編譯型語言與解釋型語言的真正區別

其他 · 發表 2018-11-03

編譯型語言與解釋型語言的真正區別

JavaScript、PHP、Python三種語言

PHP 處理異常 和 Python 處理異常 的區別

python也太自動，函式傳遞前會判斷是否要‘值傳遞’，而PHP通通‘值傳遞’

一文解決 PHP靜態（區域性/全域性）變數、auto(區域性/全域性)變數、類中static

使用axis呼叫webservice時,服務端接收到的引數為null

struts2,hibernate,spring下載整合所需JAR包

flex 4 攝像頭拍照

ExtJS4.2學習基於表格的擴充套件外掛---rowEditing

一、編譯型語言與解釋型語言的必需知識點

解釋性語言和編譯型語言

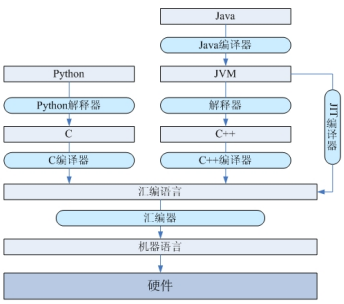
	解釋性語言	編譯型語言
概念	計算機不能直接的理解高階語言，只能直接理解機器語言，所以必須要把高階語言**翻譯**成機器語言，計算機才能執行高階語言的編寫的程式。**翻譯**的方式有兩種，一個是**編譯**，一個是**解釋**。兩種方式只是翻譯的時間不同。	
特徵	解釋性語言是指它常用的執行機制是使用一個“直譯器”來執行，直譯器對於程	編譯型語言是指它常用的執行機制是使用一個“編譯器”來編譯成機器語言，然後你就可以直接執行（執行）這個編譯成的“可執行檔案”。例如C

ExtJS4.2學習(七)EditorGrid可編輯表格

Extjs4.2經驗

	式是一句一句“翻譯”成機器語言來一句一句執行，例如shell指令碼語言。
區別	語言你也可以為解釋性語言（如shell指令碼語言)寫個編譯器來編譯，這樣它就成了“編譯語言” **不管是解釋性語言還是編譯型都可編譯或解釋，前提是有這樣的編譯器或直譯器**（比如你自己寫一個），找不到這樣的編譯器你當然不能編譯對於語言本身來說，各種程式語言本質沒什麼不同。所謂的“解釋性”和“編譯”指的是執行機制上的不同。

二、Python和Java解釋型語言的主流執行方式：



不管是啥語言，都是要轉換成計算機能識別出來的語言。

三、進而可推之解釋性語言和編譯型語言的優缺點

編譯型語言：

1. 編譯型語言最大的優勢之一就是其執行速度。用C/C++編寫的程式執行速度要比用Java編寫的相同程式快30%-70%。
2. 編譯型程式比解釋型程式消耗的記憶體更少。
3. 不利的一面——編譯器比直譯器要難寫得多。
4. 編譯器在除錯程式時提供不了多少幫助——有多少次在你的C語言程式碼中遇到一個“空指標異常”時，需要花費好幾個小時來明確錯誤到底在程式碼中的什麼位置。
5. 可執行的編譯型程式碼要比相同的解釋型程式碼大許多。例如，C/C++的.exe檔案要比同樣功能的

Java的.class檔案大很多。

6. 編譯型程式是面向特定平臺的因而是平臺依賴的。
7. 編譯型程式不支援程式碼中實現安全性——例如，一個編譯型的程式可以訪問記憶體的任何區域，並且可以對你的PC做它想做的任何事情（大部分病毒是使用編譯型語言編寫的）
8. 由於鬆散的安全性和平臺依賴性，編譯型語言不太適合開發因特網或者基於Web的應用。

解釋型語言：

1. 解釋型語言提供了極佳的除錯支援。一名Java程式設計師只需要幾分鐘就可以定位並修復一個“空指標異常”，因為Java執行環境不僅指明瞭異常的性質，而且給出了異常發生位置具體的行號和函式呼叫順序（著名的堆疊跟蹤資訊）。這樣的便利是編譯型語言所無法提供的。

2. 另一個優勢是直譯器比編譯器容易實現
3. 解釋型語言最大的優勢之一是其平臺獨立性
4. 解釋型語言也可以保證高度的安全性——這是網際網路應用迫切需要的
5. 中間語言程式碼的大小比編譯型可執行程式碼小很多
6. 平臺獨立性，以及嚴密的安全性是使解釋型語言成為適合網際網路和Web應用的理想語言的2個最重要的因素。
7. 解釋型語言存在一些嚴重的缺點。解釋型應用佔用更多的記憶體和CPU資源。這是由於，為了執行解釋型語言編寫的程式，相關的直譯器必須首先執行。直譯器是複雜的，智慧的，大量消耗資源的程式並且它們會佔用很多CPU週期和記憶體。
8. 由於解釋型應用的decode-fetch-execute（解碼-抓

取-執行）的週期，它們比編譯型程式慢很多。

9. 直譯器也會做很多程式碼優化，執行時安全性檢查；這些額外的步驟佔用了更多的資源並進一步降低了應用的執行速度。

標籤： 解釋 語言 編譯 程式 程式碼 執行 執行 Java

👉 您可能也會喜歡...

編譯型語言與解釋型語言的真正區別	編譯原理與組合語言的概念區別
	Python程式碼是編譯執行還是解釋執行？
linux驅動編譯時make -C M= 解釋	vs+msys2+yasm編譯libx264+ffmpeg詳細解釋（32位或64位）
關於python代碼是編譯執行還是解釋執行	Python代碼是編譯執行還是解釋執行？
常用cl編譯命令參數解釋	編譯時異常與執行時異常的區別
OC與C語言的幾點區別	異常：編譯時異常和執行時異常

&throw和throws區別try-catch的應用
Java中break, continue, return語句的使用區別
if(){}else和try{}catch{}語句的本質區別
DV型、OV型、EV型證書的主要區別
TCP協議與UDP協議之間的九個區別-JAVA網路面試題

首頁
Python教學