**27 Day**

Difference between High-Level and Low-Level Languages:

1. High-Level Language:

Abstraction Level: High-level languages are designed to provide a high level of abstraction from the hardware. This means they use human-readable code with words and phrases that resemble natural language, making it easier for programmers to write and understand code.

Readability and Ease of Use: High-level languages prioritize readability and ease of use. They use meaningful variable names, structured control flow (like loops and conditionals), and provide built-in functions and libraries for common tasks.

Portability: High-level code is often platform-independent or requires minimal modifications to run on different systems. This portability is achieved through the use of interpreters or compilers that translate high-level code into machine code for different platforms.

Execution Speed: Code written in high-level languages may run slower than low-level code because of the additional layers of abstraction. However, modern high-level languages often use optimization techniques to minimize this performance gap.

2. Low-Level Language: -

Abstraction Level: Low-level languages provide a low level of abstraction and are closely tied to the hardware. They use simple, machine-specific instructions that are challenging to read and write for humans.

Readability and Ease of Use: Low-level languages involve using cryptic symbols and instructions specific to a particular computer architecture. Writing and understanding low-level code is difficult and error-prone compared to high-level languages.

Portability: Low-level code is highly dependent on the hardware and may require significant modifications to run on different systems. It lacks the portability of high-level languages.

Execution Speed: Low-level code can be highly optimized for specific hardware, resulting in faster execution. It has a performance advantage over high-level code but sacrifices readability and ease of development.

JavaScript (JS) as a High-Level Language:

JavaScript is a high-level language. Here's why:

Abstraction Level: JavaScript abstracts many complex details of computer hardware and provides a user-friendly, human-readable syntax. It's designed to be accessible and understandable for web developers.

Readability and Ease of Use: JavaScript code is relatively easy to read and write, thanks to its English-like syntax, use of variables, and structured control flow. This makes it suitable for web development and scripting tasks.

Portability: JavaScript is highly portable because it can run in various web browsers and server environments without significant modifications to the code. It's not tightly bound to specific hardware.

Execution Speed: While JavaScript may not be as fast as low-level languages like C or assembly, modern JavaScript engines use Just-In-Time (JIT) compilation and optimizations to deliver competitive performance.

JavaScript's high-level nature, along with its ubiquity in web development, makes it an accessible language for a wide range of developers. It abstracts many hardware complexities, allowing programmers to focus on creating web applications and interactive websites without needing to worry about low-level details.