Rohit Roy

February 20th, 2024

Internet and Web Programming (CISC-3300-R01)

Fordham University

**Homework 4**

1. JavaScript is a high-level, interpreted scripting language that is widely used to create dynamic and interactive effects on web pages. Unlike compiled languages, interpreted languages like JavaScript are not transformed into machine code before being run. Instead, they are read and executed line by line by an interpreter, which is typically the web browser for JavaScript. This means JavaScript can be written and run on-the-fly without the need for a compilation step, making it quick to deploy and test. However, this can also lead to slower execution times compared to compiled languages, which are converted into machine code by a compiler before being run, allowing for optimizations that can make them faster to execute.
2. **var**, **let**, and **const** are all used to declare variables in JavaScript, but they have different scopes and behaviors. **var** is function-scoped, meaning if it's declared inside a function, it can only be accessed within that function. However, if it's declared outside of a function, it has global scope. Variables declared with **var** can be re-declared and updated.  
     
   **let** and **const** are both block-scoped, meaning they are only accessible within the block they are defined in (e.g., if, for loops). **let** variables can be updated but not re-declared within their scope. On the other hand, **const** is used to declare constants; variables that cannot be re-declared or updated.
3. Anonymous functions are functions without a name. They are often used as arguments to higher-order functions or as a way to encapsulate functionality to prevent polluting the global namespace. For example, **function() { ... }** is an anonymous function.  
     
   Higher-order functions are functions that take other functions as arguments or return a function as their result. They are a fundamental concept in functional programming, allowing for the creation of abstractions and manipulations of behaviors with functions. An example of a higher-order function is **Array.prototype.map**, which takes a function as an argument and applies it to every element of an array.
4. Objects in JavaScript are collections of properties, where a property is an association between a name (or key) and a value. A value of a property can be a function, in which case the property is known as a method. In JavaScript, objects are prototype-based, which means that they can inherit properties from other objects (prototypes), rather than from classes (as is common in class-based OOP languages like Java or C++). This prototype-based inheritance is a form of OOP but differs from the classical OOP approach, where inheritance is class-based and objects are instances of classes.
5. See homework4.js