Java Script Day 7

17 January 2024 14:05

function has 2 properties → functions inside object = method → of object
↓ ↓ datatype inside objects = property → of object.
functions objects.

new Keyword → Super imp. → perform 3 things

- Create an empty object
- initialize this to empty object
- establish a _proto_ connection to the prototype
- of the marked function

```
Const User1 = new GetUser ("Rit", "Bio", 100)

function GetUser ( name, class, marks) {
    this.name = name
    this.class = class
    this.marks = marks }

GetUser.prototype.forgetName = function ( ){ return (this.name) }

Console.log( User1)  → {"Rit", "Bio", 100 }
Console.log( User1.forgetName ()) → Rit

Console.log (User1.Keys ()) → [ name, class, marks, forgetname ]
```

→ all Keys including prototype object.

Constructor function
↓
identity → (Function name will be always starts with Capital letter)

Solving _Proto_ & Prototype Complexity → New Keyword

```
Class ClassName {

    Constructor (name, course, marks) {
        this.name = name
        this.course = course
        this.marks = marks }

    gettingInfo () {
        return ` ${this.name} and ${this.class} ` }
}

User1 = new ClassName ('Ritesh', 'Biology', 100)
Console.log (User1.name)
Console.log (User1.gettingInfo())
```

imp ↑

new → this refrence
Constructor → for dtypes
methods → for functions.

at Class]

```
Set NewObject (name) {
Const [ name, class, marks ] = name.
this.name = name
this.class = class
this.marks = marks }
```

Or → SetNames (name, class){
this.name = name
this.class = class
}

Outside
↓

Inheritance: → using extend Keyword You can

Inheritance: → using extend Keyword you can get all the properties & methods of the parent class → class Cname {

constructor (11) {

"

}

fun1() { " }

fun2() { " }

}

super → for getting properties of previous constructors to update the newer class constructor

extending

class chclass extends Cname {

Constructor (extra, previousClass Constructors) {

super (previous class constructors)

this.extra = extra }

const User = new chclass (11)

Console.log(User.fun1())

this.class ...

this.marks = marks }

Class, new object

= "Riks Second 100"

Getter & Setter

Using get Keyword to convert any method of class into property

eg

class A {

constructor (name) {

this.name = name}

get yourName () { return ~ Name is ${this.name}

const info = new A("Mohit")

Console.log(info.yourName) → its now a property not a method

Static Method & functions

Will only be called with the name of class not with the object name.

→ at Class

static id = 5000;

static information () { return ~ Ms Explorer ~ }

outside

Console.log(ClassName.id)

Console.log(ClassName.information ())

[ClassName is a class not a object]