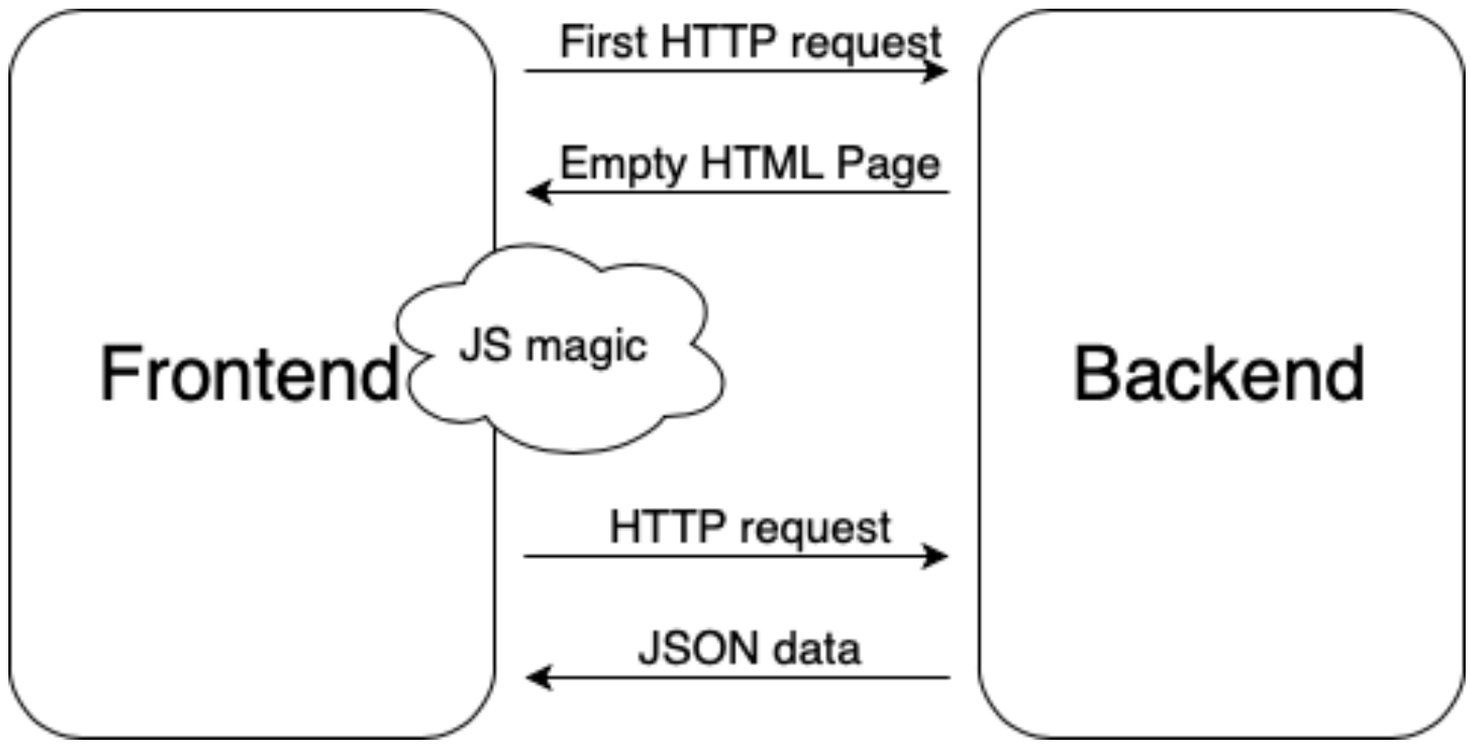


Варианты взаимодействия

SPA - single page application

Плюсы:

- Богатый и отзывчивый пользовательский интерфейс
- Возможность переиспользования одного сервера для desktop’ой версии сайта, мобильного сайта и приложения
- Быстрый переход между страницами

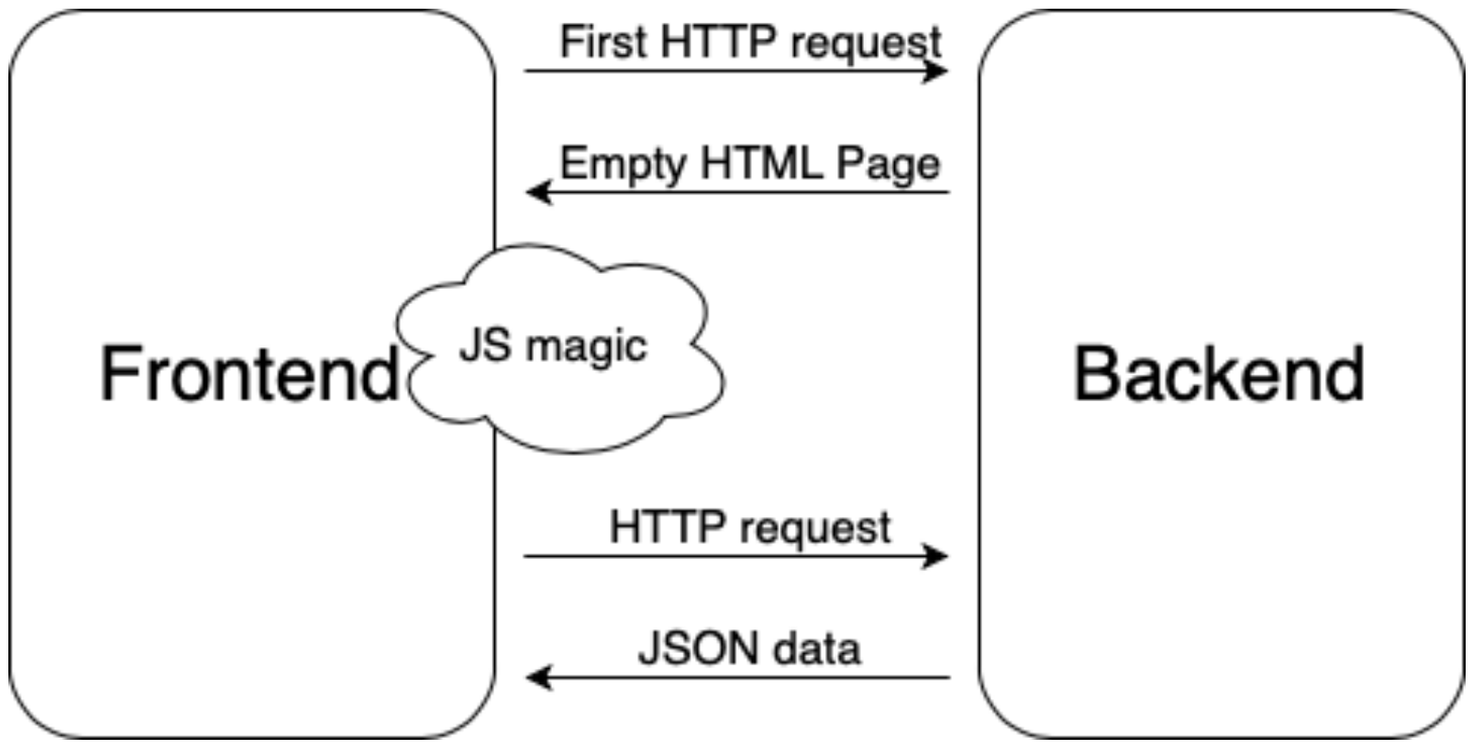


Варианты взаимодействия

SPA - single page application

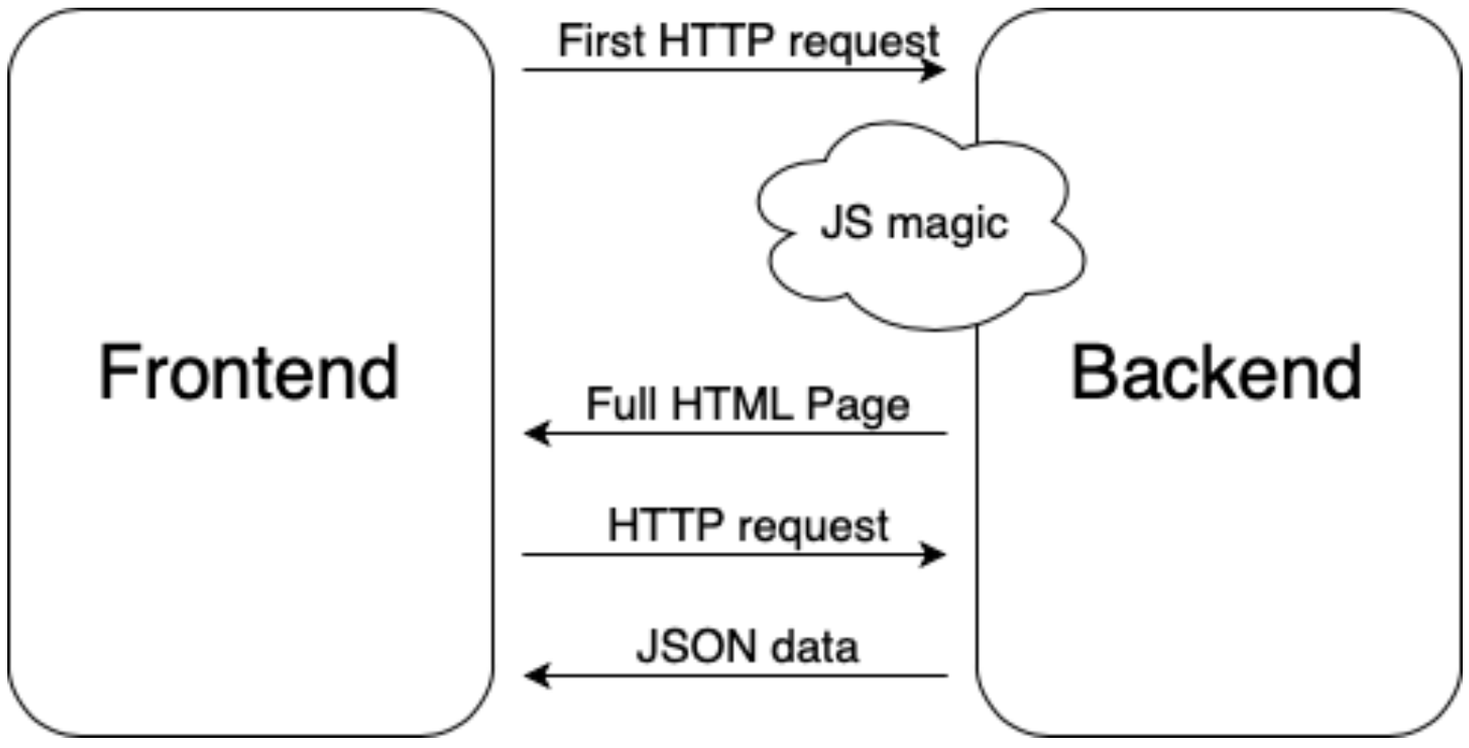
Минусы:

- Утечка памяти - плохой код может сильно нагрузить компьютер пользователя
- При первом посещении сайта нужно загрузить объемные js файлы => долгая первая загрузка
- «Невидимость» для поисковых роботов



Варианты взаимодействия

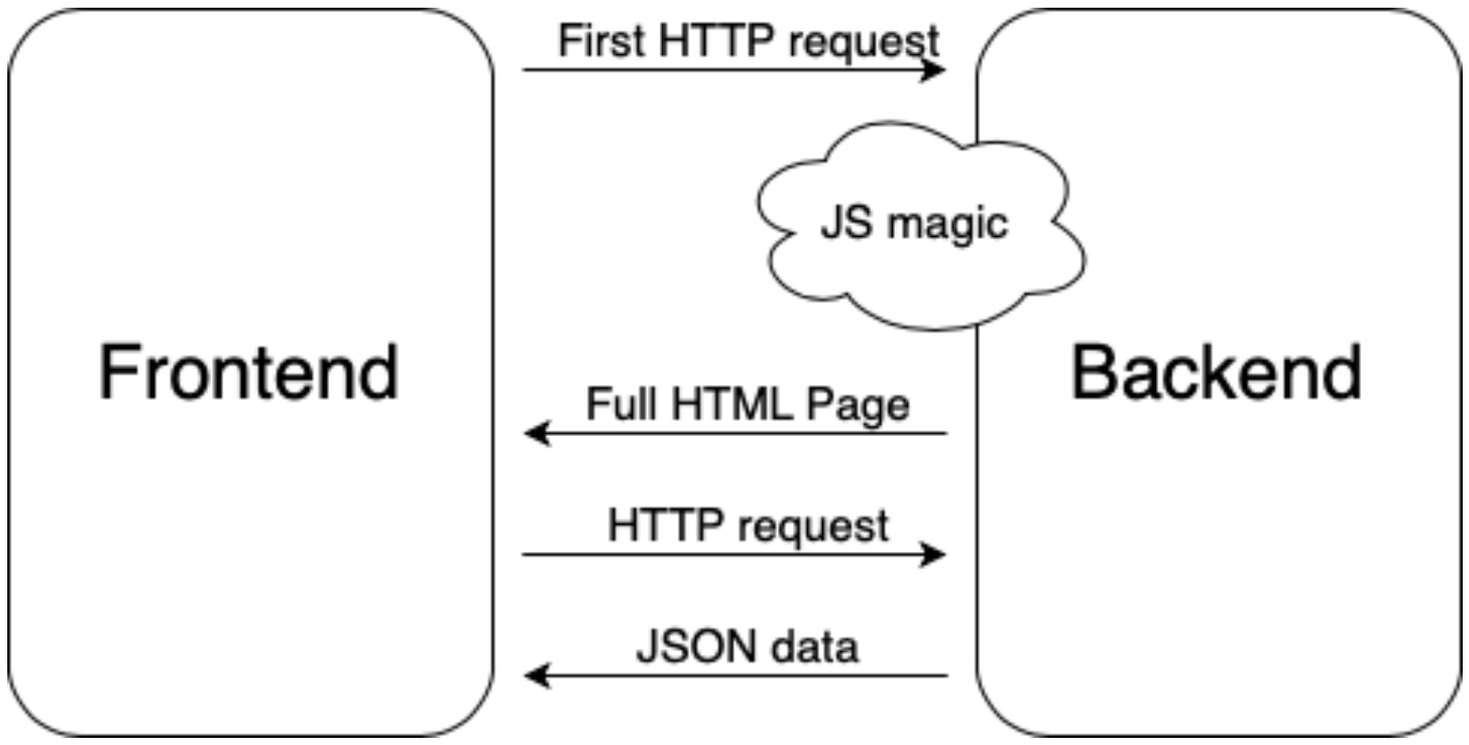
- Клиент делает первый запрос на сервер
- На сервере с помощью JS наполняется HTML и отдается пользователю
- Новые данные подгружаются динамически



Варианты взаимодействия

SSR - server side rendering

- Клиент делает первый запрос на сервер
- На сервере с помощью JS наполняется HTML и отдается пользователю
- Новые данные подгружаются динамически

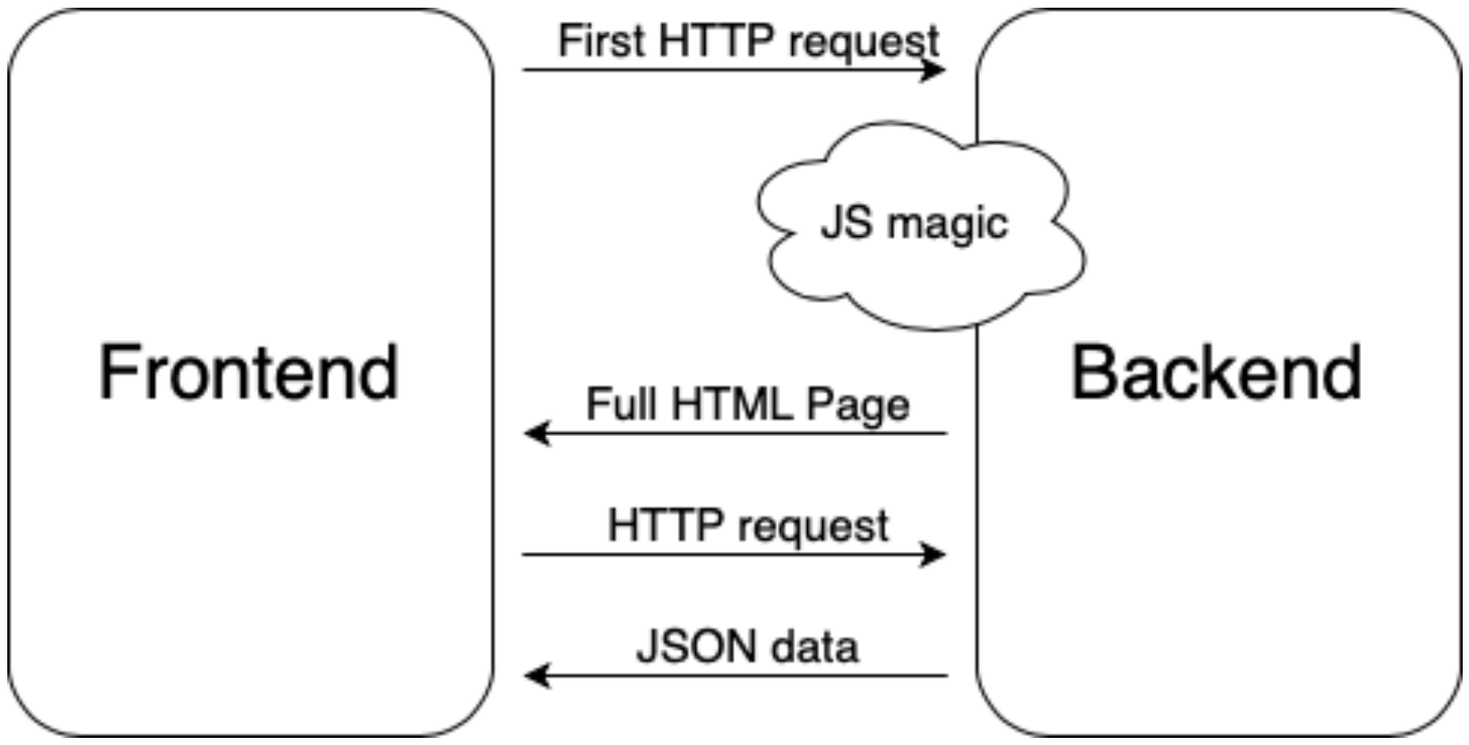


Варианты взаимодействия

SSR - server side rendering

Плюсы:

- Первая загрузка страницы быстрее, чем у SPA
- Доступен для поисковых роботов

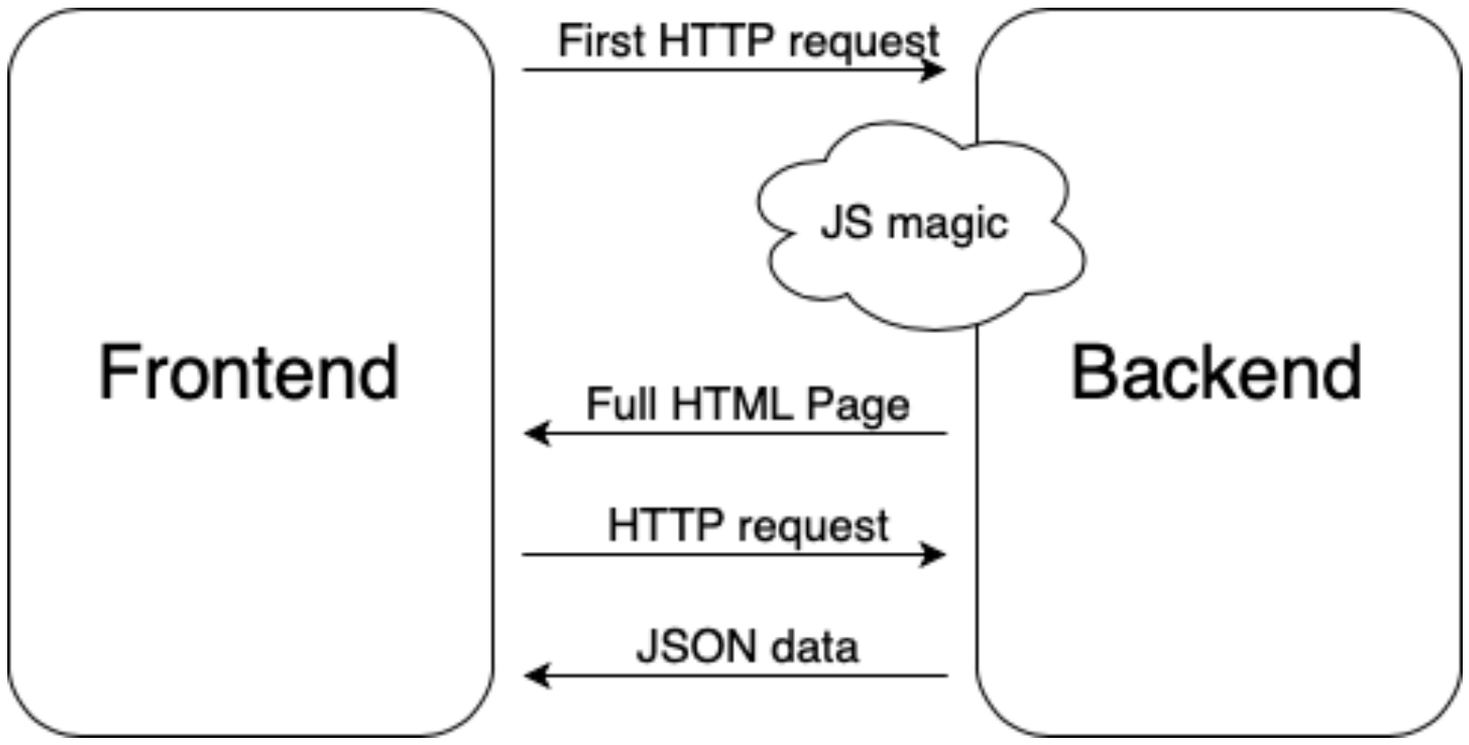


Варианты взаимодействия

SSR - server side rendering

Минусы:

- Большая нагрузка на сервер
- Сложнее развертывать



HTML

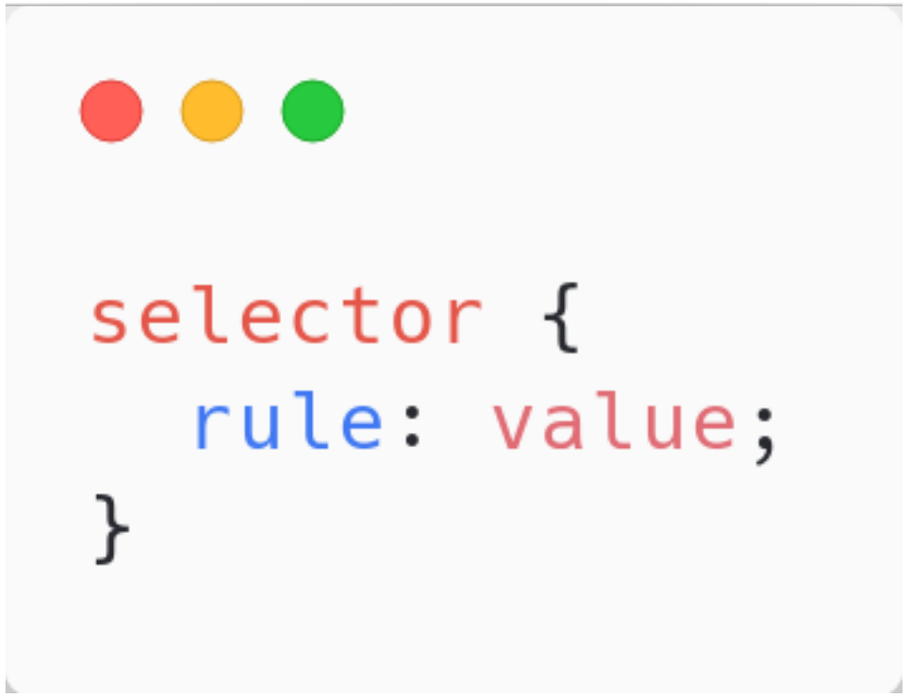
```
<a href="https://example.com">Hello world!</a>
```

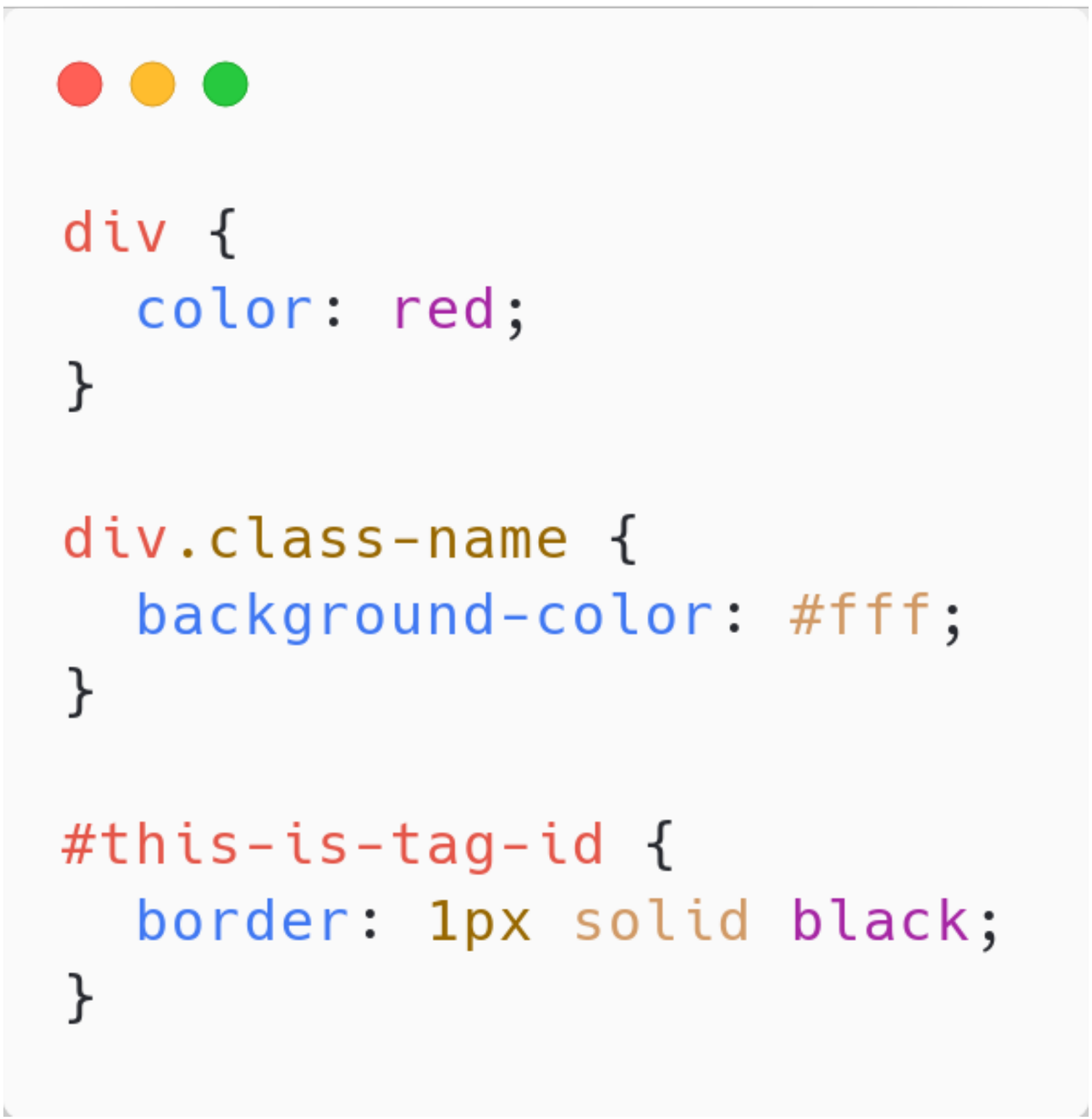
Основные теги

- <html>
- <head>
- <body>
-
-
- <a>
- +
- +
- <input>
- <button>

```
<!DOCTYPE html>
<html>
    <head>
        <title>Page Title</title>
    </head>
    <body>
        <h1>This is a Heading</h1>
        <p>This is a paragraph.</p>
    </body>
</html>
```

CSS


selector {
rule: value;
}



```
div {
   color: red;
}

div.class-name {
   background-color: #fff;
}

#this-is-tag-id {
   border: 1px solid black;
}
```

```css
* {
  border-color: white;
}

*.class-name {
  margin: 10px;
}

.class-name {
  padding: 20px;
}
```

```
[href] {
  rule: value;
}

[href="http://example.com"] {
  rule: value;
}

[href*="http://example.com"] {
  rule: value;
}
```

Псевдоклассы

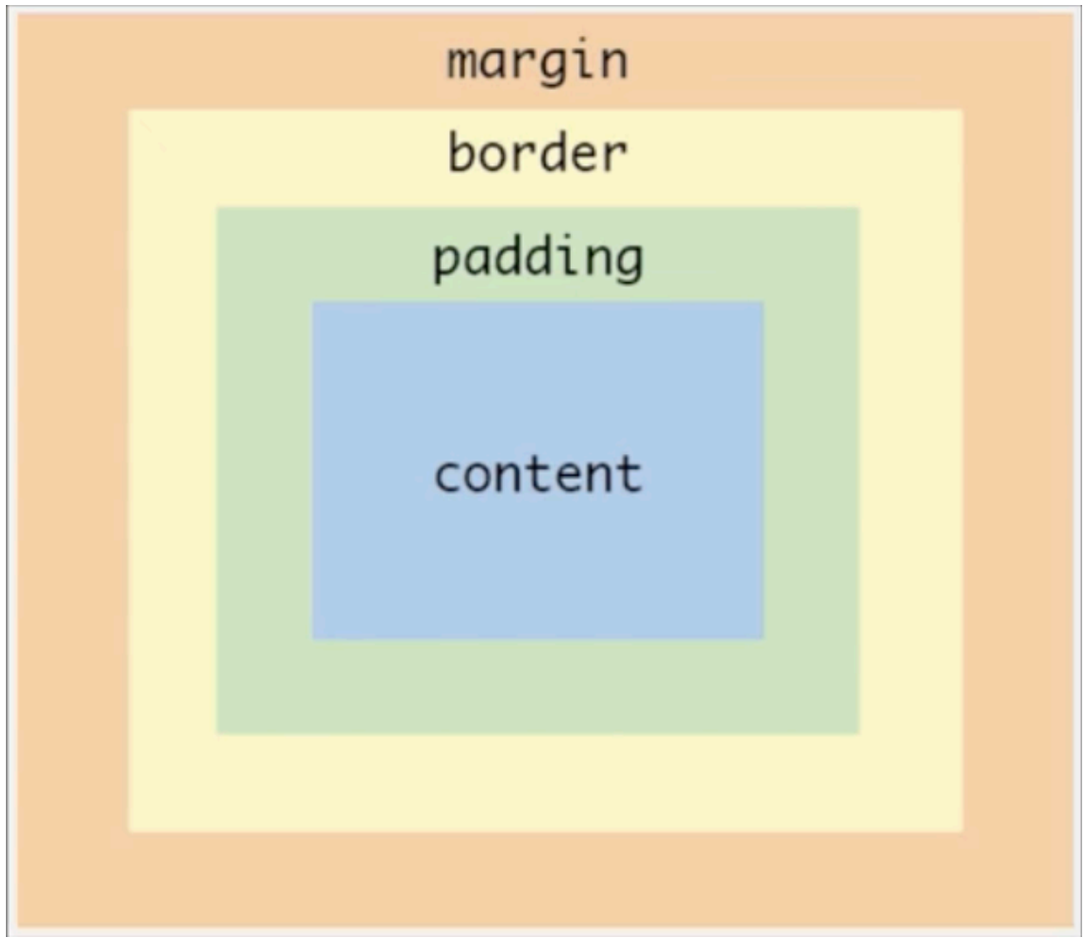
```
a:link { color: blue; }
a:hover { color: #123ffa; }
a:active { color: black; }
a:visited { color: red; }
a:focus { color: green; }
```

Структурные псевдоклассы

```
div:first-child {
  rule: value;
}

div:last-child {
  rule: value;
}

div:only-child {
  rule: value;
}

div:nth-child(2n) {
  rule: value;
}
```

Модель отображения

Все есть прямоугольники



Block элементы

- Всегда начинаются на новой строке и занимают всю доступную ширину
- div
- p
- h1, h2, ...

Inline элементы

- Не реагируют на ширину и высоту
- Не реагируют на вертикальные margins
- Не затрагивают высоту строки
- span
- a

Inline-block элементы

- Реагируют на ширину и высоту
- Реагируют на вертикальные margins
- Затрагивают высоту строки
- button

Системы позиционирования

selector {
position:

- static - значение по умолчанию
- relative - реагирует на top/bottom, left/right
- absolute - выводится из потока, содержащим блоком является ближайший предок с position = relative | absolute | fixed
- fixed - абсолютно позиционированный элемент относительно viewport
- sticky - ведет себя как relative, пока не дойдет до определенной границы, после нее ведет себя как fixed

```html
<body>
  <div class="relative-parent">
    <div class="static">Hello, I'm static</div>
    <div class="relative">Hello, I'm relative</div>
    <div class="absolute">ABSOLUTE</div>
  </div>
</body>
```


```css
:root {
  width: 300px;
}

.relative-parent {
  position: relative;
}

.static {
  background: green;
}

.relative {
  position: relative;
  background: blue;
  left: 20px;
}

.absolute {
  position: absolute;
  background: red;
  top: 10px;
  left: 50%;
}
```