***JavaScript OOP***

OOP => Objcet Oriented Programming.

OOP is A pradigm or style of code.

**Constructor Function**

function User(i, u, s) {

  this.id = i;

  this.username = u;

  this.salary = s;

}

let userOne = new User(105, "Bilal", 6000);

let userTwo = new User(125, "Hassan", 800);

let userThree = new User(128, "anas", 900);

console.log(userOne, userTwo, userThree);

SomeTimes we need a **‘blueprint’ مخطط** for creacting many Objects of the same ‘type’.

The way to create an ‘object type’, is to use an **object constructor function.**

In the exaple above , function User() is an object constructor function .

Objects of the same type are created by calling the constructor function with new keyword.

**Notes :**

It is considered good practice to name construtor funcitons with an upper-case first letter.

The constuctor function is very usefull especially when you have to create many obj.

***About this***

In a constructor function this does not have a value. It is a substitute for the new object. The value of this will become the new object when a new object is created .

**Constructor Function : new syntax**

In Es2015, a new syntax appear :

So for converting code from the old form to new one press on lamp after pressing 3 dots

The code will become like this :

class User {

    constructor(i, u, s) {

        this.id = i;

        this.username = u;

        this.salary = s;

    }

}

**Instance of :**

For checking constructor , return boolean (true or false )

console.log(userOne instanceof User); // true

let literalString = "This is a literal string";

let stringObject = new String("String created with constructor");

literalString instanceof String; // false, string primitive is not a String

stringObject instanceof String; // true

literalString instanceof Object; // false, string primitive is not an Object

stringObject instanceof Object; // true

**constructor:**

The **construtor** property retunrs the function that created Object prototype.

console.log(userOne.constructor);

/\*

class User {

  constructor(i, u, s) {

    this.id = i;

    this.username = u;

    this.salary = s;

  }

}

\*/