**Функции конструкторов.**

Пишутся с большой буквы. this.name используется как обещание создать будущий объект. С помощью функции new создается новый объект, который имеет параметры основного массива.

function *User*(name, age, status) {  
 this.name = name; // Создаем новый объект  
 this.age = age; // Создаем новый объект  
 this.greeting = function (msg) { // Создаем новый объект  
 console.log(`${msg} my name is +${this.name}`)  
 }  
}  
let user1 = new *User*('vasya', 44, false); // функция **new** создает новый объект  
console.log(user1)

user1.greeting('Hello')

console.log(user1.age) *// Выведет возраст user1*

**Конструктор**

function *Dev*(name, age, skills) { *//* ***Это конструктор Dev****, который создает шаблон с именем, возрастом* this.name = name; *// и скилами. Доп функция skill будет работать если нужно добавить* this.age = age; *// какие то новые скилы.* this.skills = skills;  
 this.addSkill = function (skill) {  
 this.skills.push(skill);  
 }  
}  
  
*Dev*.prototype.addSkill = function (skill) { *// Характеристика функции prototype // нужна если нет доступа до конструктора* this.skills.push(skill)  
}  
  
let dev = new *Dev*('kokos', 123, 'jawa') *// Добавили новый обьект dev с помощью new и Конструктора*console.log(dev)

**Функция call**. **Подмена**

function *Human*(name) {  
 this.name = name;  
 this.greeting = function (msg) {  
 console.log(`${msg} my name is ${this.name}`)  
 }  
}  
let human1 = new *Human*('vasya')  
let friend = {  
 name: 'kokos',  
 age: 123  
}  
human1.greeting.call(friend, ‘Hi’) *// У friend нет функции greeting , а нам //нужно ее  
// вызвать. Для этого используем call для human1. В //кавычках у call  
// пишем кому переназначить функцию greeting. Т.е. для friend. Это подмена*

**Функция apply. Подмена**

human1.greeting.apply(friend, ['Hello', 'By']) *// Функция apply делает то же что // и call, только заполняется массивом*

**Функция bind. Подмена контекста**

let greetingCopy = human1.greeting.bind(friend); *// Функция подмена.*greetingCopy('hahaha');

**Функция Prototype.** Для создания новых библиотек.

let obj = {};   
console.log(obj.\_\_proto\_\_); *// Выводит прототип объекта obj*

**Классы.** Нужно что бы по другому описать функцию конструкторов.