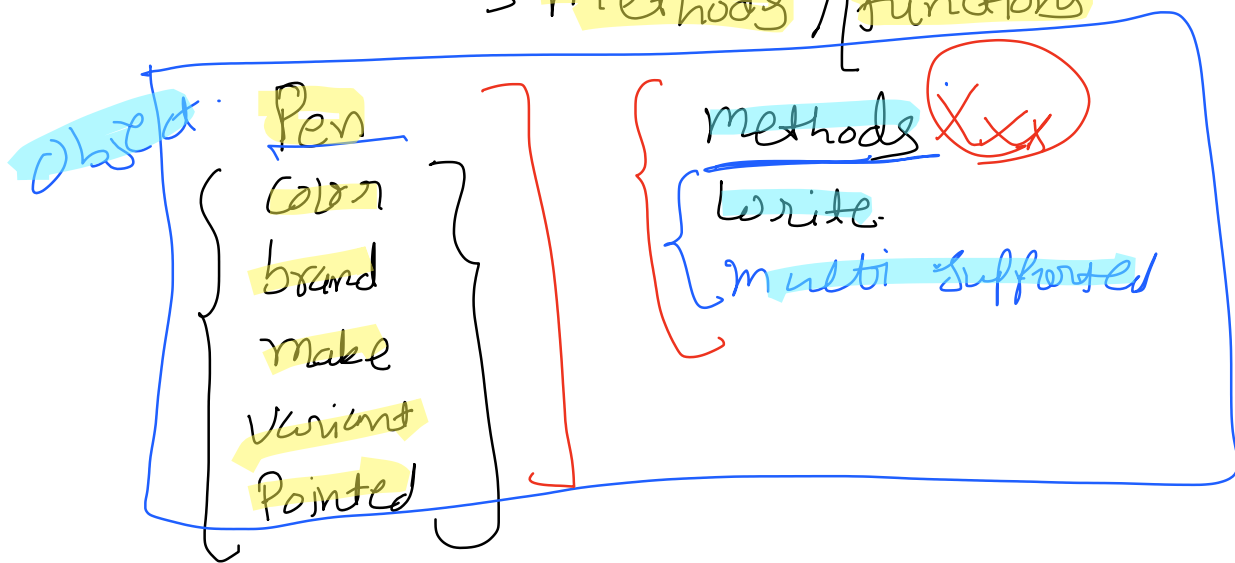


06_08- Array -2

Saturday, 6 August 2022 10:14 AM

What is Object??

Object in JS \rightarrow Property \Rightarrow key-value pair
 \rightarrow methods / functions



key \rightarrow String / Primitive type

Primitive data type

\therefore Default / built in types

string, number, boolean,
undefined, null, Symbol

Non Primitive

Derived

array, object

Value :-

key-value pair

Obj \equiv {
Primitive types {key1: val1}
non-pri {key2: val2}

Obj = {

name : "Jitendra", // string

age : 23, // number

address : {
"city" : "Hyd", // obj
"country" : "India"

Siblings : ["Rahul", "Sudhanshu"]
 // array

Comp :-

Primitive

Motherboard, keyboard, ram, screen,
 body, charger, ports, hd // built in

derived { extend num, cover, webcam, ext hdd, mouse, }

2) Constructal way:- new Array()

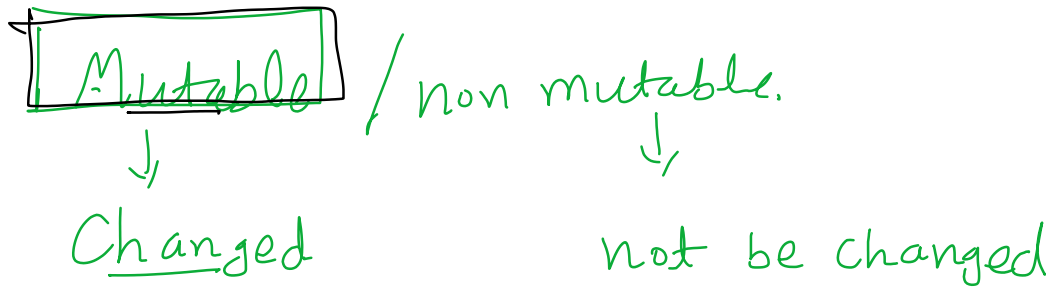
- constructor is a special kind of fun, which is invoked when you try to create obj using new keyword.

JSON

```
37
38
39 var profile = {
40   name: "Srihari",
41   address: {
42     city: "HYD"
43   }
44 }
45
46 console.log(profile)
47
48 profile.name = "Sudhanshu";
49
50 console.log(profile)
```

Terminal Output:

```
nm: 23,
age: 'Abhishek',
siblings: [ 'Rahul', 'Sudhakar' ],
isStudent: true,
address: { city: 'HYD', country: 'IND' }
}
prakashshah@192 Code % node Arrays-2.js
{ name: 'Srihari', address: { city: 'HYD' } }
{ name: 'Sudhanshu', address: { city: 'HYD' } }
prakashshah@192 Code %
```

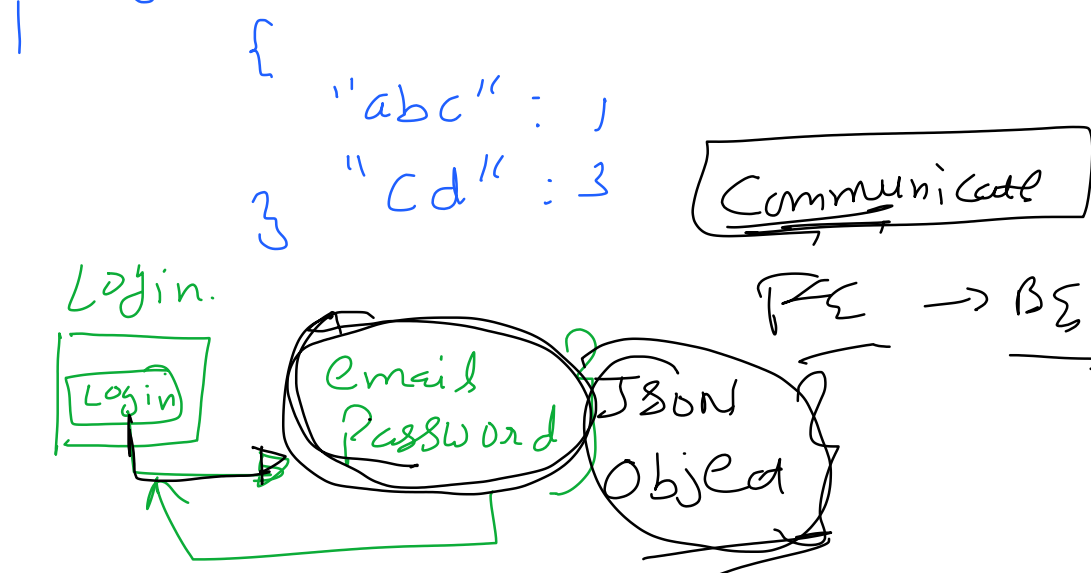


Non-Primitive

Object/array

JSON ⇒ Javascript object notation

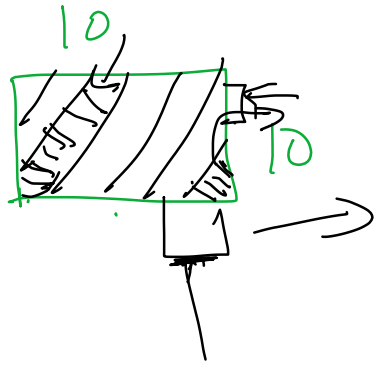
obj ⇒ key & value string.



React ↔ Java

2 · Asked by Jitendra Vaishnav

```
obj={
  name:"j",
  confidential:{
    salary: 50,}}
obj.name="jk"
Object.freeze(obj.confidential)
obj.confidential.salary=60
console.log(obj.confidential.salary)
```



⇒ 100824

Extensible

1) Map:- used to transform array

[5, 1, 3, 2, 6]



[10, 2, 6, 4, 12]

arr = [10, 2, 6, 4, 12]

// double

[10, 2, 6, 4, 12]

```
// IMPPPPPPPPPPP
// 1. Map method :
let arr = [5, 1, 2, 3, 6]; // [10, 2, 4, 6, 12]
arr.map()
```

arr.map (new fn)

```
function double(x) {
  return x*2;
}
```

```
let arr = [5, 1, 2, 3, 6]; // [10, 2, 4, 6, 12]
function double(x) {
  return x*2;
}
```

```

let output = arr.map(double);
console.log(output);
console.log(arr);

```

② Arrow funⁿ

Arguments \Rightarrow logic

$(x) \Rightarrow \{$
 $\quad \text{return } x * 2;$
 $\quad \}$

2. filter:-

arr.filter ()

arr = [5, 1, 3, 2, 6]

```

var output = arr.filter(getEvenValues);
var output1 = arr.filter(function getEvenValues(x){
  return x%2==0;
});
var output2 = arr.filter((x) => {
  return x%2==0;
});
console.log(output, output1, output2);
console.log(arr);

```

[2, 6]

$5 \div 2 \neq 0 \Rightarrow \text{false}$

$1 \div 2 \neq 0$

$2 \div 2 = 0$

$3 \div 2 \neq 0$

$6 \div 2 = 0$

$x \div 2 == 0 \ \&\& \ x >= 0$