**Способы создания объекта**

Мы с вами уже умеем создавать объект собственными руками и получать из его полей какие-то данные.

Напомню, что это делается следующим образом:

const p1 = {

name: 'Igor',

age: 25,

format() {

return `${this.name} (${this.age})`

}

}

console.log(p1.name) //в консоли отобразится 'Igor'

console.log(p1.format()) //в консоли отобразится отформатированный текст 'Igor (25)'

Сегодня мы научимся создавать объекты другими способами.

**Конструктор**

**Конструктор** - функция вызываемая при создании объекта.

Имена конструкторов записываются с заглавной буквы, как и имена классов в JavaScript (не путать с классами в вёрстке), которые мы еще не изучали.

Рассмотрим создание конструкторов на примере:

function Person(name, age) {

this.name = name

this.age = age

this.format = function() {

return `${this.name} (${this.age})`

}

}

const p2 = new Person('Igor', 25)

console.log(p2.format())

Слово **new** означает, что мы создаём пустой объект и устанавливаем его контекстом вызываемой функции, при этом возвращая в конце этот же контекст.

**Object.create()**

**Object.create()** создаёт новый объект на основе указанного в скобках прототипа.

const p1 = {

name: 'Igor',

age: 25,

format() {

return `${this.name} (${this.age})`

}

}

const p3 = Object.create(p1)

p3.name = 'Aybek'

console.log(p3) //отобразит { name: 'Aybek' }

console.log(p3.format()) //отобразит 'Aybek (25)'

Как мы видим, сам объект p3 содержит только поле name: 'Karen', однако мы можем использовать поля из оригинального объекта p1, так как p1 является прототипом p3.

**Метод constructor()**

У каждого объекта есть метод constructor().

С помощью данного метода, мы можем создавать новый объект, на основе другого объекта, который ранее был создан с помощью функции-конструктора.

Рассмотрим на примере:

function Person(name, age) {

this.name = name

this.age = age

this.format = function() {

return `${this.name} (${this.age})`

}

}

const p2 = new Person('Igor', 25)

const p4 = new p2.constructor('Egor', 24)

console.log(p4.format()) //отобразит 'Egor (24)'

console.log(p4.\_\_proto\_\_) //отобразит {}

Интересно то, что у такого объекта не будет прототипа, так как мы содали его с нуля, используя функцию-конструктор объекта p2.

Важно помнить, что создавая объект через метод constructor() мы должны передать данные во все аргументы.