

# Описание курса



## JavaScript. Уровень 1. Основы JavaScript

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

## *Объектно-ориентированное программирование*

Басов Денис



## В этом модуле мы рассмотрим:

- ✓ Функции-конструкторы объектов
- ✓ Прототипы
- ✓ Классы
- ✓ Наследование



# Создание функции-конструктора

```
1  /**
2   * Функция-конструктор и прототипное наследование
3   */
4  // создаем функцию-конструктор
5  let Person = function (firstName, lastName, age) {
6      this.firstName = firstName;
7      this.lastName = lastName;
8      this.age = age;
9  };
10
```



# Создание объектов с помощью конструктора

```
1  /**
2   * Функция-конструктор и прототипное наследование
3   */
4  // создаем функцию-конструктор
5  let Person = function (firstName, lastName, age) {
6      this.firstName = firstName;
7      this.lastName = lastName;
8      this.age = age;
9  };
10
11 // создаем объекты по шаблону функции-конструктора
12 let ivan = new Person("Иван", "Иванов", 44);
13 let peter = new Person("Питер", "Паркер", 15);
14
```



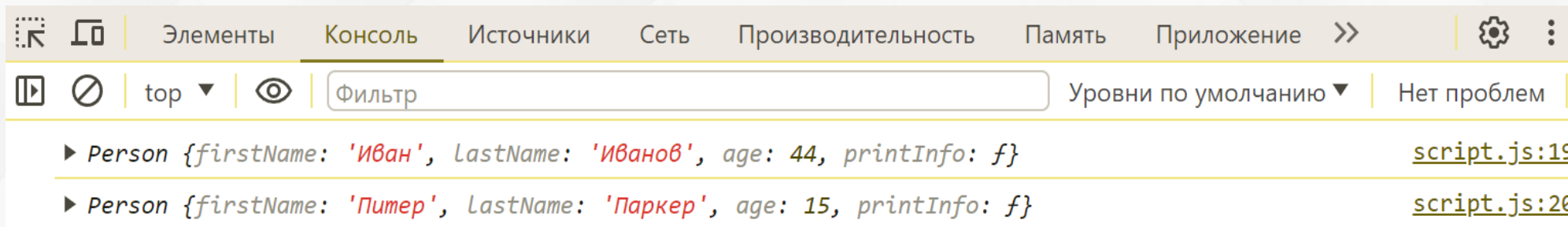
# Добавляем в конструктор метод

```
1  /**
2   * Функция-конструктор и прототипное наследование
3   */
4  // создаем функцию-конструктор
5  let Person = function (firstName, lastName, age) {
6      this.firstName = firstName;
7      this.lastName = lastName;
8      this.age = age;
9
10     this.printInfo = function () {
11         console.log(`Имя: ${this.firstName}. Фамилия: ${this.lastName}.
12         Возраст ${this.age}.`);
13     };
14 }
```

# Метод добавлен каждому объекту. Не эффективно.

```
// создаем объекты по шаблону функции-конструктора
let ivan = new Person("Иван", "Иванов", 44);
let peter = new Person("Питер", "Паркер", 15);

console.log(ivan);
console.log(peter);
```





# Прототипное наследование в JS

```
23 // Проверяем прототип Person
24 console.log(Person.prototype);
25 console.dir(Person.prototype);
26
27 // записываем в прототип метод для печати данных об объекте
28 Person.prototype.printInfo = function () {
29     console.log(`Имя: ${this.firstName}. Фамилия: ${this.lastName}.
30     Возраст ${this.age}.`);
31 };
32
33 // проверяем
34 ivan.printInfo();
35 peter.printInfo();
```





# Создание объектов с помощью конструктора

```
1  /**
2   * Функция-конструктор и прототипное наследование
3   */
4  // создаем функцию-конструктор
5  let Person = function (firstName, lastName, age) {
6      this.firstName = firstName;
7      this.lastName = lastName;
8      this.age = age;
9  };
10
```



# ES6 классы. Конструктор класса

```
36  /**
37   * Классы ES6
38   */
39  class Student {
40      constructor(firstName, lastName, age) {
41          this.firstName = firstName;
42          this.lastName = lastName;
43          this.age = age;
44      }
45  }
```

# Создание методов в классе



```
39 class Student {
40     constructor(firstName, lastName, age) {
41         this.firstName = firstName;
42         this.lastName = lastName;
43         this.age = age;
44     }
45
46     greet() {
47         console.log(`Привет, ${this.firstName}!!!`);
48     }
49
50     printInfo() {
51         console.log(`Имя: ${this.firstName}. Фамилия: ${this.lastName}.
52         Возраст ${this.age}.`);
53     }
54 }
```



# Создание объекта по шаблону класса

```
46     greet() {  
47         console.log(`Привет, ${this.firstName}!!!`);  
48     }  
49  
50     printInfo() {  
51         console.log(`Имя: ${this.firstName}. Фамилия: ${this.lastName}.  
52         Возраст ${this.age}`);  
53     }  
54  
55     let anna = new Student("Анна", "Петрова", 22);  
56     anna.greet();  
57     anna.printInfo();  
58
```

# Наследование классов в JavaScript



```
59  /**
60   * Наследование Классы ES6
61   */
62  class NewStudent extends Student {
63      constructor(firstName, lastName, age, schoolNumber) {
64          super(firstName, lastName, age);
65          this.schoolNumber = schoolNumber;
66      }
67
68      introduce() {
69          console.log(`Меня зовут ${this.firstName}, мне ${this.age} лет,
70                     я из школы ${this.schoolNumber}.`);
71      }
72  }
73
74  let sarah = new NewStudent("Сара", "Коннор", 18, 23);
75  sarah.introduce();
76  sarah.printInfo();
```

**Спасибо**  
**за внимание!**  
Ваши вопросы...

