**Report about Inheritance in function constructor**

**Created by: William George William Botros**

**Track: PHP-ITP Alex Q2, 2023/2024-Intake3**

Constructor functions define the prototype of the properties an object will contain.

Using the constructor function, we can create a new object after passing the required parameters.

Inheriting a previously defined constructor function means using the parameters of the previously defined function along with adding some new parameters to the newly defined constructor function.

For this, we need to use the [call()](https://www.geeksforgeeks.org/javascript-function-prototype-call-method/) function which allows us to call a function defined somewhere else in the current context.

**Syntax:**

|  |
| --- |
| myFunction.call( this, property1, property2, ... , propertyN ) |

**Example**:

 Here, we create an ‘Employee’ constructor function.

A new ‘Developer’ constructor function is created that will inherit the basic properties of ‘Employee’ as well as will contain some new properties.

|  |
| --- |
| function Employee(name, age, gender, id) {  this.name = name;  this.age = age;  this.gender = gender;  this.id = id;  };  function Developer(name, age, gender, id,  specialization) {  // Calling Employee constructor function  Employee.call(this, name, age, gender, id);  // Adding a new parameter  this.specialization = specialization;  }  console.log(Employee.prototype);  console.log(Developer.prototype); |

**Output:**

|  |
| --- |
| Object  constructor: ƒ Employee(name, age, gender, id)  [[Prototype]]: Object  Object  constructor: ƒ Developer(name, age, gender, id, specialization)  [[Prototype]]: Object |

We can notice that the ‘Developer‘ constructor function inherits the properties of the ‘Employee‘ constructor function along with a new parameter ‘specialization‘.

Here, we called the Employee function using the call() function to pass the required parameters to the Employee constructor function.

We can also create objects using these constructor functions after passing the values of the required properties of that object.

**Syntax:**

|  |
| --- |
| let Obj1 = new Object( parameters ) |

**Example:**

This example describes the object created with a [new](https://www.geeksforgeeks.org/javascript-new-keyword/)keyword to create an instance of an object that has a constructor function.

|  |
| --- |
| function Employee(name, age, gender, id) {  this.name = name;  this.age = age;  this.gender = gender;  this.id = id;  };  function Developer(name, age, gender, id, specialization) {  // Calling Employee constructor function  Employee.call(this, name, age, gender, id);  // Adding a new parameter  this.specialization = specialization;  }  // Creating objects  let Employee1 = new Employee("Ahmed", 28, "Male", 564);  let Developer1 = new Developer("Mohamed", 31, "Male", 345,  "Frontend Developer");  console.log(Employee1);  console.log(Developer1); |

**Output:**

|  |
| --- |
| Employee {name: 'Ahmed', age: 28, gender: 'Male', id: 564}  age: 28  gender: "Male"  id: 564  name: "Ahmed"  [[Prototype]]: Object  Developer {name: 'Mohamed', age: 31, gender: 'Male', id: 345,  specialization: 'Frontend Developer'}  age: 31  gender: "Male"  id: 345  name: "Mohamed"  specialization: "Frontend Developer"  [[Prototype]]: Object |

We can observe that the constructor function of Employee is inherited to create a new constructor function Developer which can be used to create objects with new properties along with the inherited properties of the parent constructor.