Rozdělení programovacích jazyků na kompilované a interpretované vychází z toho, jakým způsobem se zpracovává a vykonává zdrojový kód programu.

### **Kompilované jazyky:**

Kompilované jazyky jsou ty, jejichž zdrojový kód je nejdříve přeložen pomocí **kompilátoru** do strojového kódu, tedy do formy, které rozumí konkrétní hardware, na kterém má být program spuštěn. Tento strojový kód je poté vykonán přímo procesorem bez nutnosti dalšího překladu během běhu programu. Proces kompilace probíhá ještě před samotným spuštěním programu.

**Kompilátor** je tedy program, který překládá celý zdrojový kód programu do spustitelného formátu najednou, což má určité výhody i nevýhody. Výhodou je, že výsledný program běží rychleji, protože již není potřeba žádné další překládání za běhu. Nevýhodou může být delší doba vývoje, protože každý upravený kód je nutné znovu přeložit před dalším spuštěním.

**Příkladem kompilovaného jazyka** je například **C** nebo **C++**, a příkladem kompilátoru pro tento jazyk je třeba **GCC** (GNU Compiler Collection). V tomto případě se zdrojový kód programu přeloží do binárního spustitelného souboru, který je možné spustit na cílovém systému.

### **Interpretované jazyky:**

Na druhé straně interpretované jazyky jsou ty, jejichž zdrojový kód je překládán a vykonáván postupně po jednotlivých částech za běhu programu. Tento proces zajišťuje **interpret**, což je program, který provádí příkazy přímo ze zdrojového kódu, bez nutnosti překladu celého kódu dopředu.

**Interpret** tedy zpracovává a vykonává kód přímo při spuštění programu, což znamená, že každá změna ve zdrojovém kódu je okamžitě viditelná při dalším spuštění. Tento přístup je výhodný při ladění, protože není nutné program znovu překládat. Nevýhodou je ale to, že vykonání programu může být pomalejší, protože kód je překládán a vykonáván za běhu.

**Příkladem interpretovaného jazyka** je **Python** a příkladem jeho interpretu je například **CPython**. V tomto případě interpret čte a vykonává Python kód přímo, takže při každém spuštění probíhá překlad kódu znovu.

Zatímco kompilované jazyky obvykle vedou k rychlejšímu běhu výsledného programu, interpretované jazyky poskytují větší flexibilitu během vývoje a ladění. Oba přístupy mají své výhody a nevýhody, a volba mezi nimi závisí na konkrétním použití a požadavcích na vývoj programu.