**Object Blueprint (Class)**

We can create object blueprint using the classes.

There are two ways of creating a class.

**1.Class declaration**

class class\_name{}

**2.Class Expression**

const class\_name = class {}

Both are valid, you can use any of them.

**Declaring Object Properties**

In normal programming you first declare properties of object inside class but in js you can’t do as you are not able use *let* and *var* inside class this because you can’t create a unknown block inside a class as a result of this variable block scoping won’t work.

class person{

    let name;

    let age;

    let email;

    let mobile;

}

So you have to use **constructor** for that purpose.

Note: You can use *let* and *var* inside the constructor.

class Person{

    constructor(name,age,email,mobile){

        this.name =new String(name);

        this.age = new Number(age);

        this.email=new String(email);

        this.mobile= new Number(mobile);

    }

}

const person1 =

new Person('Vally',21,'vallyross@gmail.com',1234567890);

**Constructor**

The constructor method is a special method of a [class](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/class) for creating and initializing an object instance of that class.

**Note:**

String, Number, Date, etc. are also called **Global Objects** in JS.

**Object Constructor Function(**Alternate way of creating objects**)**

You can also create using object constructor function.

function Person(name,age,email,mobile){

    this.name =new String(name);

    this.age = new Number(age);

    this.email=new String(email);

    this.mobile= new Number(mobile);

}

const person1 = new Person('Vallyross',21,'vallyross@email.com',1234567890);

person1.name

String {'Vallyross'}

This also produce same as of object created using class.

But this approach is less preferred as it creates problem when adding more feature as compare to class.