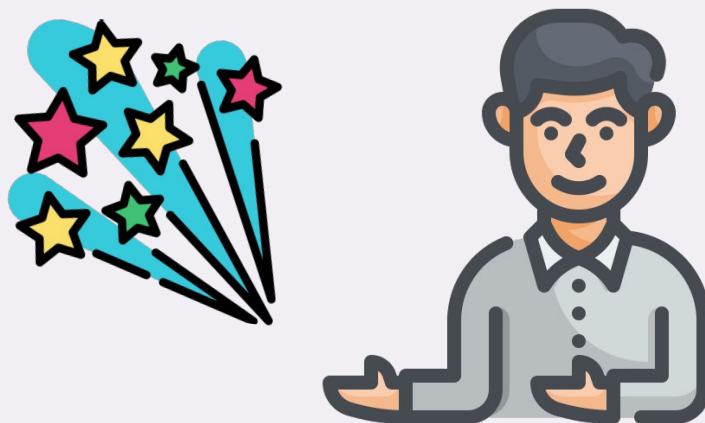


Fullstack-разработка

JSON

Повторение

Добро пожаловать в
последний модуль
нашего курса!



На следующих
занятиях мы будем
соединять
клиент и сервер



Напишем последнюю аттестацию



Изучим фреймворк
Vue.js



и напишем
финальный проект

Цель урока:

Повторить основы
HTML, CSS и JS

Благодаря Bootstrap
мы много уроков
вообще не вспоминали
про



Время это исправить!

Никита

1000\$

Рюкзак

- Зелье жизни - 2
- Хлеб - 5

Магазин

- Звездный кристалл - 300
- Драконий глаз - 100
- Руна бессмертия - 50
- Голова медузы - 80
- Зелье жизни - 10
- Хлеб - 5

Теория

Вёрстка

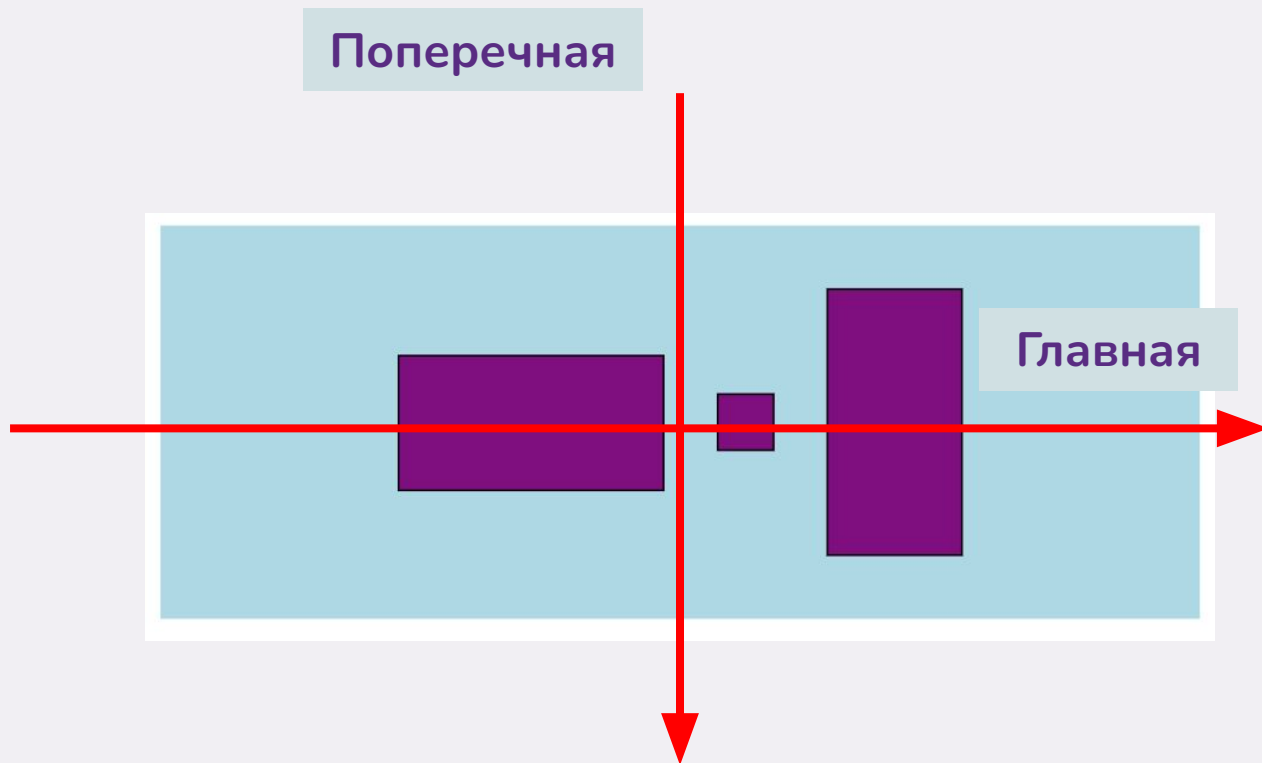
Главные теги

- header
- main
- footer
- section
- h1 - h5
- div
- span
- ul
- li
- a
- img
- input
- select
- textarea

Главные CSS-свойства

- font-size
- font-weight
- font-style
- font-family
- color
- background-color
- margin
- padding
- border
- width
- height

Flexbox



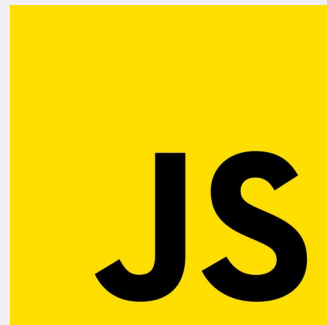
Практика

Вёрстка

Теория

Javascript

JavaScript в браузере
отвечает за
взаимодействие
между пользователем
и страницей.



Нашему магазину как раз не хватает интерактива.

Никита

1000\$

Рюкзак

- Яблоко - 2
- Хлеб - 5

Магазин

- Звездный кристалл - 300
- Дракониный глаз - 100
- Руна бессмертия - 50
- Голова медузы - 80
- Зелье жизни - 10
- Хлеб - 5

Самое главное

- `document.querySelector(`.class`)`
- `document.querySelectorAll(`.class`)`
- `node.innerHTML = 'Привет!';`
- `node.classList.add('hidden');`
- `node.addEventListener(
 'click', function () { ... }
)`

Практика

Вывод данных

Добавление в рюкзак

<script>

подключаем в head

```
<head>
```

```
...
```

```
<script src="js/index.js"
      defer
```

```
></script>
```

```
...
```

```
</head>
```

путь к файлу

выполнится после
загрузки страницы

document

Это специальный объект, который предоставляет доступ из JS-кода к HTML-документу.

Именно с помощью него мы находим нужные элементы на странице.

```
document.querySelector(`#id`)
```

```
document.querySelector(`.class`)
```

innerHTML

1. Добавить атрибут id

```
<div id="output">  
  <!-- место для данных -->  
</div>
```

index.html

2. Найти элемент на странице по id

```
let outputNode = document.querySelector(`#output`);
```

3. Записать ответ в этот элемент

```
outputNode.innerHTML = `Это увидит  
пользователь!`;
```

js/index.js

addEventListener

1. Добавить кнопку с атрибутом id

index.html

```
<button id="click">Кнопка</button>
```

2. Найти кнопку на странице по id

```
let buttonNode = document.querySelector(`#click`);
```

3. Подписаться на событие Клик по кнопке

```
buttonNode.addEventListener(`click`, function () {  
    console.log(`Меня нажали!`);  
});
```

js/index.js

Строка-шаблон

```
let string = `  
  <h1>${title}</h1>  
    
`;  
;
```

Внутри косых кавычек можно
подставлять переменные.

Это называется **шаблонизация**.

Изменения класса

По клику:

```
// добавить класс
```

```
node.classList.add("circle");
```

```
// удалить класс
```

```
node.classList.remove("square");
```

До

```
<div id="node"  
      class="square">  
</div>
```

После

```
<div id="node"  
      class="circle">  
</div>
```

Коллекция

```
let people =  
  document.querySelectorAll(`.person`)
```

Коллекция – это все элементы на странице, которые удалось найти по **селектору** через метод **querySelectorAll**

События для коллекции

```
let buttons = document.querySelectorAll(`.button`);

for (let i = 0; i < buttons.length; i++) {
  let button = buttons[i];
  button.addEventListener(`click`, function () {
    console.log(`Нажали на кнопку ${i}`);
  });
}
```

Цель урока:

- ✓ Повторить основы HTML, CSS и JS