**JAVASCRIPT Basics**

**Object in JS**

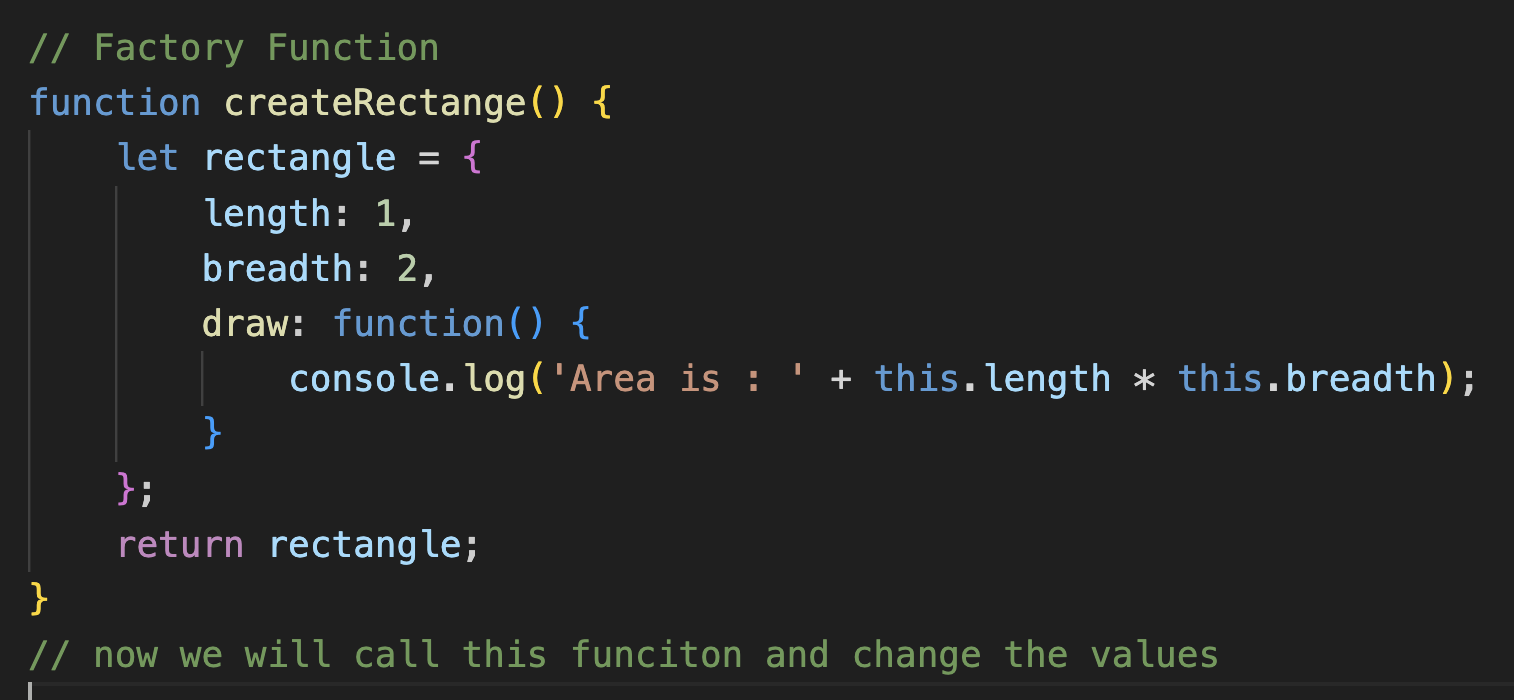
* Encapsulating the properties and function(behavior) in a single entity or multiple linked variables can be stored in a single entity can be stored in Object.
* Example:
  + const rectangle = {

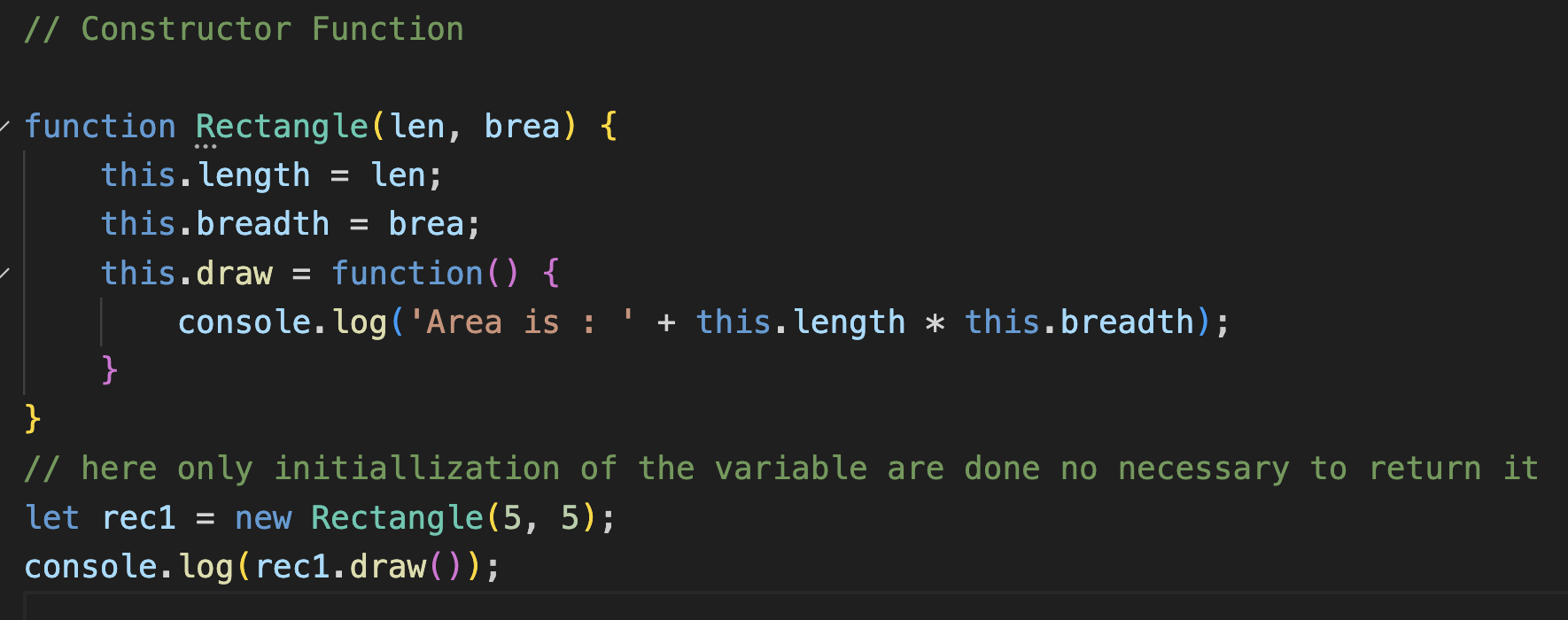
length: 1,

breadth: 2

};



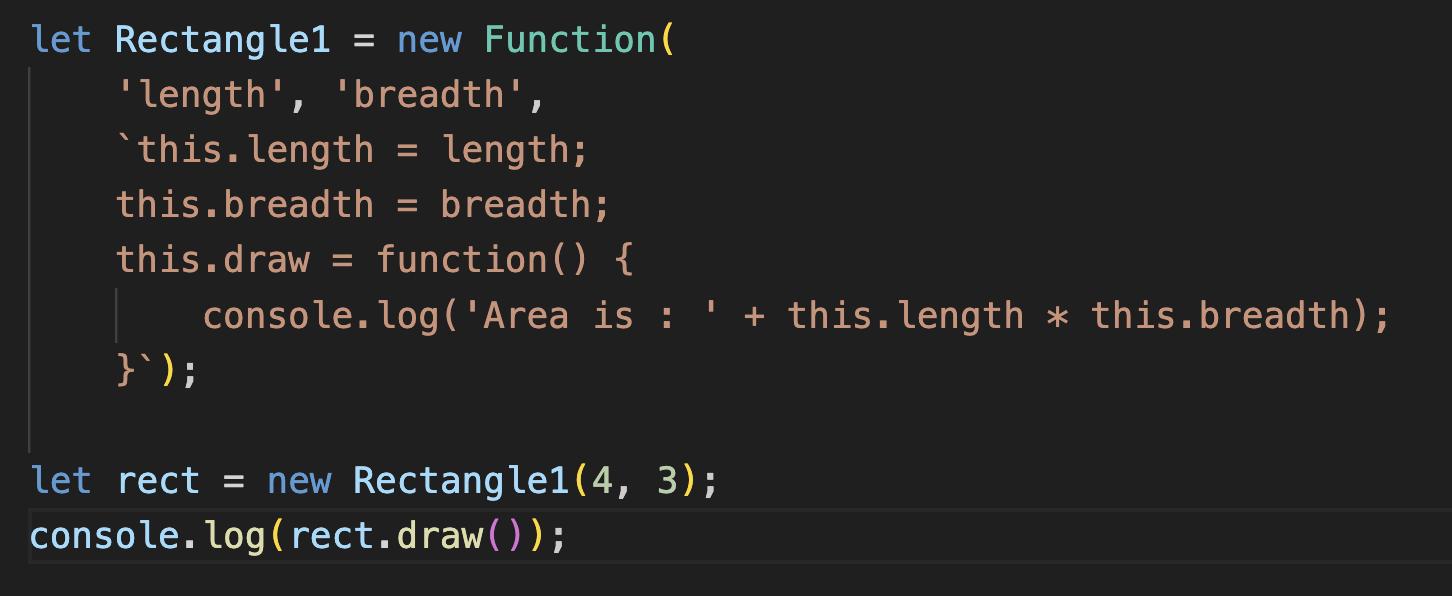
* + if we are creating a function in object we call it as method.
* These are basically called as Object Oriented Programming
* Now if we want to create multiple objects of same types
* Two ways to do this
  + Factory Function
    - 
    - Here but the main issue is that the values are constant and we want that the value should be updated at the time of creation then there we can use parameters
  + Constructor Function
    - Pascal Notation is used (NumberOfStudents)



* Dynamic Nature of Object
  + We can add, update, remove the property after declaration



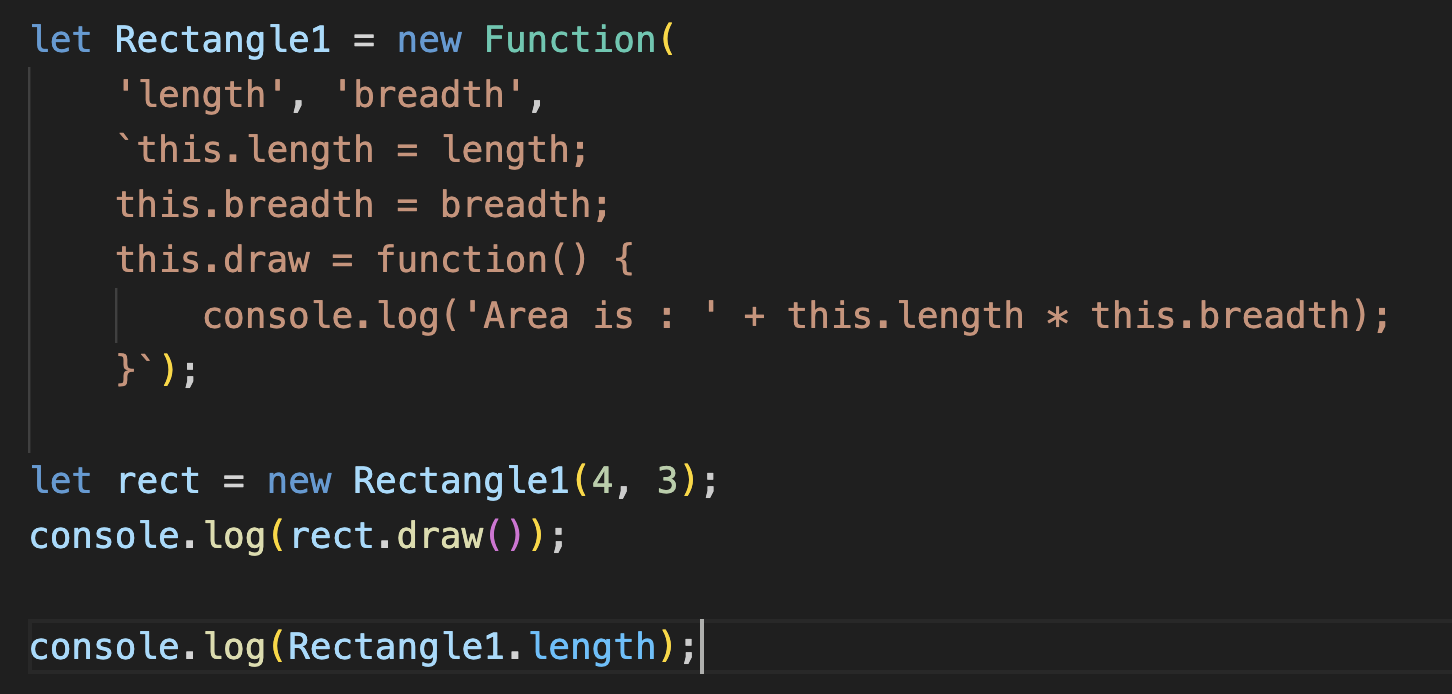
* Constructor Property
  + It is a predefined function which initialize the object or create the object.
  + Every object has a constructor function
  + If we have created our own constructor then that function will also have inbuilt constructor function.



* + Using back-tick character to give the body of the constructor function.

**Functions are Object?**

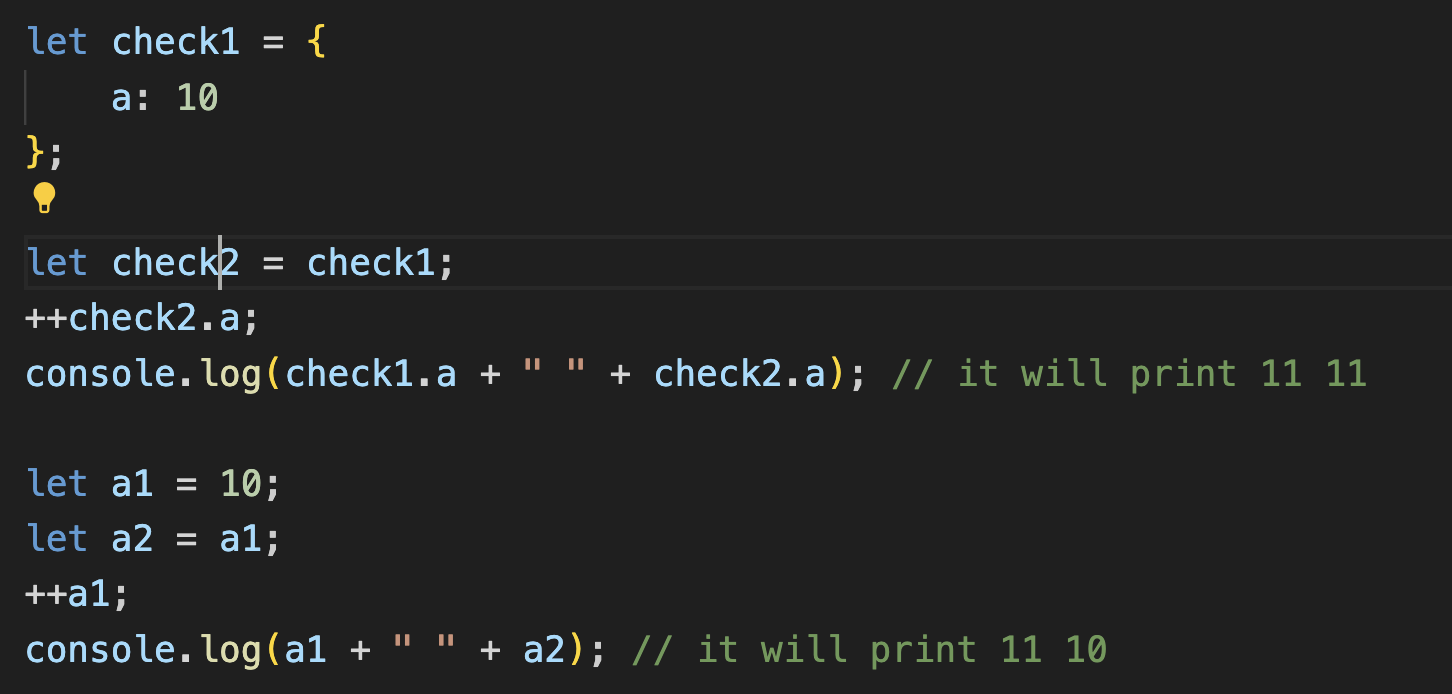
* Yes, functions are also object.

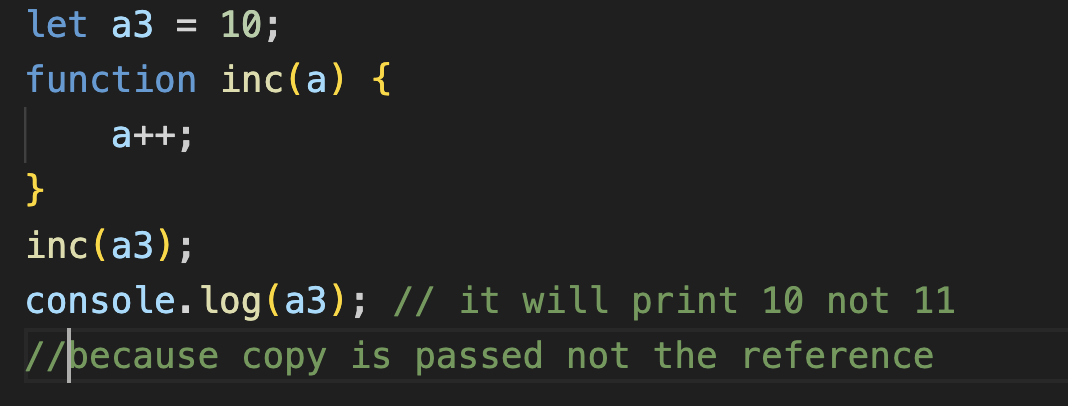


* Here we can see we can use directly length using the function name.

**Difference between the Primitive Types and Reference Types**

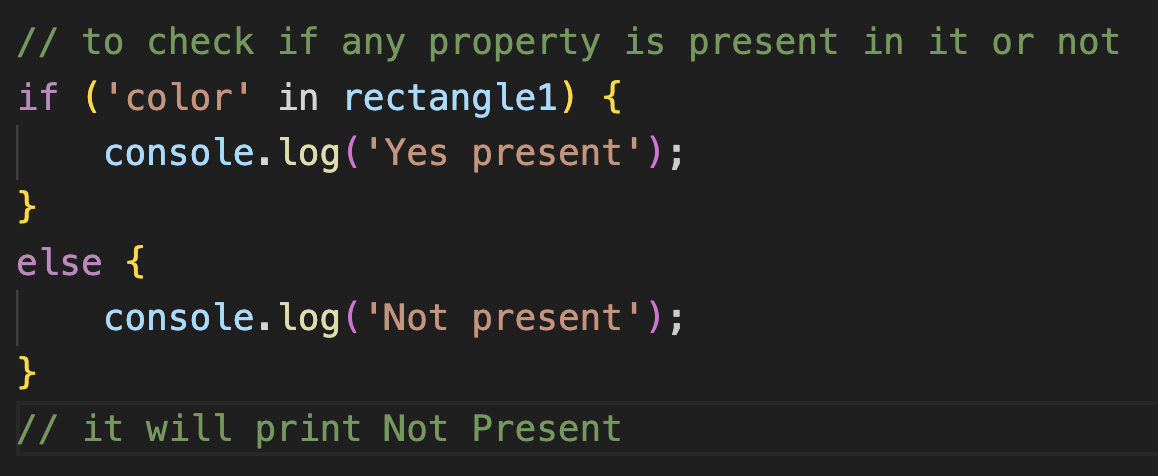
* Reference (address) is same in reference types but not in primitive types.





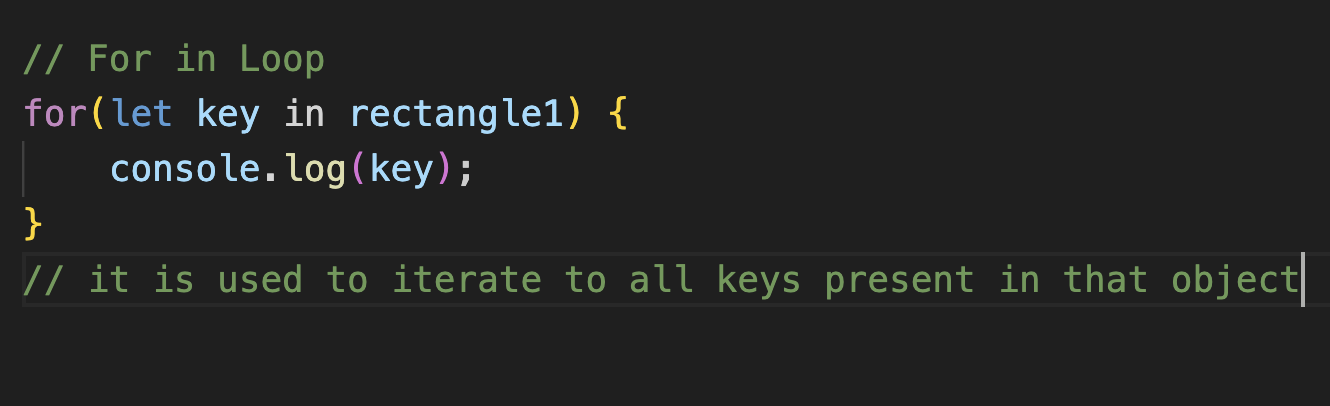
**To check if any properties is present or not**

* Simply use if else

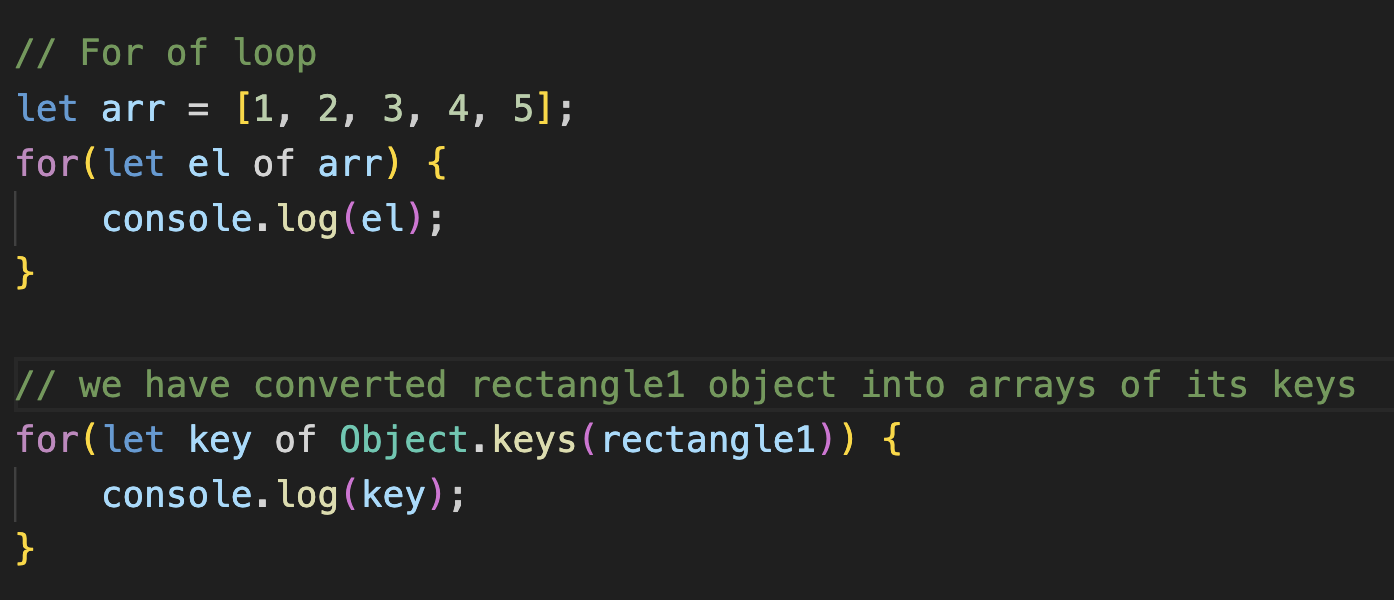


**Iterating through Object**

* **For in Loop**
  + To iterate the property



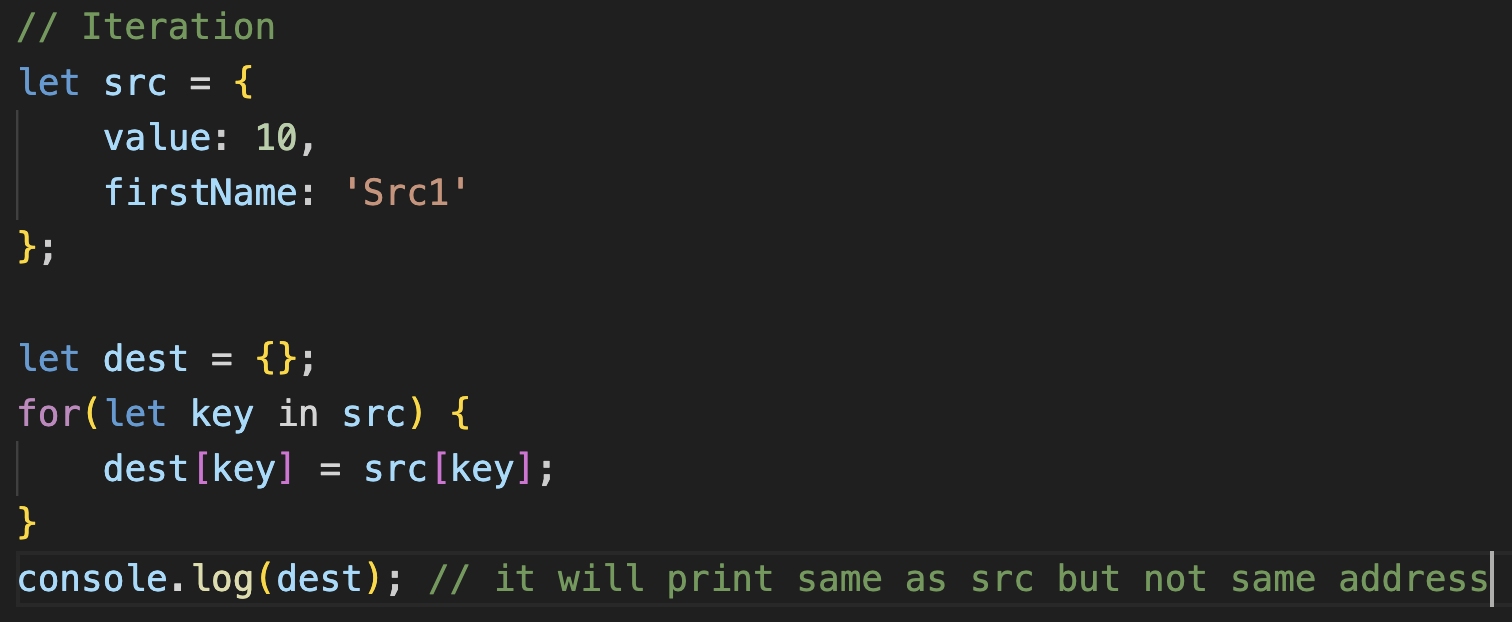
* + Here rectangle1 is an object.
* For of Loop
  + It is used on iterrable like arrays, maps etc.
  + It cannot be used on object it will give error
  + But if we want to use it on Object we need to convert that object into arrays.



* + To convert its entries simply write Object.entries(rectangle1)

**Object Cloning (different address or memory space for different variable)**

* Iteration:
  + Create empty object
  + Use for in loop and copy the values



* Assign:
  + Using assign keyword
  + let dest = Object.assign({}, src);



* + here empty object is being filled by src object values.
  + We can use multiple src object like {}, src1, src2 and so on. It will copy all in that empty object.
* Spread:
  + Using …obj\_name



**Garbage Collection**

* Automatically free up the space which are not being used currently, done via garbage collector.
* We have no control on garbage controller. Runs in the background.