

**ADDIS ABABA UNIVERSITY**

**ADDIS INSTITUTE OF TECHNOLOGY**

**CENTER OF INFORMATION TECHNOLOGY AND SCIENTIFIC COMPUTING**

**ASSIGNMENT 1 (New One)**

**Prepared By - Binyam Lemma**

**ID - ATR/1341/10**

**Department - IT**

**To - Mr. Fistum A.**

**Introduction**

**What is JavaScript?**

JavaScript is the world’s most popular programming language.

JavaScript is a text-based lightweight, interpreted, or just-in-time (JIT), compiled with the first-class function programming language used on both the client-side and server-side that allows a user to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages. JavaScript gives web pages interactive elements that engage a user. Common examples of JavaScript that people might use every day include the search box on Amazon, or refreshing a Twitter feed.

**Applications of JavaScript**

JavaScript is usually used for web-based applications and web browsers. However, JavaScript can also be applied beyond the Web in software, servers, and embedded hardware controls. Here are some basic things JavaScript is used for:

1. **Adding interactive behavior to web pages**

JavaScript allows users to interact with web pages. There are almost no limits to the things that can be done using JavaScript on a web page – these are just a few examples:

* Show or hide more information with the click of a button
* Change the color of a button when the mouse hovers over it
* Slide through a carousel of images on the homepage
* Zooming in or zooming out on an image
* Displaying a timer or count-down on a website
* Playing audio and video on a web page
* Displaying animations
* Using a drop-down hamburger menu

1. **Creating web and mobile apps**

Developers can use various JavaScript frameworks for developing and building web and mobile apps. JavaScript frameworks are collections of JavaScript code libraries that provide developers with pre-written code to use for routine programming features and tasks—literally a framework to build websites or web applications around.

Popular JavaScript front-end frameworks include React, React Native, Angular, and Vue.js. Many companies use Node.js, a JavaScript runtime environment built on Google Chrome’s JavaScript V8 engine. A few famous examples include PayPal, LinkedIn, Netflix, and Uber!

1. **Building web servers and developing server applications**

Beyond websites and apps, developers can also use JavaScript to build simple web servers and develop the back-end infrastructure using Node.js.

1. **Game development**

Of course, you can also use JavaScript to create browser games. These are a great way for beginning developers to practice their JavaScript skills.

**Why use JavaScript over other programming languages?**

Aside from the unlimited possibilities, there are many reasons for web developers to use JavaScript over other programming languages:

* JavaScript is the only programming language native to the web browser
* JavaScript is the most popular language
* There’s a low threshold to get started
* It’s a fun language to learn

**Compiled language VS Interpreted language**

Both Interpreter and Compiler are special programs that convert a program into machine language.

Compiled languages are the ones who comply with the human-understandable language (programming languages) to the machine-readable language before they are executed. However, interpreted languages are also human-readable languages (programming languages) and need a translation down to machine languages to get executed, but this translation is done at runtime. Therefore there should be an interpreter in the environment, before running an interpreted language, but compiled language applications can run directly once they are compiled.

An interpreter runs and interprets the code line-by-line (making it slower than a compiler), checks for errors along with it, and stops when there is one. However, a compiler does not run the code. It directly compiles the whole code and is faster in checking for errors as compared to an interpreter.

**Is JavaScript Interpreted Language in its entirety?**

Traditionally, JavaScript is an interpreted language, but this is not necessarily true at all times. For instance, JavaScript runs the V8 engine on Chrome, which compiles its native code internally. This compilation helps realize results on time. The same holds for Trace Monkey and Rhino.

The truth is that JavaScript has undergone significant evolution. This evolution has prompted the development of JIT compilers, which help optimize execution.

This is a common misconception now when someone says that javascript is an interpreted language then yes there is some truth to it but it depends on the implementation, you can implement a javascript engine that only compiles

In conclusion, JavaScript can be interpreted and/or compiled it depends on the implementation.

**The history of “type of null”?**

From the MDN page about the behavior of the type of operator: null. // This stands since the beginning of JavaScript type of null === 'object';. In JavaScript, “type of null” is 'object', which incorrectly suggests that null is an object. This is a bug and one that unfortunately can't be fixed because it would break existing code.

Consequently, null had 0 as its type tag, and since 0 is for objects, then “type of” null fraudulently gives us “object”. For an in-depth explanation and  In JavaScript, “type of” null is 'object', which incorrectly suggests that null is an object.

**Why is it not fixed?**

This bug is not fixed unfortunately because it would break existing code.

**Explain in detail why hoisting is different with let and const?**

**What is Hoisting?**

Hoisting in terms of JavaScript means that a variable is created in memory during the compile phase, and thus it can be used before it is declared.

**Semicolons in JavaScript: To Use or Not to Use?**

**Expression vs  Statement in JavaScript?**

**Reference**

Hoisting -<https://www.w3schools.com/js/js_hoisting.asp#:~:text=Hoisting%20is%20JavaScript's%20default%20behavior%20of%20moving%20all%20declarations%20to,script%20or%20the%20current%20function>)