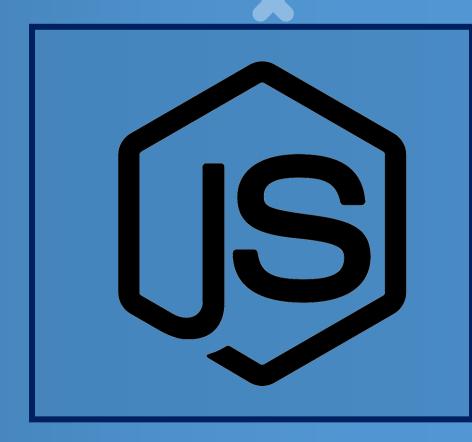


Классы и объекты



ООП

Когда рассматривают ООП в различных языках программирования, то обычно говорят о создании объектов на основе классов. С 2015 года (стандарт ES6) классы появились и в JavaScript, хотя и до этого времени в нем можно было создавать объекты.

Объекты

В прошлых темах мы работали с примитивными данными - числами, строками, но данные не всегда представляют примитивные типы. Например, если в нашей программе нам надо описать сущность человека, у которого есть имя, возраст, пол и так далее, то естественно мы не сможем представить сущность человека в виде числа или строки.

```
var cheloveck = {
  age: 32,
  name : "Name",
  gender : "w"
console.log(cheloveck)
```

ДЛЯ ДОБАВЛЕНИЯ ПАР «КЛЮЧ-ЗНАЧЕНИЕ» В ОБЪЕКТ МОЖНО ИСПОЛЬЗОВАТЬ ДВА ПОДХОДА.

Первый подход через стандартный

```
var cheloveck = {
  age : 32,
  name : "Name"
cheloveck["gender"] = "man"
console.log(cheloveck)
```

Второй подход

```
var cheloveck = {
  age: 32,
  name : "Name"
cheloveck.gender = "man"
console.log(cheloveck)
```

Вы можете использовать любой из двух способов, но если у вас возникнет необходимость хранить в качестве ключа строку, содержащую с пробелами, то тогда второй синтаксис не подойдет.

```
var cheloveck = {
  age : 32,
  name : "Name"
cheloveck["name with space"] = "man"
console.log(cheloveck)
```

При создании объекта можно внутрь вставить другой объект.

```
var cheloveck = {
  age : 32,
  name : "Name",
  sobaka : {
    nick : "Sam",
    pol : false
console.log(cheloveck)
```

Удаление свойств

```
var cheloveck = {
  age : 32,
  name : "Name",
  del1 : "123",
  del2: 123
console.log(cheloveck)
delete cheloveck.del1
delete cheloveck["del2"]
console.log(cheloveck)
```

Проверка существования свойства внутри

объекта

```
Она
возвращает
true, если
свойство
есть внутри
Объекта.
```

```
var cheloveck = {
  age : 32,
  name : "Name"
if ("Age" in cheloveck) {
  console.log("exist")
else {
  console.log("not exist")
```

Просмотр всех свойств внутри объекта

```
var cheloveck = {
  age: 32,
  name: "Name"
for (var i in cheloveck){
  console.log(i)
```

Задание

Создать объект, описывающий прямоугольник (хранит координаты левой верхней и правой нижней точек), и написать следующие функции для работы с таким объектом.

- 1. Функция принимает объект и выводит инфу о нем.
- 2. Функция принимает объект и возвращает его ширину.
- 3. Функция принимает объект и возвращает его высоту.
- 4. Функция принимает объект и возвращает его площадь.
- 5. Функция принимает объект и возвращает его периметр.

Класс

```
class Rectangle {
  constructor(h, w) {
    this.height = h
    this.width = w
```

Экземпляры класса

```
var p1 = new Rectangle(34,54)
var p2 = new Rectangle(2,23)
var p3 = new Rectangle(41,33)
```

Методы внутри класса

```
class Rectangle {
  constructor(h, w) {
    this.height = h
    this.width = w
  getArea() {
    console.log(this.height * this.width)
```

Обращение через точку

```
var p1 = new Rectangle(34,54)
p1.getArea()
var p2 = new Rectangle(2,23)
p2.getArea()
var p3 = new Rectangle(41,33)
p3.getArea()
```

Задание

Ecть класс Person, со свойствами: имя, фамилия и квалификация специалиста.

Есть метод, который возвращает строку, включающую в себя всю информацию о сотруднике.

Создайте три экземпляра класса Person.

Посмотрите информацию о сотрудниках и увольте самое слабое звено.

ВОПРОСЫ

ПРАКТИКА



Давайте подведем итоги.