

* ~~Prototypes~~ → Prototypes are the mechanism by which JavaScript objects inherit features from one another.

→ While using Constructor functions we saw that there is something called " __proto__ : Object" inside the object we created.

-1 for every object we create it will be having " __proto__ " Property.

Ex- Person.prototype.workingBio = function(
experience, age)

console.log(this.name + " " + this.experience
+ " " + this.age);

3

So, → the "--Proto--" Property will point towards the constructor function with prototype property.

--Proto-- → Person.prototype.
↓
(Constructor
function)

Note → So, all the objects created using the "Person" constructor function will be having "workingBio" as a property in their "--proto--" object.

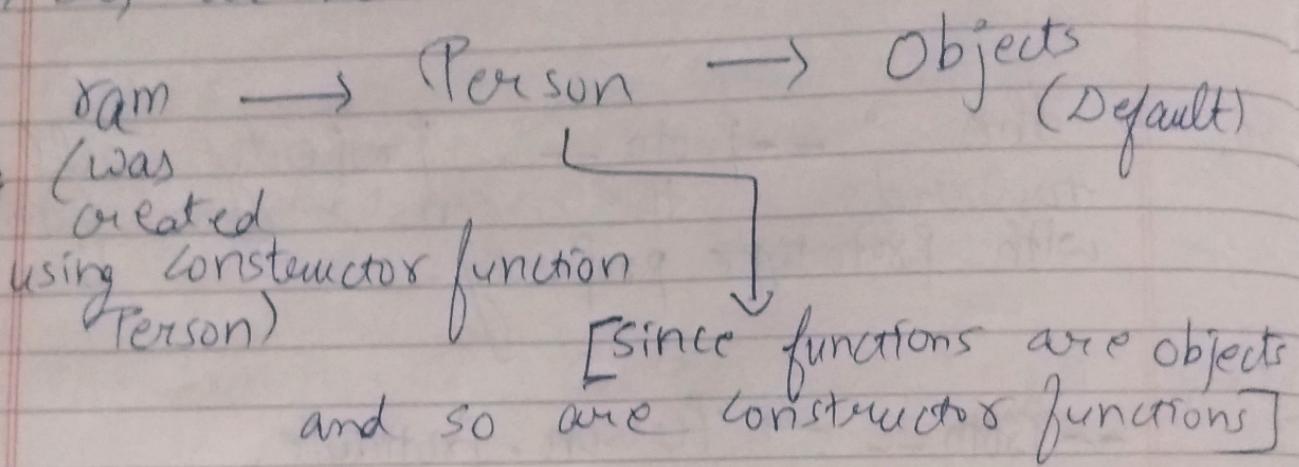
So, we can access that function by simply →

→ ram.workingBio(6, 29);
→ sita.workingBio(3, 21);

But this not we want for the function to be copied again and again.]

So, let's understand the '`--proto--`' function of constructor function which is their inside the objects '`-proto--`' function.

→ So, we have



So, what so ever Properties we set in 'Object.prototype' will come under '`--proto--`' property of the Person

and what so ever Properties we set using "Person.prototype" will come under

"Person.prototype" will come under '`--proto--`' property of the ram object, or any object we created from Person (constructor function).

"This is prototype chaining happening here"

→ So, anything inside 'object.prototype' can be accessed by 'ram', 'sita' or any object we created using constructor function ("Person").

* Object.create() →

→ By using this method, we will see that how can we create new objects.

Ex → var ram = Object.create();
Console.log(ram); ↴ Empty object

O/P → {}
 ► --proto-- : Object
 ↳ // ram.__proto__ = Object.prototype
 (Since there are no parameters passed)

* To add Properties →

ram.name = "Ram";
ram.profession = "Professor";

Console.log(ram);

→ `ram. WORKExperience = function () {
 console.log ("Ram is working");
}`

* But when we do,

→ `var ram = Object.create(null);
console.log(ram);`

Output → {}

→ no properties.

→ It won't be having
--proto--: Object.

→ We also use `Object.create()` if
we want to inherit the properties
of another object, but can't
change the inherited property, but
can create new properties for
the object which has inherited
the properties.

— X —