Name: Krish Mehta

Topic: Class and Inheritance in JavaScript

Class Inheritance

To create a class inheritance, use the extends keyword.

A class created with a class inheritance inherits all the methods from another class:

Example

Create a class named “Model” which will inherit the methods from the “Car” class:

```
Class Car {
  Constructor(brand) {
    This.carname = brand;
  }
  Present() {
    Return ‘I have a ‘ + this.carname;
  }
}

Class Model extends Car {
```

```
  Constructor(brand, mod) {
    Super(brand);
    This.model = mod;
  }
  Show() {
    Return this.present() + ‘, it is a ‘ + this.model;
  }
}
```

```
Let myCar = new Model(“Ford”, “Mustang”);
Document.getElementById(“demo”).innerHTML = myCar.show();
```

The super() method refers to the parent class.

By calling the super() method in the constructor method, we call the parent’s constructor method and gets access to the parent’s properties and methods.

Example

You can use the underscore character to separate the getter/setter from the actual property:

```
Class Car {
  Constructor(brand) {
    This._carname = brand;
  }
  Get carname() {
    Return this._carname;
  }
  Set carname(x) {
    This._carname = x;
  }
}

Const myCar = new Car(“Ford”);

Document.getElementById(“demo”).innerHTML = myCar.carname;
```

To use a setter, use the same syntax as when you set a property value, without parentheses:

Example

Use a setter to change the carname to “Volvo”:

```
Class Car {
  Constructor(brand) {
    This._carname = brand;
  }
  Get carname() {
    Return this._carname;
  }
  Set carname(x) {
    This._carname = x;
  }
}

Const myCar = new Car(“Ford”);
myCar.carname = “Volvo”;
document.getElementById(“demo”).innerHTML = myCar.carname;
```

Hoisting

Unlike functions, and other JavaScript declarations, class declarations are not hoisted.

That means that you must declare a class before you can use it:

Example

```
//You cannot use the class yet.
//myCar = new Car(“Ford”) will raise an error.

Class Car {
  Constructor(brand) {
    This.carname = brand;
  }
}

//Now you can use the class:
Const myCar = new Car(“Ford”)
```