Assignment 2

Marianna Hollanda Campos Pedroso

Holland College: CIS-2286

February 2nd, 2021

1. A closure is the way that global variables can be made local (or private). It is a scope chain, a feature in Javascript where an inner function has access to the outer (enclosing) function’s variables. The closure has three scope chains: it has access to its own scope, it has access to the outer function’s variables, and it has access to the global variables. A simple closure example is:

function outer() {

var b = 10;  
 function inner() {  
   
 var a = 20;   
 console.log(a+b);  
 }  
 return inner;  
}

**var X = outer(); //outer() invoked the first time  
var Y = outer(); //outer() invoked the second time**

When the “outer ( )” function is first invoked, a variable “b” is created, its scope is limited to the “outer ( )” function, and its value is set to 10. The next line is a function declaration, it has nothing to execute. The “return inner” looks for a variable called “inner,” finds that this is actually a function, and returns the entire body of the function “inner.” The contents returned by the return statement are stored in “x,” that will store the following:

function inner() {  
var a=20;  
console.log(a+b);  
}

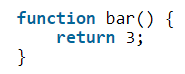
The function “outer ( )” finishes execution, and all variables within the scope of “outer ( )” now no longer exist. Once a function completes its execution, any variables that were defined inside the function scope cease to exist. In “console.log (a+b),” the variable “b” exists only during the execution of the “outer ( )” function. Once the “outer” function has finished execution, the variable “b” no longer exists.

When “outer ( )” is inked the second time, a new variable “b” is created, its scope is limited to the “outer ( )” function, and its value is set to 10. The next line is a function declaration. The “return inner” returns the entire body of the function “inner.” The contents returned by the return statement are stored in “y.” The “outer ( )” finishes execution, and all variables within the scope of “outer ( )” now no longer exist.

In other words, the inner function preserves the scope chain of the enclosing function at the time the enclosing function was executed, and thus can access the enclosing function’s variables.

2) A function declaration defines a named function variable without requiring variable assignment and a function expression defines a function as a part of a larger expression syntax. Function declarations occur as standalone constructs and cannot be nested with non-function blocks. Functions expressions can be named or anonymous. Function declarations must begin with “function” and function expressions must not start with “function.”

An example of function declaration:



An example of function expression:

