

Difference Between Compiled and Interpreted Languages

The comparison between **compiled** and **interpreted** languages lies in how source code is **translated into executable instructions** for a machine.

Compiled Languages

- **Definition:** The entire source code is translated into machine code **before execution** by a compiler.
- **Output:** Produces a **standalone binary** or executable file.
- **Execution Speed:** Typically **faster**, since the code is already translated.
- **Error Detection:** Most errors are caught **at compile time**.
- **Examples:** C, C++, Rust, Go

Key Characteristics:

- Requires a compilation step before running.
 - Optimized for performance.
 - Platform-dependent unless explicitly cross-compiled.
-

Interpreted Languages

- **Definition:** Source code is **executed line by line** by an interpreter **at runtime**.
- **Output:** No separate binary; runs through an interpreter.
- **Execution Speed:** Usually **slower**, since translation happens on the fly.
- **Error Detection:** Errors may only appear **during execution**.
- **Examples:** Python, JavaScript, Ruby

Key Characteristics:

- Easier to debug and modify on the fly.
 - More flexible for scripting and rapid development.
 - Platform-independent, assuming interpreter is available.
-

Hybrid Languages

Some modern languages use a **mix** of both compilation and interpretation for flexibility and performance.

C# – Compiled or Interpreted?

C# falls into the **hybrid** category:

- **Step 1: Compilation**
 - C# source code (.cs) is compiled by the **C# compiler** into **Intermediate Language (IL)** code.
 - This produces a .dll or .exe file that runs on the **.NET runtime**.
- **Step 2: Execution (JIT Compilation)**
 - At runtime, the **Common Language Runtime (CLR)** uses a **Just-In-Time (JIT) compiler** to convert IL into native machine code.
 - This native code is then executed by the CPU.

Summary of C# Behavior:

- **Compiled:** Yes, to Intermediate Language (IL).
- **Interpreted:** Not exactly, but IL is **JIT-compiled** at runtime.
- **Result:** C# is **not purely compiled or interpreted**, but uses **JIT compilation** for optimized execution.

Aspect	Compiled	Interpreted	C#
Translation Time	Before execution	At runtime	Before + during runtime
Output	Machine code	Interpreter executes	Intermediate Language
Performance	Fast	Slower	Fast (after JIT)
Portability	Low	High	Medium (via .NET)
Example Languages	C, C++	Python, JS	C#