

Описание курса



JavaScript. Уровень 1. Основы JavaScript

Тема	Часы
Модуль 1. Основы программирования <i>Операторы, переменные, типы данных...</i>	4
Модуль 2. Управляющие конструкции <i>if-else, for, while, switch...</i>	4
Модуль 3. Функции <i>Синтаксис функции, аргументы, области видимости...</i>	4
Модуль 4. Объектные типы <i>Свойства, методы, массивы...</i>	4
Модуль 5. Объектно-ориентированное программирование <i>Функция конструктор, прототипы, классы...</i>	4
Модуль 6. Дополнительная информация <i>Работа со строками, регулярные выражения, JSON...</i>	4

Программа курса предусматривает лабораторные работы по каждой теме, а также выполнение домашних заданий и контроль знаний



Модуль 4 Объектные типы

Басов Денис

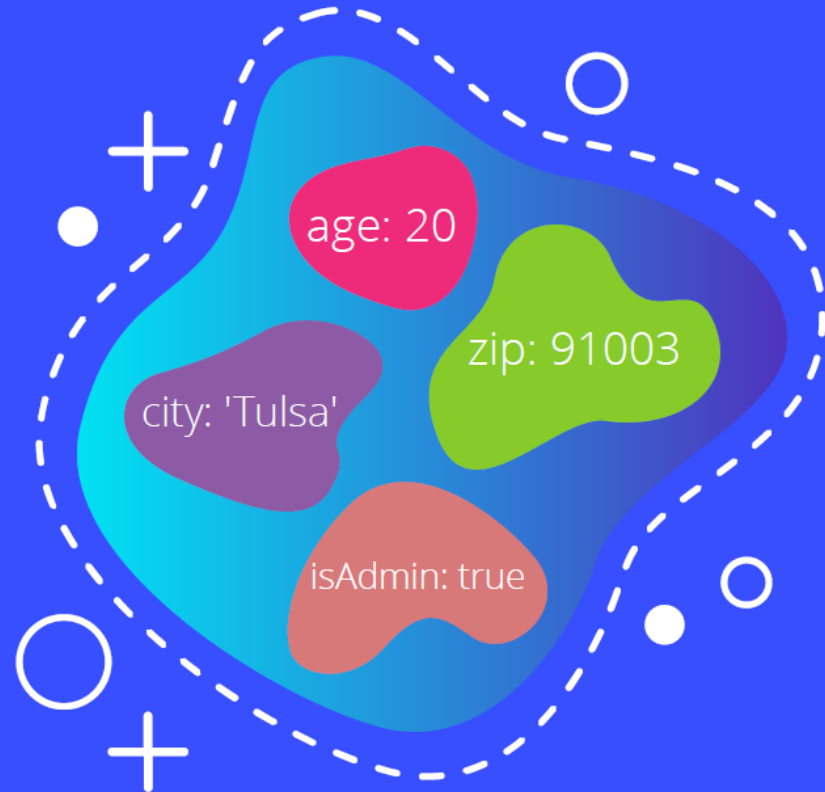


В этом модуле мы рассмотрим:

- ✓ Объектный тип: Объект (Object)
- ✓ Свойства объекта
- ✓ Методы объекта
- ✓ Объектный тип: Массив (Array)
- ✓ Методы для работы с массивами

Объекты


- Объекты - это коллекции свойств
- Свойства - это пары ключ-значение
- Для получения доступа к значениям свойств используются ключи



КАК ВЫ СОХРАН ИТЕ ЭТО?



ОБЪЕКТ !



```
const fitBitData = {  
  totalSteps      : 308727,  
  totalMiles      : 211.7,  
  avgCalorieBurn  : 5755,  
  workoutsThisWeek : '5 of 7',  
  avgGoodSleep    : '2:13'  
};
```

СВОЙСТВО =
КЛЮЧ
+
ЗНАЧЕНИЕ





пары ключ - значение

username: —————→ 'crazyCatLady'

upvotes: —————→ 7

text —————→ 'great post!'

СЛОВАРЬ

страна: Россия

город: Новый
Уренгой

улица: Ломоносова

дом: 45

страна: Россия

город: Великий
Новгород

улица: Пушкина

дом: 13



ИСПОЛЬЗУЮТСЯ ВСЕ ТИПЫ ДАННЫХ !



```
let comment = {  
  username      : 'sillyGoose420',  
  downVotes     : 19,  
  upVotes       : 214,  
  netScore       : 195,  
  commentText   : 'Tastes like chicken lol',  
  tags: ['#hilarious', '#funny', '#silly'],  
  isGilded: false  
};
```





ДОСТУП К ДАННЫМ



```
const palette = {  
  red: '#eb4d4b',  
  yellow: '#f9ca24',  
  blue: '#30336b'  
}
```



```
palette.red //"#eb4d4b"
```

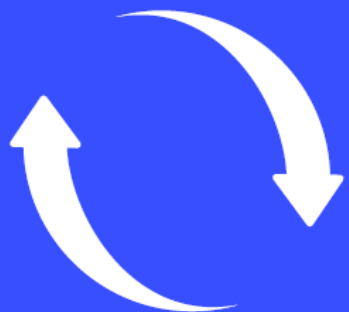


```
palette['blue'] //"#30336b"
```



```
let color = 'yellow';  
palette[color] //"#f9ca24"
```

ОБНОВЛЕНИЕ И ДОБАВЛЕНИЕ СВОЙСТВ



```
const fitBitData = {  
  totalSteps      : 308727,  
  totalMiles      : 211.7,  
  avgCalorieBurn  : 5755,  
  workoutsThisWeek : '5 of 7',  
  avgGoodSleep    : '2:13'  
};  
//Updating properties:  
fitBitData.workoutsThisWeek = '6 of 7';  
fitBitData.totalMiles += 7.5;  
  
//Adding a new property  
fitBitData.heartStillBeating = true;
```





МАССИВЫ + ОБЪЕКТЫ

```
const shoppingCart = [  
  {  
    product: 'Jenga Classic',  
    price: 6.88,  
    quantity: 1,  
  },  
  {  
    product: 'Echo Dot',  
    price: 29.99,  
    quantity: 3,  
  },  
  {  
    product: 'Fire Stick',  
    price: 39.99,  
    quantity: 2,  
  },  
]
```

```
const student = {  
  firstName: 'David',  
  lastName: 'Jones',  
  strengths: ['Music', 'Art'],  
  exams: {  
    midterm: 92,  
    final: 88,  
  },  
}
```



МЕТОДЫ ОБЪЕКТОВ

```
const math = {  
  multiply : function(x, y) {  
    return x * y;  
  },  
  divide   : function(x, y) {  
    return x / y;  
  },  
  square   : function(x) {  
    return x * x;  
  }  
};
```

Мы можем добавить функцию
как свойство объекта.

Мы называем их методами!

THIS В МЕТОДАХ

Ключевое слово ***this*** используется для получения доступа к свойствам текущего объекта

```
const person = {  
  first: 'Robert',  
  last: 'Herjavec',  
  fullName() {  
    return `${this.first} ${this.last}`  
  }  
}  
  
person.fullName(); //"Robert Herjavec"  
person.last = "Plant";  
person.fullName(); //"Robert Plant"
```



Массивы

Упорядоченные коллекции данных

- Жилой массив
- Лесной массив
- Горный массив



Создание массивов



```
// To make an empty array  
let students = [];
```

```
//An array of strings  
let colors = ['red', 'orange', 'yellow'];
```

```
//An array of numbers  
let lottoNums = [19, 22, 56, 12, 51];
```

```
//A mixed array  
let stuff = [true, 68, 'cat', null];
```





Массивы индексируются



```
let colors = ['red', 'orange', 'yellow', 'green'];
```

```
colors.length //4
```

```
colors[0] // 'red'
```

```
colors[1] // 'orange'
```

```
colors[2] // 'yellow'
```

```
colors[3] // 'green'
```

```
colors[4] // 'undefined'
```



Модификация массивов



```
let colors = ['rad', 'orange', 'green', 'yellow'];  
  
colors[0] = 'red';  
  
colors[2] = 'yellow';  
colors[3] = 'green';  
  
colors[4]; //undefined  
colors[4] = 'blue';  
//["red", "orange", "yellow", "green", "blue"]
```



Методы для работы с массивами

- `Push` - добавляет элемент в конец массива
- `Pop` - удаляет элемент с конца массива
- `Shift` - удаляет элемент с начала массива
- `Unshift` - добавляет элемент в начало

Еще больше методов!

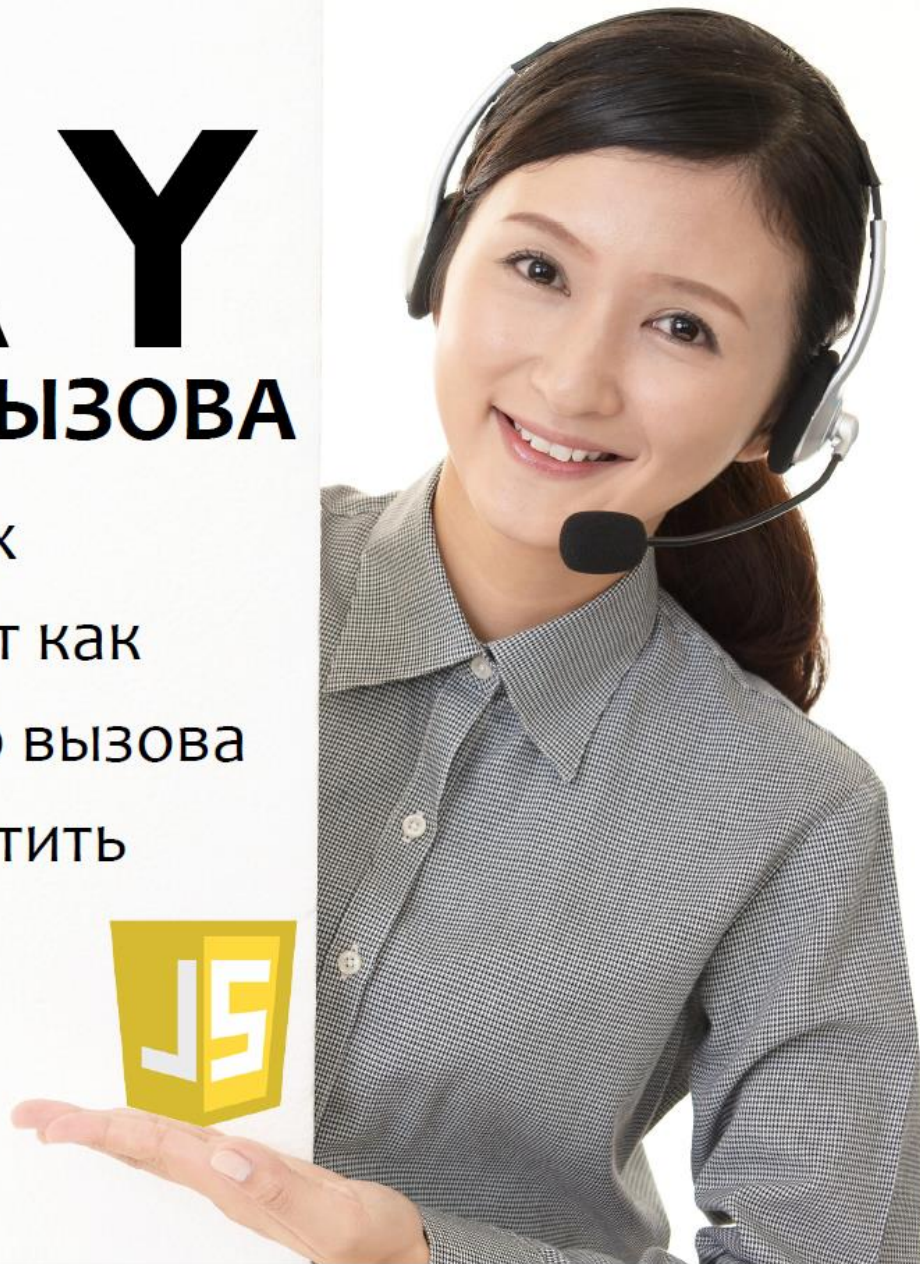
- `concat` - объединяет массивы
- `includes` - ищет значение
- `indexOf` - возвращает индекс элемента
- `join` - преобразует массив в строку
- `reverse` - переворачивает массив
- `slice` - копирует часть массива
- `splice` - удаляет/заменяет элементы
- `sort` - сортирует массив



ARRAY

МЕТОДЫ ОБРАТНОГО ВЫЗОВА

- У массивов много встроенных методов, которые принимают как аргумент функцию обратного вызова
- Эти методы позволяют упростить работу с массивами





FOREACH

```
const nums = [9, 8, 7, 6, 5, 4, 3, 2, 1];

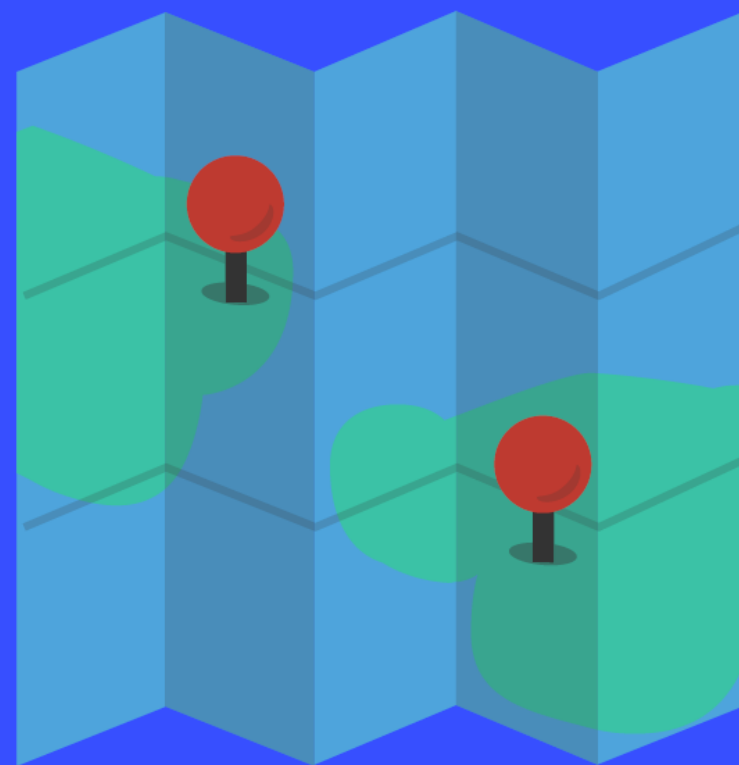
nums.forEach(function (n) {
  console.log(n * n)
  //prints: 81, 64, 49, 36, 25, 16, 9, 4, 1
});

nums.forEach(function (el) {
  if (el % 2 === 0) {
    console.log(el)
    //prints: 8, 6, 4, 2
  }
})
```

Принимает функцию обратного вызова .
Вызывает функцию один раз для каждого элемента массива .

MAP

Создает новый массив с результатами вызова функции обратного вызова на каждом элементе массива



MAP



```
const texts = ['rofl', 'lol', 'omg', 'ttyl'];  
const caps = texts.map(function (t) {  
  return t.toUpperCase();  
})  
texts; //["rofl", "lol", "omg", "ttyl"]  
caps;  //["ROFL", "LOL", "OMG", "TTYL"]
```



Спасибо
за внимание!
Ваши вопросы...

