بسم الله الرحمن الرحيم

مقالة اليوم تتحدث عن ال الجافا سكريبت

ما لا تعرفه عن الجافا سكريبت

كيف تعمل وراء الكواليس

أولا لماذا يعد هذا الأمر مهم بالنسبة اليك

لأن هذا له علاقة بالمعلومات الخاصة باللغة وكيفية تحليل المحرك لها ز.وذلك لتحسين كتابتك للكود ليكون كود سليم وتفاعلى

ما هو محرك الجافا سكريبت

\*هو ببساطة برنامج كمبيوتر لتحليل كود الجافا سكريبت.وهو المسؤول المباشر عن الكود

\*ومحركات البحث الشهيرة تحتوى على محرك للجافا سكريبت

ومن أشهرهم محرك فى8 ونود جى اسعلى متصفح جوجل كروم

والذى استخدم مؤخرا فى بناء سيرفرات جوجل

مكان تخزين الجافا سكريبت

تخزن الداتا الخاصة بالجافا سكريبت فى مكانين

الأول:فى ما يسمى بال كول ستاك

والثانى بال هيب

\*الكول ستاك

هو المكان التى تخزن الكود

\*الهيب

هو المكان التى تسحب منه الداتا الخاصة بالمشروع ككل

كيف يتم تشغيل الجافا سكريبت

الجافا سكريبت عبارة عن بيت يحتاج الى جميع المكونات فى وقت واحد

Web ApIs

وهو اداة وظيفية تدعم المحرك ولكنها ليست جزء من لغة الجافا سكريبت

فهى تسهل عملية المرونة فى نقل الداتا من خلال نموذج دوم

والرسم البيانى لتشغيل الجافا سكريبت على المتصفح يحتوى على

\*محرك الجافا سكريبت

Web ApIs\*

\*وال كول باك

فعند تشغيل المتصفح فيقوم بربط كل عناصر الجافا سكريبت ببعضها عن طريق المرور على الكود وعمل فيد باك له

استراتيجيات الجافا سكريبت

\*just in time

وهى عبارة عن الوقت هنا (في حرف T في المصطلحين) هو وقت التنفيذ.

المصنف compiler يحول من ملف المصدر باللغة العالية مثل سي إلى ملف تنفيذي بلغة الآلة. يتم حفظ الملف التنفيذي وتنفيذه أي عدد من المرات دون الرجوع لملف المصدر. بمعنى إن قمت بتنفيذ الملف 100 مرة فلن يتم إضاعة أي وقت في قراءة وتفسير الملف المصدر بل إن الملف المصدر في الغالب لا يتم توزيعه مع الملف التنفيذي (يتم التحويل على جهاز المطور مرة واحدة) ومن هنا جاء اسم ahead of time compilation.

المفسر Interpreter يستخدم مع اللغات التفسيرية والتي يتم تداولها غالبا في صورتها المصدرية (وتسمى تلك المصادر مخطوطات أو script كما في javascript ). في كل مرة يتم فيها تنفيذ المخطوطة يتم إعادة إعرابها (تقطيعها وتحليلها ...إلخ) وتنفيذها (سطر فسطر وجزء جزء). قد تختزل بعض اللغات مثل بايثون عمليات الإعراب والتقطيع وذلك من خلال حفظ ملف يسمى بايت كود byte code أو كود الآلة الافتراضية VM وهو باختصار يحتوي إعادة كتابة للبرنامج بلغة غير بشرية أسهل على الحاسوب التعامل معها وهي تكون سلسلة من رموز العمليات OP Codes. عند تنفيذ البرنامج من جديد يتم النظر إلى هذا الملف إن كان تاريخه أحدث من الآخر وإلا يتم إعادة توليده. وإلى هنا لا يوجد عملية ترجمة حقيقية إلى لغة الآلة بمعنى أنك لو نظرت إلى كود أي من تلك المفسرات لن تجد فيها تعليمات بلغة الآلة أو لغة التجميع assembly لكن ستجد عبارة if أو switch تقول إن وجدت رمز عملية الجمع اسحب آخر رقمين في المكدس واجمعهما ثم ادفع الناتج.

لغة جافا كانت تستخدم مفسر حتى ظهر hotspot وهو JIT أي مصنف يعمل في الوقت المناسب على الطاير.

يمكنه تحويل من رموز العلميات بصورة byte code أو virtual machine code إلى لغة الآلة الحقيقية native machine code لكنه لا يحتفظ بها في ملف مستقل بل يفعل ذلك على الطاير وهذا يؤدي إلى تسريع الحلقات التكرارية مقارنة بالمفسر لأنها قبل أول دورة تتحول إلى لغة الآلة وكل الدورات التالية تكون بلغة الآلة.

من الطرائف أن بايثون لديها مصنف اسمه jython يمكنه تحويل برامج بايثون إلى ملفات بلغة الآلة الوهمية الخاصة بجافا .class (وبالتالي .jar) وتنفيذها على JIT الخاص بجافا.

كذلك هناك مفسر للغة بايثون كتاب بلغة بايثون اسمه pypy تم دمج JIT فيه أيضا