|  |  |
| --- | --- |
|  | **Jan,2021** |
|  | ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)  Prepared By, Simele Geleta,Atr/9018/12 |



|  |
| --- |
| **[Reading Assingnment on Based on Lecture 04]** |
| The assignment is based on lecture 04(Java script). It contains 5 questions with good explanations of the questions. |

Contents

1. Is Javascript Interpreted  Language in it entirety ? : Check this [Link](https://medium.com/@almog4130/javascript-is-it-compiled-or-interpreted-9779278468fc) and Make Up your justification

Before deciding whether java script is interpreted entirely or not first let’s see some points about java script.

JavaScript is primarily a client-side language. JavaScript started at Netscape, a web browser developed in the 1990s. A webpage can contain embedded JavaScript, which executes when a user visits the page. The language was created to allow web developers to embed executable code on their web pages, so that they could make their web pages interactive, or perform simple tasks. Today, browser scripting remains the main use-case of JavaScript.

JavaScript’s syntax is heavily inspired by C++ and Java. If you have experience in C++ or Java, JavaScript’s syntax will seem familiar to you. However, the inner workings of JavaScript is closer to a dynamically-typed, interpreted language such as Python or Ruby.

Now let’s see the difference between interpreted and compiled programming languages

**Compiled Languages**

Compiled languages are converted directly into machine code that the processor can execute. As a result, they tend to be faster and more efficient to execute than interpreted languages. They also give the developer more control over hardware aspects, like memory management and CPU usage.

Compiled languages need a “build” step – they need to be manually compiled first. You need to “rebuild” the program every time you need to make a change. In our hummus example, the entire translation is written before it gets to you. If the original author decides that he wants to use a different kind of olive oil, the entire recipe would need to be translated again and resent to you.

Examples of pure compiled languages are C, C++, Erlang, Haskell, Rust, and Go.

**Interpreted Languages**

Interpreters run through a program line by line and execute each command. Here, if the author decides he wants to use a different kind of olive oil, he could scratch the old one out and add the new one. Your translator friend can then convey that change to you as it happens.

Interpreted languages were once significantly slower than compiled languages. But, with the development of [just-in-time compilation](https://guide.freecodecamp.org/computer-science/just-in-time-compilation), that gap is shrinking.

Examples of common interpreted languages are PHP, Ruby, Python, and JavaScript.

Some say “JavaScript is an interpreted language, not a compiled language. A program such as C++ or Java needs to be compiled before it is run. The source code is passed through a program called a compiler, which translates it into bytecode that the machine understands and can execute. In contrast, JavaScript has no compilation step. Instead, an interpreter in the browser reads over the JavaScript code, interprets each line, and runs it. More modern browsers use a technology known as Just-In-Time (JIT) compilation, which compiles JavaScript to executable bytecode just as it is about to run.”