Objects or Constructor Functions?

This lesson discusses why constructor functions are used in JavaScript.

We'll cover the following

Functions as Objects

As discussed previously, functions are also objects in JavaScript. This is because, just like objects, they have their own properties and methods. Functions can also be used to construct objects; these type of functions are known as **constructor functions**.

Why Use Constructor Functions?

Let's answer this question by rewinding to the last chapter where we discussed *object literals*. In order to create an `employee` of a company, we created an object like this:

```
//creating an object named employee

var employee1 = {
  //defining properties of the object
  //setting data values
  name : 'Joe',
  age : 28,
  designation : 'Developer',
  //function to display name of the employee
  displayName() {
    console.log("Name is Joe")
  }
}

//displaying the properties of the object
//the method to access properties will be discussed in detail the next lesson
employee1.displayName()
console.log("Age is:",employee1.age)
console.log("Designation is:",employee1.designation)
```

Now, what if you wanted to create another employee?

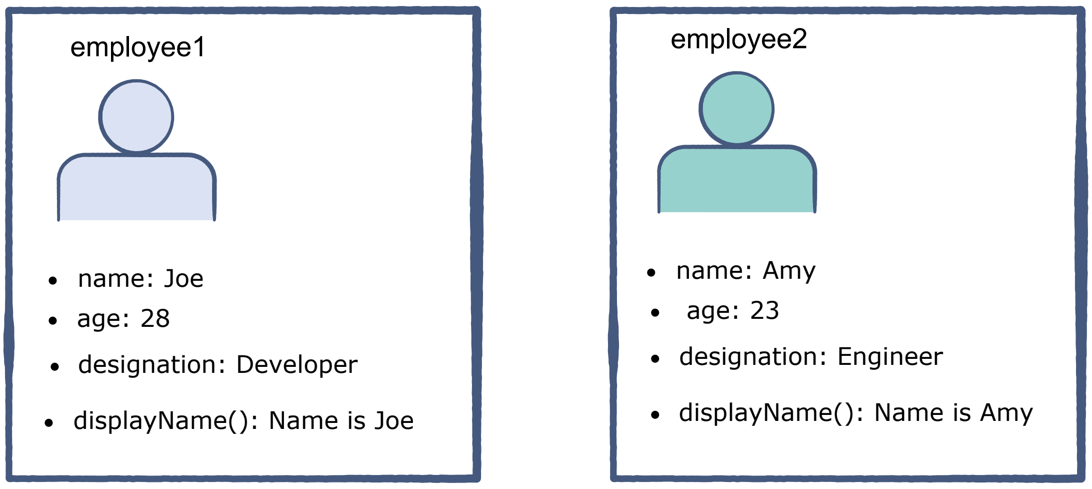
Using the above approach, we would write a code similar to the one shown below:

```
//creating an object named employee2

var employee2 = {
  //defining properties of the object
  //setting data values
  name : 'Amy',
  age : 23,
  designation : 'Engineer',
  //function to display name of employee2
  displayName() {
    console.log("Name is Amy")
  }
}

//displaying the properties of the object
//the method to access properties will be discussed in detail the next lesson
employee2.displayName()
console.log("Age is:",employee2.age)
console.log("Designation is:",employee2.designation)
```

This time we named the employee `employee2` since `employee1` is already taken.



Two employee objects with their properties

What if there are **100** or **1000** employees in the company? Creating separate object literals for each is a tiring and a cumbersome task. Another thing to note is that both `employee1` and `employee2` have all the properties in common; the difference lies only in their object names and property values.

This brings us to the question: *Is there a better approach for doing this?*

Yes, there is. This is where *constructor functions* come into play.

Now that you are clear on why we need constructor functions, let's discuss them in detail in the next lesson.