|  |  |
| --- | --- |
|  | **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 Java script Interpreted Language in it entirety ? :

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 byte code 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 byte code just as it is about to run.”

2.The history of “type of null”

The null value is technically a primitive, the way "object" or "number" are primitives. This would typically mean that the type of null should also be "null". Null in java script is actually value and type created to stimulate errors and keywords common in other programming languages However, this is not the case because of a peculiarity with the way JavaScript was first defined.

In the first implementation of JavaScript, values were represented in two parts - a type tag and the actual value. There were 5 type tags that could be used, and the tag for referencing an object was 0. The null value, however, was represented as the NULL pointer, which was 0x00 for most platforms. As a result of this similarity, null has the 0 type tag, which corresponds to an object.

Null and undefined in JavaScript are actually values and types created to simulate errors and keywords common in other programming languages.

When a variable is `undefined`, or unitialized, in most programming languages it means that a space in memory has been assigned to a variable name, but the programmer has not yet done anything with that space in memory. This usually results in a compile time error.

When a variable is `null` in other programming languages, null is typically a keyword to indicate the space in memory is a pointer (reference), and that pointer is pointing to an invalid memory address (usually 0x0). This is usually used when a programmer is done using the value of a variable and wants to purposefully clear it by literally pointing it to nothing.

In JavaScript, `null` and `undefined` are values and types. Just like numbers and characters, `null` has a specific configuration of 1’s and 0’s that indicates it’s type is `null` and that it’s value is `null`. Same with `undefined`. These are used in JavaScript to act as placeholders to let the programmer know when a variable has no value.

References

* <https://web.stanford.edu/class/cs98si/slides/overview.html#:~:text=JavaScript%20is%20an%20interpreted%20language,compiled%20before%20it%20is%20run>.
* <https://www.freecodecamp.org/news/compiled-versus-interpreted-languages/#:~:text=Interpreted%20vs%20Compiled%20Programming%20Languages%3A%20What's%20the%20Difference%3F,-Every%20program%20is&text=In%20a%20compiled%20language%2C%20the,reads%20and%20executes%20the%20code>.
* <https://medium.com/@almog4130/javascript-is-it-compiled-or-interpreted-9779278468fc>
* <https://medium.com/@stephenthecurt/a-brief-history-of-null-and-undefined-in-javascript-c283caab662e>
* <https://bitsofco.de/javascript-typeof/>
* <https://medium.com/javascript-in-plain-english/how-hoisting-works-with-let-and-const-in-javascript-725616df7085>