

Lenguajes compilados vs. interpretados



Diferencias, ventajas, ejemplos y usos

Septiembre 2025

Juan Romero Sánchez

IES Trassierra



Introducción

- Existen dos grandes tipos de lenguajes de programación: compilados e interpretados.
- La diferencia está en cómo se traduce y ejecuta el código fuente.
- Ejemplos:
 - Compilados: C, C++, Rust.
 - Interpretados: Python, JavaScript, PHP.

Diferencia principal

- Lenguajes compilados:
 - El código fuente se traduce completamente a lenguaje máquina (binario).
 - El ejecutable es específico de una plataforma.
- Lenguajes interpretados:
 - El código se traduce y ejecuta línea por línea en tiempo real.
 - Mayor flexibilidad y portabilidad.

Proceso de compilación vs. interpretación

- Compilación:
 - Traducción previa del programa completo.
 - Requiere un compilador.
 - Resultado: archivo ejecutable.
- Interpretación:
 - Traducción y ejecución simultánea.
 - Requiere un intérprete.
 - No genera archivo binario independiente.

Python como lenguaje interpretado

- Python traduce el código a un bytecode intermedio.
- Ese bytecode es ejecutado por la Máquina Virtual de Python (PVM).
- Ventajas:
 - Portabilidad entre plataformas.
 - Fácil depuración y modificación.
- Desventajas:
 - Menor rendimiento que lenguajes compilados.
 - Mayor uso de memoria en aplicaciones intensivas.

Tabla comparativa

Característica	Lenguajes Compilados	Lenguajes Interpretados
Traducción	Código fuente → binario	Línea por línea en ejecución
Ejecución	Muy rápida	Más lenta
Portabilidad	Limitada (depende de plataforma)	Alta (mismo código en diferentes sistemas)
Detección de errores	En tiempo de compilación	En tiempo de ejecución
Seguridad del código	Mayor (código fuente oculto)	Menor (código fuente visible)
Flexibilidad	Menor	Mayor (más fácil de modificar)
Tamaño del programa	Suele ser mayor	Suele ser menor
Necesidad de herramientas	Compilador	Intérprete
Ejemplos	C, C++, Rust, Go	Python, JavaScript, PHP
Uso principal	Sistemas embebidos, videojuegos, apps de alto rendimiento	Scripts, prototipos, automatización, web

Ejemplos de uso (interpretados)

- Cuándo convienen:
 - Desarrollo web, prototipado rápido, automatización.
 - Análisis de datos, machine learning, scripts de sistema.
- Ejemplos:
 - Python: análisis de datos, automatización.
 - JavaScript: interacción web.

Ejemplos de uso (compilados)

- Cuándo convienen:
 - Aplicaciones de alto rendimiento (videojuegos, simulaciones).
 - Software de sistemas, drivers, backend de alto rendimiento.
 - Cuando se necesita seguridad (ocultar el código fuente).
- Ejemplos:
 - C++: videojuegos y aplicaciones gráficas.
 - Rust: servicios backend, gestión de memoria crítica.

Consideraciones finales

- Ambos paradigmas tienen ventajas y desventajas.
- Hoy en día muchos proyectos combinan ambos enfoques:
 - Núcleo en compilado para rendimiento.
 - Scripts en interpretado para flexibilidad.
- Python es un gran ejemplo de lenguaje interpretado muy utilizado actualmente.