

تفاوت مفسر و کامپایلر در برنامه نویسی

نوشته شده توسط معین حاج ملک

در ابتدا برای اینکه یک زبان برنامه نویسی بتواند یک برنامه را اجرا کند باید آن برنامه را به زبان ماشین تبدیل کند تا بتوان با استفاده از سیستم آن را اجرا کرد.

یکسری از زبان های برنامه نویسی هستند که به زبان ماشین نزدیک ترن مانند زبان ماشین و زبان اسمبلی که به انها زبان سطح پایین میگویند که نیازی به تبدیل ندارند ولی زبان هایی مثل پایتون که زبان سطح بالا هست باید اول به زبان ماشین تبدیل بشوند و سپس اجراشون در این بین یک سری واسطه قرار دارد که زبان برنامه نویسی را به زبان ماشین تبدیل میکنند که یکی شون **کامپایلر** و دیگری **مفسر** نام دارد.

کامپایلر و مفسر یک نرم افزاری هستند که کد های برنامه نویسی را به زبان ماشین که برای کامپیوتر قابل فهم باشد تبدیل میکند.

اما تفاوت های این دو در چیست :

در کامپایلر باید قبل از اجرای برنامه ها کد ها یک بار به زبان ماشین کامپایل شوند تا برنامه درست کار کند . هر تغییری در کد ها باید دوباره از اول کد ها کامپایل شوند و بعد از کامپایل کردن کد ها یک فایل خروجی از برنامه برای ما ایجاد میشود. اگر هم خطایی در کد ها وجود داشته باشد برنامه کامپایل نمیشود و ارور میدهد.

زبان های برنامه نویسی کامپایلی : C / C++ / C# / Java / Go

در مفسر موقع اجرای برنامه به صورت خط به خط به زبان ماشین ترجمه میشود. در کامپایلر بعد از هر کامپایل یک فایل خروجی به ما میدهد ولی در مفسر اینکار را نمیکند.

زبان های برنامه نویسی مفسری : JavaScript / PHP / Python / Ruby

تفاوت ها :

در زبان کامپایلری قبل از اجرای برنامه همه کد ها به زبان ماشین تبدیل میشوند. ام در زبان های مفسری کد ها به صورت خط به خط به زبان ماشین تبدیل میشوند

در زبان کامپایلری بعد از کامپایل یک فایل خروجی به ما میدهد ولی در مفسری به این صورت نیست و فایلی به ما نمیدهد.

اگر خطایی در کامپایل کردن رخ دهد برنامه خروجی را به ما نمیدهد و ارور میدهد . در مفسری به دلیل اینکه خط به خط کد تبدیل میشود به زبان ماشین اگر در خط کدی مشکلی وجود داشته باشد برنامه در همان جا ایست میکند و ارور را به ما نمایش میدهد.

سرعت اجرای زبان کامپایلری به دلیل اینکه یک فایل خروجی به ما میدهد و دیگر نیز به کامپایل نیست بیشتر از مفسری است که برای هر اجرا نیاز به ترجمه خط به خط دارد.

منابع :