JS-Basics - II

Objects in JS ?

Muliple variable linked with each other.
Objects has its properties with it's value.

Creation of objects :-

```
const obj_name={
    properties : value,
    properties : value,

    f_name : function(){
        //code
    }
}
```

const → types
f_name → method name

{ } → object literals

When you create function in object it will called method.

Object creation :-

① Factory Function :- it will create object and return it.

```
function creatRectangle(l,b) {        → lenght, breath
    return rectangle = {
        Lenght : 1,
        breath : 2,
        draw() {console.log("Drawing ");
        }
    };
```

② Constructor Function

↳ Use pascal notation → MarksOfSubjects

↳ 'This' keyword represents the current object.

```
function Rectangle()
{
    this.length = 1;
    this.breadth = 2;
    this.draw() {console.log('Drawing');}
}
```

↳ Here no need to return object.

```
let rectangle.obj = new Rectangle();
```

Dynamic nature of objects

You can easily add and remove properties. easily by using ~~new~~ & ~~dete~~ delete

```
rectangleObj.area = rectangleObj.length * rectangleObj.breath;
let rectangleObj = new Rectangle(4,6);
dete delete rectangleObject.color; // deletion
```

Constructor Property :-

Functions is also can object & all objects have constructor.

With [.] dot operator you can use many properties of object.

The constructor data property of an Object returns a reference to the constructor function that created the instance object.

for example `console.log(rectangleObj.constructor);`

Types in JS

Primitive (value types)

copy by their value

it creates copy

- Numbers
- String
- Boolean
- Undefine
- Null

Reference (object types)

copied by their address/reference

same address points by different name.

- Functions
- Objects
- Array

Iteration through objects

For-in

used in objects to check properties is present or not.

```
for (key in objects)
{
    //Block of code
}
```

For-of

used in iterables like array ...

```
for (key of iterables)
{
    //Block of code
}
```

Object Cloning :

Using iteration

```
obj2 = { }
obj = {a:10, b:20, c:30};
for (let key in obj)
{
    console.log
    obj2 [key] = obj [key];
}
```

Using assign

```
let src = {a:10, b:20, c:30};
let dest = Object.assign
           ({ }, src);
Console.log (dest)
```

Using spread

```
let src = {a:10, b:20, c:30};
let dest = { ... src };
console.log (dest);
```

Garbage Collection

It is use to deallocate var / const ~~and switch~~ which is not in use.

We have no control on garbage collection it works automatically