Andrew Rachek

Professor Hishon

Web & Internet Programming

2/20/2024

HW #4

1. Javascript is a scripting language that allows you to dynamically update a webpage while it is being viewed in the browser. Javascript allows you to access and modify content while also creating program rules and reacting to events. An interpreted program language means that the program’s instructions will be read and executed by another program. In comparison, compiler programs will compile the code and express the instructions of the target machine. Some of the benefits of using an interpretive program include having the capability of dynamically changing code without a compiler stage.
2. A var element can be redefined and updated. They also may act as global variables unless they are placed within a function block, in which case they’ll act as local variables. Let will define a variable that can be updated but not redefined. It will also be locally scoped to the block that it is placed in. Const variables cannot be redefined or redeclared. They are also scoped to the block they are defined in.
3. An anonymous function is a function that has no name. This can be achieved by placing the function in the position where an interpreter would expect an expression. One example is below:

var area = function(width, height) {

return width \* height;

}

A higher-order function is a function that accepts functions as parameters and/or returns a function.

1. An object groups together a set of variables and functions to create a model of something you would recognize from the real world. In an object, variables become known as properties, and functions become known as methods. One key difference between JavaScript and other OOP languages is the use of ‘this’ as a keyword. The ‘this’ will refer back to the default object defined in the global scope. It can be used within a function to refer to properties and methods defined in the object that was declared in the global scope. In the context of using ‘this’ in an object, ‘this’ will refer to the object that is currently executing the code. This means that within a method of an object, ‘this’ refers to the object itself. For example, an object with the method: console.log(this) would refer to the object itself when called upon. Objects in javascript are different because they inherit properties and behaviors directly from other objects through their prototype chain. This is called prototype-based inheritance, and it differs from other OOP languages that use class-based inheritance. This allows javascript to be more dynamic and flexible compared to class-based inheritance and allows the objects to inherit and modify properties at runtime.