Lenguajes compilados vs Lenguajes interpretados

Lenguajes compilados: Son lenguajes de programación que deben ser convertidos a ***código de máquina*** previamente, este paso es conocido como la compilación. Por ello, las aplicaciones suelen ser más rápidas y eficientes al compararlo con lenguajes interpretados.

Ventajas:

- Se pueden detectar errores durante el proceso de compilación.
- Tienen mejor rendimiento que los lenguajes interpretados.

Desventajas:

- Dependen de la plataforma en la cual haya sido generado el código binario en el proceso de compilación.
- Con cada cambio es necesario compilar nuevamente la aplicación.
- No puedes ejecutar instantáneamente el código para hacer pruebas. Por lo cual puede ser más lento validar los cambios realizados al código.

Ejemplos de Lenguajes Compilados:
C, C++, Rust, entre otros.

¡Pero hay lenguajes híbridos! Un ejemplo de esto es Java, el código fuente es compilado a un lenguaje intermedio llamado bytecode y este es interpretado para ser ejecutado por JVM (Java Virtual Machine) en alguna plataforma específica (Linux, Windows, etc.)

Lenguajes interpretados: Son lenguajes de programación que son ejecutadas directamente sin un paso previo (compilación) para transformarlo en lenguaje de máquina, este código necesita de un intérprete el cual ejecuta directamente las instrucciones del código realizado.

Ventajas:

- Facilidad para prototipar aplicaciones.
- Portátiles y multiplataforma, por lo tanto, el mismo código puede ejecutarse en diversos sistemas operativos y plataformas de hardware.

Desventajas:

- El debugging ocurre en tiempo de ejecución.
- Suele ser más lento.
- Mayor consumo de recursos.

Ejemplos de Lenguajes Interpretados:
JavaScript, Python, PHP, entre otros.

Links de referencia 1 2 Cursos relacionados JS Más recursos