

ما هو الفرق بين Compiler vs. Interpreter

Compiler:

المترجم هو عبارة عن برنامج وظيفته أخذ مصدر الكود (**source code**) وأن يترجمه بالكامل قبل عملية تنفيذه ولذلك فهو يأخذ مدة طويلة في عملية ال **Compiling**, ولكن في عملية تنفيذ الكود بالعكس تماماً يكون سريع جداً

وأيضاً من عيوب اللغات المترجمة أنها لغات تعمل على نظام واحد ولا يمكن اخذ ال executive program نفسه الذي نتج من نظام **Windows** وتنفيذه في نظام **Linux** علي سبيل المثال لماذا؟؟ لأن كل نظام يتعامل مع ال hardware بشكل مختلف وتمثيل ال Binary يمكن أن يختلف

فلذلك تحتاج ترجمته من جديد على كل نظام تشغيل

Interpreter:

المفسر هو عبارة عن software أيضاً وظيفته أخذ المصدر الكودي وتفسيره سطر ب سطر (**runtime**) ولذلك ف هو ربما يأخذ وقتاً أقل في عملية التفسير (نظرياً) ولكن سيأخذ وقتاً أطول في تنفيذ الكود ولذلك باقي الاكواد ستتعمل تنفيذاً إلى أن يتم الانتهاء من تفسير وتنفيذ الاكواد التي قبلها

من **مميزات اللغات المفسرة** أن ال executive program يمكن أن يعمل على العديد من أنظمة التشغيل بدون ترجمته/تفسيره مرة أخرى

HTTPS vs. HTTP

HTTP:

هو بروتوكول (**Protocol**) مجموعة من ال rules التي تحكم عمليات إرسال الطلبات واستقبالها ولكن بدون أي Secure Layer، وبالتالي يمكن التجسس على تلك البيانات

HTTPS:

هو بروتوكول (**Protocol**) مثل السابق ولكن مضافاً إليه Secure Layer كـ **التشفير** لحماية تلك البيانات من عمليات التنصت أو التجسس بها وسرقتها

< ال http هو أول بروتوكول تم إنشائه علي يد العالم Tim-berners Lee الأب الروحي لل Web الحالي وكانت من ضمن العديد من الأشياء التي قام بعملها من أجل World Wide Web مثل (..Hyber-Text - html, HTTP, URL, etc)

Scripting vs. Programming Languages

Scripting

هي لغات لا تحتاج لعملية الCompilation, هي لغات تحتاج لعملية الInterpreting, كما أنها لا تمتلك الصلاحيات الكاملة في أنظمة التشغيل, مثل JavaScript لا تملك القدرة علي dealing with files - Nodejs can do it

Programming (Academic)

هي لغات تحتاج لعملية الCompiling وتحويل الكود ل0s, 1s وإستخراج الملف التنفيذي, كما أنها تمتلك القدرات الكاملة في التعامل مع أنظمة التشغيل ك (File System, Threads, Process, etc ...)

اللغات : C, C++, Java, Objective-C