Objects in Java script

22 November 2023 00:19

Object Declare

- Literal
- Constructor.

Constructor se jab object creat krege to vo singleton object bnega.

Literals

Const objectName = {}

⇒ Eg. const JsUser = {

name : "Ankit"

}

→ by default computer iss ko string ki tarah leta

Access

1.- Console.log (JsUser.name)
2. Console.log (JsUser["name"])

const Myobj {

"name": "Ankit";

}

Now you can't access using Myobj.name

Can access only Myobj["name"]

Use key as a Symbol.

```
const mySymbol = Symbol("myName");
const myObj = {
    mySymbol : "Ankit Narang"
}
console.log(myObj.mySymbol)
console.log(typeof(myObj.mySymbol))
```

→ string

Ankit Narang

=> Yha pr humne Symbol ko as a key define kia hai.
But yeh glt tarika hai.

Yeh Code Run to shi hoga but agar hum iska type print krege yeh ab bhi "string" dega

Output

Ankit Narang
string

Ankit Narang

=> Shi tarika kya hai Symbol ko as a key declare krne ka?

=>

```
const mySymbol = Symbol("myName");
const myObj = {
    [mySymbol] : "Ankit Narang"
}
console.log(myObj[mySymbol])
console.log(typeof( myObj[mySymbol] ))
```

→ yeh shi tarika hai

Ankit Narang

=> [ ] bracket mai likhke hum Symbol use kr sakte hai.

=> Object mai Value ko Change krna.

```
const myObj = {
    mySymbol : "Ankit Narang"
}
```

Ankit Narang

→ Aise hum keys ko chan

```
console.log(myObj.mySymbol)
myObj.mySymbol="This is change value"
// IF WE DON'T WANT TO CHANGE VALUE OF KEY IN
FUTURE
// SIMPLY USE FREEZE FUNCTION
Object.freeze(myObj);
myObj.mySymbol="Try to change after freeze"
console.log(myObj.mySymbol)
```

ki value ... kr sakte hai

Object.freeze() s... hum restrict krd... hai ki uski key ki value change ... ho.

Output.

Ankit Narang
This is change value

Ankit Narang

Using function in an object.

```
const myObj = {
    mySymbol : "Ankit Narang"
}
myObj.greeting= function(){
    console.log(`Hello ${this.mySymbol}`)
}
myObj.greeting()
```

greeting fun" object mai ... ho gya.

Ankit Narang

this ki help se hum object (My... ke saare member access kr sakte ho

=> ${ }

↳ act as a placeholder.

☞ Object as a Singleton or Create object usi... Construct...

```
const myObj = new Object()
myObj.id=2
myObj.name="Ankit"
console.log(myObj)
```

Ankit Narang

Nesting of a object.

```
const myObj = {
    name :{
        fullname:{
            firstname:"Ankit",
            Lastname:"Narang"
        }
```

Ankit Narang

```
    }
}
console.log(myObj.name)
console.log(myObj.name.fullname)
console.log(myObj.name.fullname.firstname)
```

Output :

{ fullname: { firstname: 'Ankit', Lastname: 'Narang' } }
{ firstname: 'Ankit', Lastname: 'Narang' }
Ankit

Ankit Narang

Merge Two objects

```
const myObj1 = {1:'a',2:'b',3:'c'}
const myObj2 = {1:'a',2:'b',3:'c'}
const myObj3= {1:'a',2:'b',3:'c'}
const mergeObj =
Object.assign({},myObj1,myObj2,myObj3)
console.log(mergeObj)
```

Ankit Narang

Object.assign(targ
=> target = { } -> Empt
obje
=> Sources = myObj1
myObj2
myObj

{} => optional or we can Make any obj as a target

Output :

{ '1': 'a', '2': 'b', '3': 'c' }

Ankit Narang

```
const myObj1 = {1:'a',2:'b',3:'c'}
const myObj2 = {3:'a',4:'b',5:'c'}
const myObj3= {9:'a',7:'b',6:'c'}
const mergeObj =
Object.assign({},myObj1,myObj2,myObj3)
console.log(mergeObj)
```

Ankit Narang

OUTPUT

{
'1': 'a',
'2': 'b',
'3': 'a',
'4': 'b',
'5': 'c',
'6': 'c',
'7': 'b',
'9': 'a'

Ankit Narang

∴ It Merge only Unique Values.

}

Another Way to Merge (using Spread operator (...))

```
const myObj1 = {1:'a',2:'b',3:'c'}
const myObj2 = {3:'a',4:'b',5:'c'}
const mergeObj= {...myObj1,...myObj2}
console.log(mergeObj)
```

Ankit Narang

→ (...) spread operator.

⇒ Access keys & values of an object seperately (object.keys

```
const myObj1 = {1:'a',2:'b',3:'c'}
console.log(Object.keys(myObj1))
console.log(Object.values(myObj1))
```

Ankit Narang

Output

```
[ '1', '2', '3' ]
[ 'a', 'b', 'c' ]
```

Ankit Narang

→ It gives result in array. So we ca traverse them using loop.

Also there is -> Object.entries (Name of object)
Give key value pair in a array.

```
const myObj1 = {1:'a',2:'b',3:'c'}
console.log(Object.entries(myObj1))
```

Ankit Narang

Output :

```
[ [ '1', 'a' ], [ '2', 'b' ], [ '3', 'c' ] ]
```

Ankit Narang

To check some property or key exist in an object.

⇒ use Myobj.hasOwnProperty ('name')
→ True.

```
const myObj1 = {1:'a',2:'b',3:'c'}
console.log(myObj1.hasOwnProperty('1'))
```

Ankit Narang

Output :

```
true
```

Ankit Narang

Destructing in an Objects

```
const myObj1 = {
    course : "B.Tech",
    name : "Ankit",
    age : 21
}
console.log(myObj1.course)
```

Ankit Narang

Till now we see this Method.

Output :

B.Tech

Ankit Narang

Now Another way of Destructing.

```
const myObj1 = {
    course : "B.Tech",
    name : "Ankit",
    age : 21
}
const{course} = myObj1
console.log(course)
```

Ankit Narang

→ Iska Matlb myobj se course ki value lelo. So ab hum directly course se isk value print krdge.

OUTPUT :

B.Tech

Ankit Narang

→ Aliasing

```
const{course : c} = myObj1
console.log(c)
```

Ankit Narang