**JAVA SCRIPT**

IIFE CONSTRUCT (Pronounced as “iify”):Immediatelu-Invoked Function Expression

JavaScript does not know how much memory can be used by a global variable, which results in wastage of memory and also perform issue of the website. So, to solve this problem, we use IFFY CONSTRUCT.

In iify construct we define the anonymous function and then we call the anonymous function immediately. By this we can resuce the problem of memory leaks.

Syntax:

(function(){

………

}();

NOTE: Always have only one iffy construct per js file.

\*\*When you try to add a string and a number we are going to get an output of **NaN**(not-a-number).

**OOPS (Object Oriented Programming Structure)**

There are many data types in JS and one of the special one is object. Objects gives us power to write our own structured variable/datatype.

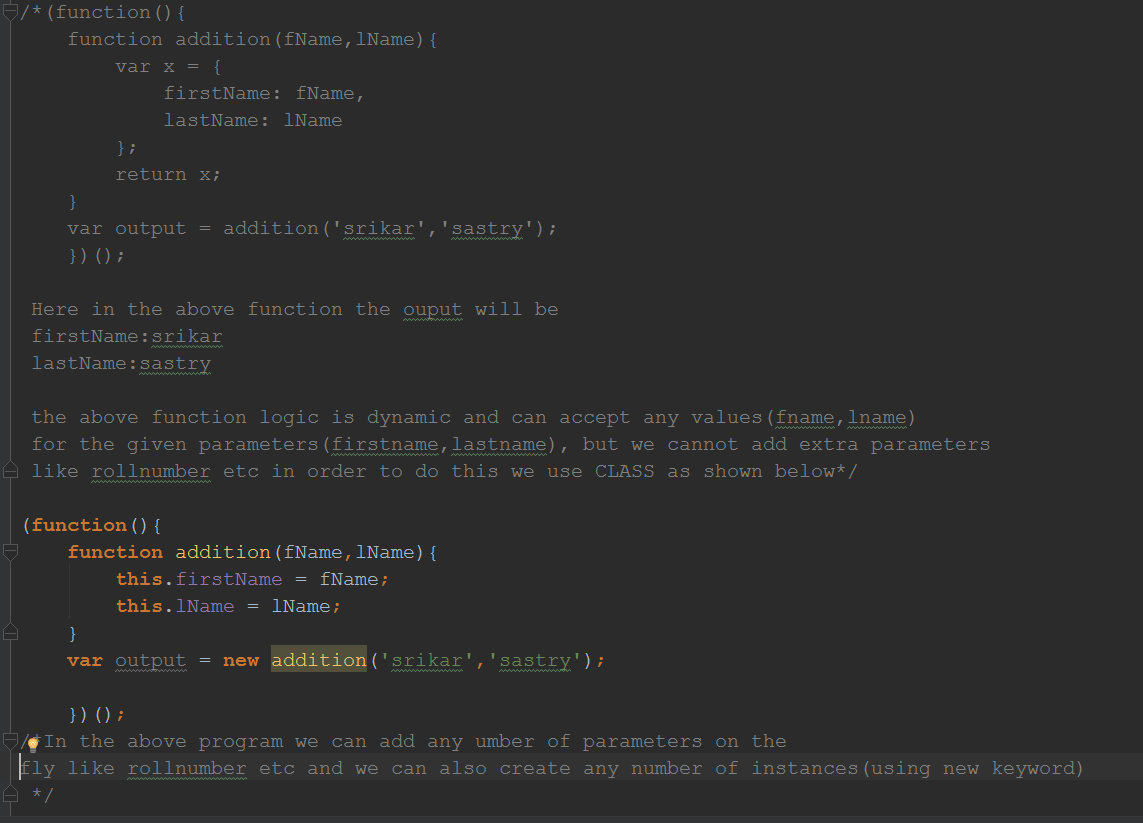
By using objects we can share the data from one machine to other very easily. This sharing of the data is called JSON(JavaScript object notation), these are very light weight objects that can be shared between two machines.

**CLASS**

In functions, we are performing the logic before the return statement (generating output), but in objects the variables are fixed(hardcoded). The only way to change them is after coming out of the object and passing **object.vaiable name= new variable.** But in case the server is giving different values depending upon the database data, then it will be problem because we always cannot write the dot notation to change, this is the problem when we deal with only objects alone. To avoid this problem, we have an option of converting this functions into objects. This is when CLASS comes into the picture.

Objects have fixed values and properties when we define them, So when we want to construct a dynamically changing object, we end up using functions.

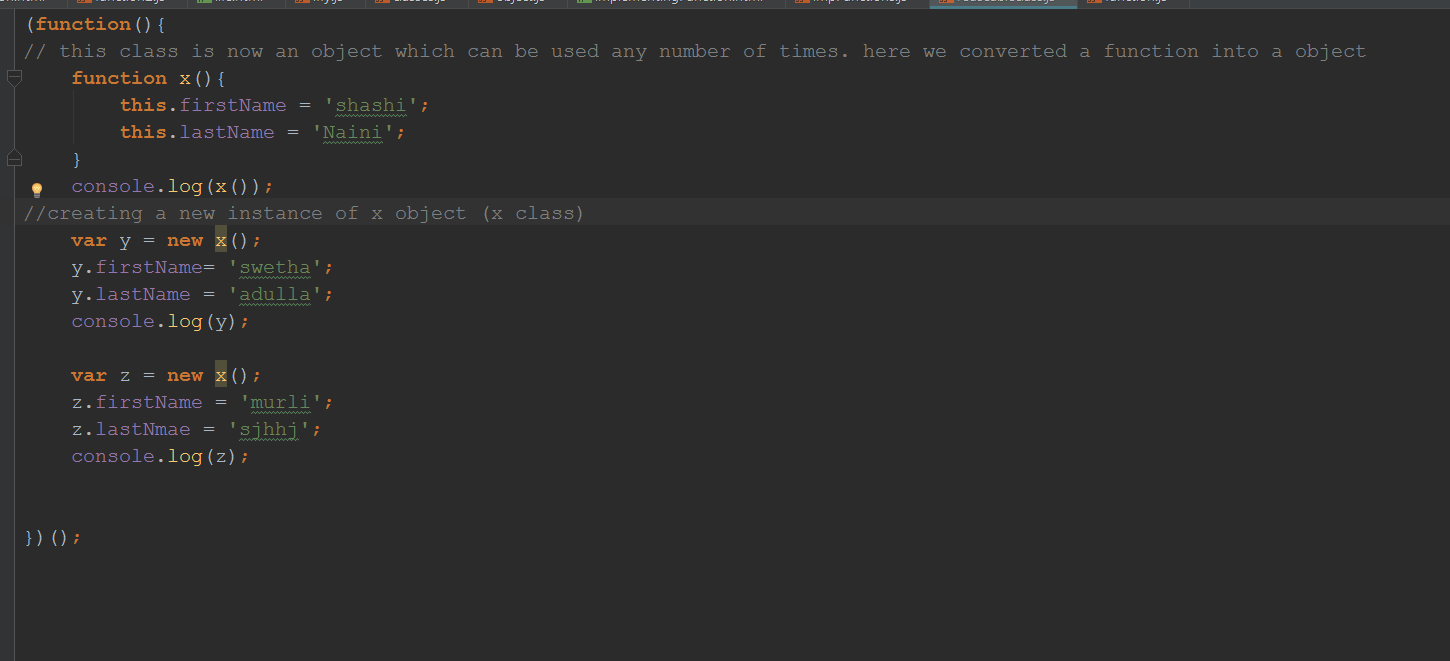
**CLASSES are nothing but functions converted into objects.**

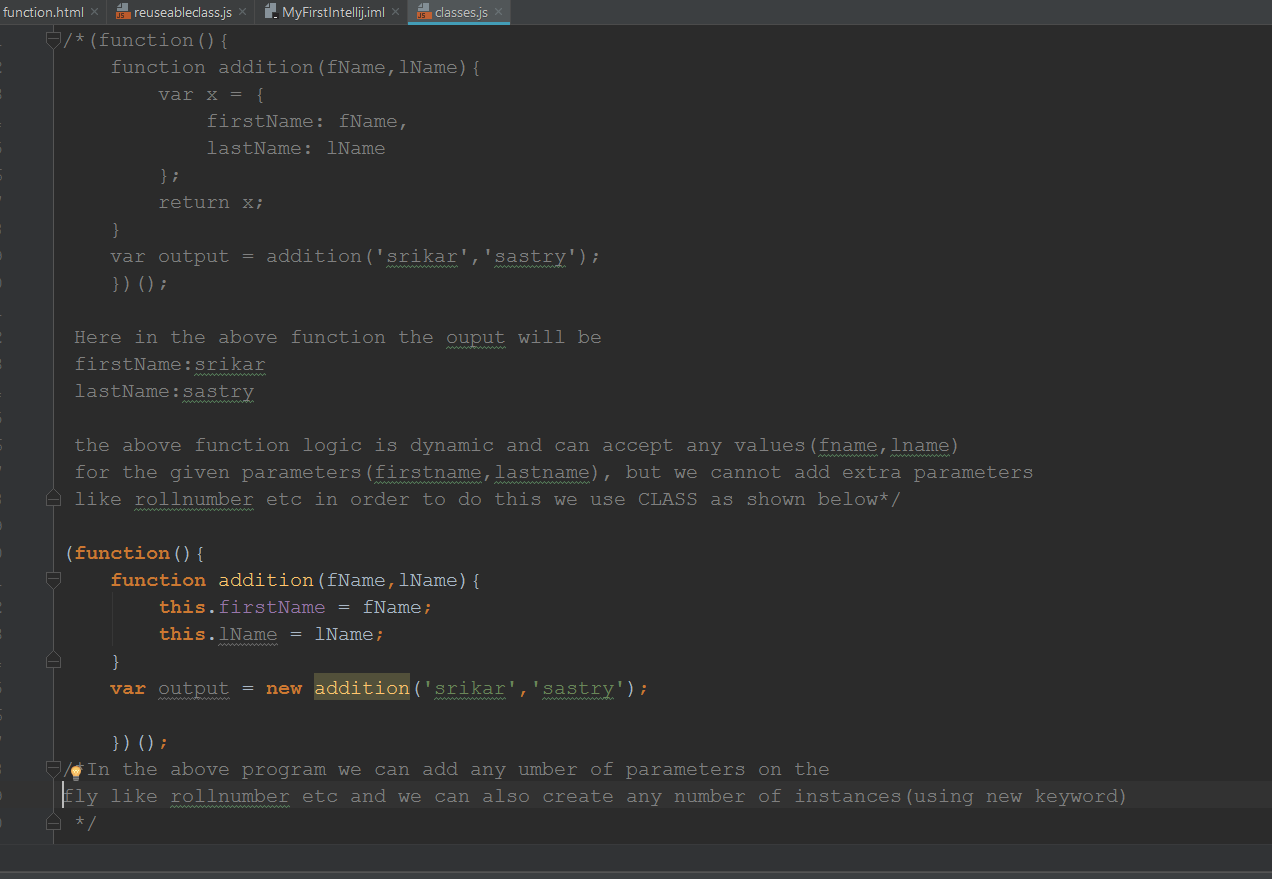


JavaScript language adds a proto by default in the console because to reuse the object anytime. Proto means a prototype, we want to assume the object as a prototype. When it comes to classes we don’t return a statement.

**What is the difference between a function and a class?**

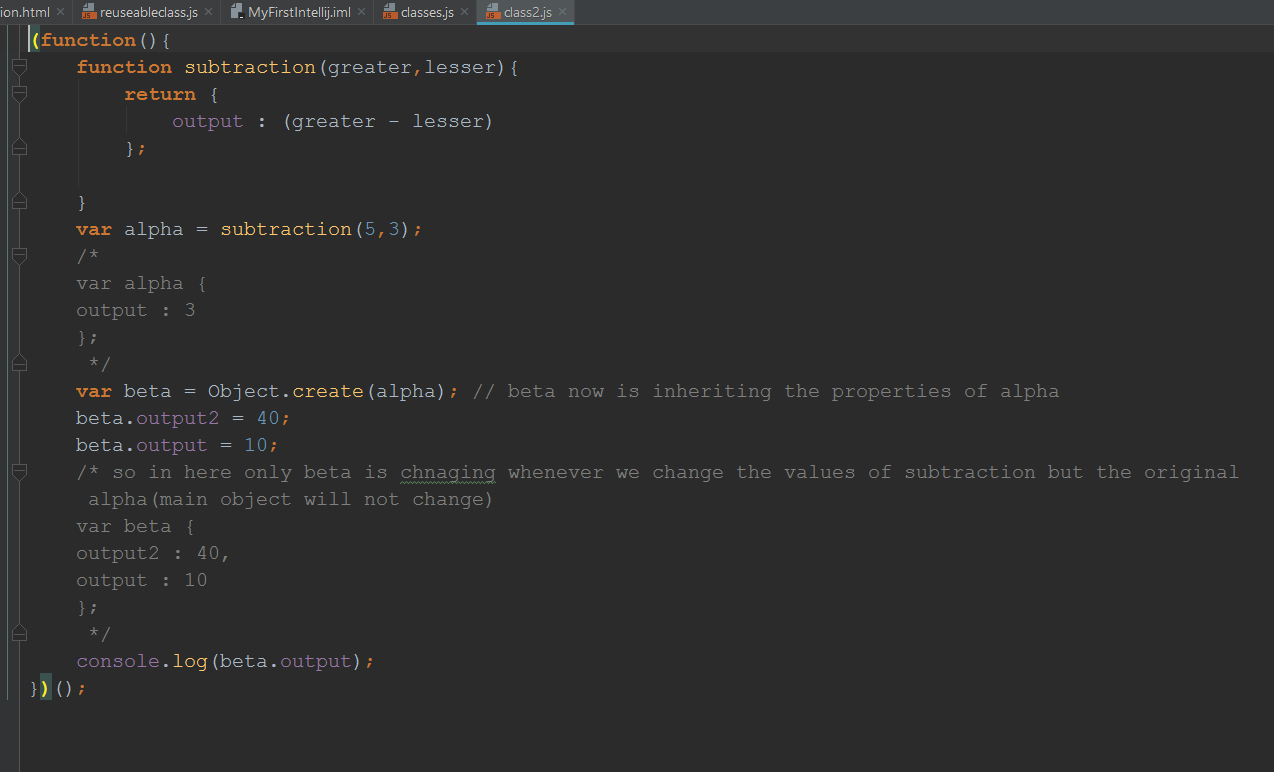
A function is a traditional function which returns a value, a class is a function which is now converted into an object. To convert a function into an object we use a keyword called **this.** This entire concept of creating the instances and inheriting the properties is called **prototypical inheritance.**



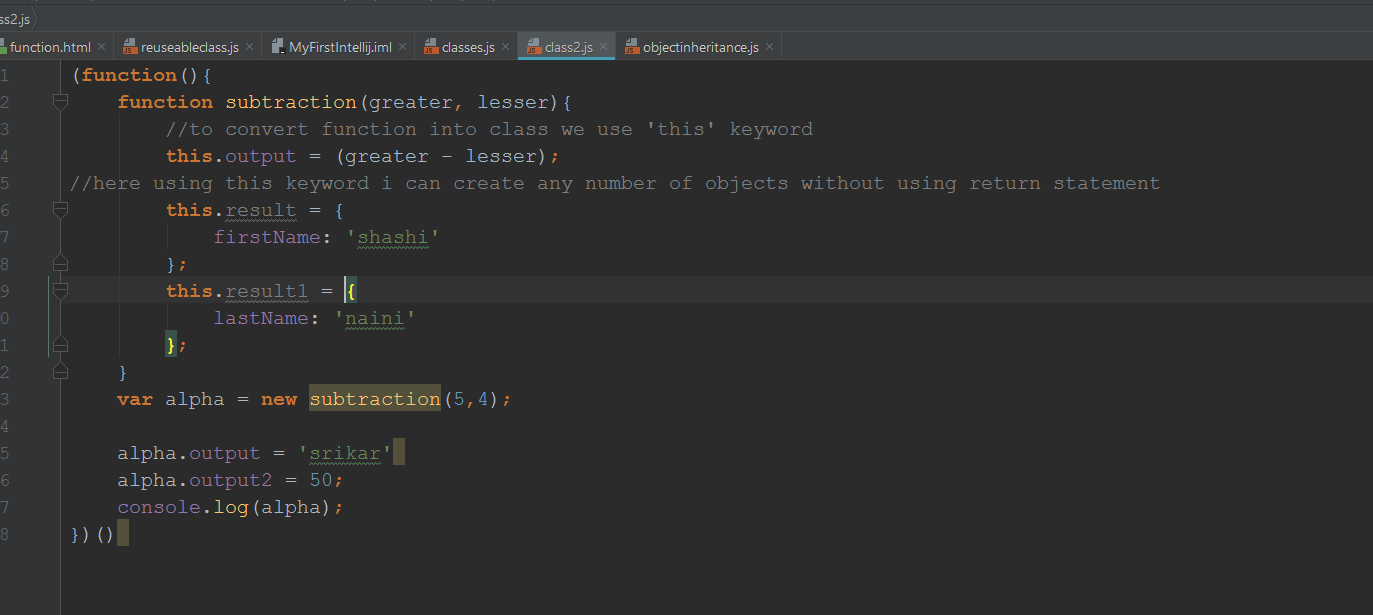


In ECMA script5 we don’t have a concept of classes. We cannot create a class from scratch we can only convert a function into class.

The proto in the console output helps us to create the instances of that object. Whenever we say **Object.create(originalobjectname) the** constructor is getting called and inside that we have create which is creating the instance of the original object. Constructor need not to be called separately whenever we say the key word **object** it is automatically calling constructor.



When we use the return statement in object inheritance it needs to be assigned to a variable for sure, this is the big drawback. I don’t want to return the value but I want to give power to inherit the object. Every function will have only one return statement, what if I have some other logic or if I have one more object to be return after the object, I cannot have another return statement. So, this way of inheriting is not preferred. To overcome this kind of issue the concept of **Classes was introduced.** **Here whatever objects I create in the function thet become the properties of the converted function which is called CLASS. So to convert a function into a class we use ‘this’ keyword.**  Whenever we are using keyword class we are only creating an instance not a bridge.



**What is a class in ECMA5/ JS?**

JavaScript will not have a keyword class but we can convert a function into class by using ‘this’ keyword. All the objects of that function becomes the properties if that class/inherited function. Then we inherit the properties by using below syntax

**Var x = new functionName();**

**Differences between class and object?**

