

برمجة الحاسب الآلي

م. محمد أنس كوارة

الأكاديمية العربية الدولية – منصة أعد

مقدمة

برمجة الحاسب: هو عملية كتابة مجموعة من التعليمات تسمى `code`, مكتوبة بلغة برمجة محددة ليتم تنفيذها بواسطة الحاسب.

(أخبار الحاسب بما يجب عليه القيام به, بإعطائه سلسلة من التعليمات باستخدام لغة يفهمها الحاسب)

لغة البرمجة : مجموعة من القواعد و التعبير المحددة تستعمل لتوصيل التعليمات والأوامر للحاسب بطريقة يفهمها و يقوم بتنفيذها.

مقدمة

الخوارزمية : مجموعة من محددة ومعرفة ومنتهاة من التعليمات لحل مشكلة محددة أو تنفيذ مهمة معينة، وهي عبارة عن مجموعة متسلسلة من التعليمات الواضحة لتحقيق هدف محدد.

خصائص الخوارزمية الجيدة

- .1 واضحة
- .2 محدودة الخطوات
- .3 حتمية
- .4 محددة الدخل و الخرج
- .5 عملية

لغات البرمجة

لغة البرمجة : مجموعة من القواعد و التعبير المحددة تستعمل لتوصيل التعليمات و الأوامر للحاسب بطريقة يفهمها و يقوم بتنفيذها.

يمكن تقسيم لغات البرمجة إلى نوعين رئисيين

لغات البرمجة

لغات برمجة منخفضة المستوى: وهي تضمن فئتين رئيسيتين

- لغة الآلة: وهي اللغة الوحيدة التي يفهمها الحاسوب بشكل مباشر وتألف من تسلسل من الأصفار والواحدات، وتختلف من حاسب إلى آخر حسب بنيه.

لغة التجميع assembly: تستخدم اختصارات معبرة من اللغة الإنكليزية لتعبير عن العمليات الرئيسية التي يقوم بها الحاسوب وهي تحتاج محولاً ليحولها إلى لغة الآلة يسمى assembler وتحتفل تعليمات هذه اللغة بحسب بنية الحاسوب المستخدم.

لغات البرمجة

لغات البرمجة عالية المستوى: تستخدم بعض الكلمات المفتاحية الشبيهة بالكلمات التي يستخدمها الإنسان في إعطاء الأوامر للحاسب، كما أنها مستقلة عن بنية الحاسب المستخدم من هذه اللغات (C,C++,JAVA, PYTHON,...)، تتمثل فائدة استخدام لغة برمجة عالية المستوى في سهولة برمجة البرنامج وفهمه وصيانته.

مقارنة بين اللغات منخفضة وعالية المستوى

	لغات البرمجة العالية المستوى High Level Language	لغات البرمجة المنخفضة المستوى Low Level Language
السهولة	أكثر سهولة للمبرمجين	أكثر سهولة للحاسب لكي يفهمها
كفاءة التخزين في الذاكرة	أقل كفاءة من حيث الذاكرة	أعلى كفاءة للذاكرة
سهولة الفهم للمبرمجين	سهلة الفهم للمبرمجين	يصعب على المبرمجين فهمها
سهولة التحليل للمبرمجين	اسهل في تحليل الأخطاء	معقدة في تحليل الأخطاء
سهولة الصيانة	سهلة الصيانة	أصعب نسبياً في صيانتها
الاعتمادية على العتاد Hardware	يمكنها العمل على اي بيئة عمل	تعتمد على العتاد التي تستخدم عليه
المفسر / المجمع	تحتاج إلى مفسر او مترجم لكي تعمل	تحتاج إلى مجمع Assembler لكي تعمل
الاستخدام	تستخدم بشكل كبير في التطبيقات	أقل استخداماً

اللغات العالية المستوى

كما ذكرنا سابقاً فإن الحاسوب يفهم لغة واحدة فقط وهي لغة الآلة و بالتالي يجب تحويل الشيفرة المصدرية CODE المكتوب بأحد اللغات عالية المستوى إلى لغة الآلة و هذا يتم بطريقتين رئيسيتين

: وفيها يتم تحويل الشيفرة المصدرية كاملة إلى لغة الآلة مباشرة بواسطة برنامج يسمى المترجم **Compiler** مثل لغة C، C++.

: ويتم فها تنفيذ البرنامج سطر بسطر و تنفيذ كل أمر لوحده و هنا يستعمل برنامج يسمى المفسر **Interpreter** مثل لغة Python، PHP، Javascript.

Python

Python هي لغة برمجة، عالية المستوى سهلة التعلم مفتوحة المصدر قابلة للتوسيع، تعتمد أسلوب البرمجة الـغرضية التوجه.

لغة بايثون هي لغة مفسّرة، ومتعددة الاستخدامات، وتستخدم بشكل واسع في العديد من المجالات، كبناء البرامج المستقلة باستخدام الواجهات الرسومية وفي تطبيقات الويب وتطبيقات الذكاء الاصطناعي.

Python

يمكن استخدام بايثون لعمل البرامج البسيطة للمبتدئين، ولإنجاز المشاريع الضخمة في الوقت نفسه. غالباً ما ينصح المبتدئون في ميدان البرمجة بتعلم هذه اللغة لأنها من بين أسرع اللغات البرمجية تعلمًا.

تستخدم بايثون الإزاحة بالفراغات بدلاً من الأقواس المعقوفة {}, و بالتالي فإن البرنامج المكتوب بلغة Python يكون سهل القراءة و الفهم بشكل كبير.

لماذا بايثون؟

تعمل Python على أنظمة تشغيل مختلفة (Windows و Mac و Linux ...).

تحتوي لغة بايثون على بناء جملة بسيط مشابه للغة الإنجليزية.

تحتوي Python على بناء جملة يسمح للمطورين بكتابة برامج ذات سطور أقل

من بعض لغات البرمجة الأخرى.

لماذا بايثون؟

تعمل Python على نظام مترجم ، مما يعني أنه يمكن تنفيذ التعليمات البرمجية

بمجرد كتابتها. هذا يعني أنه يمكن بناء نماذج أولية بسرعة كبيرة.

يمكن التعامل مع بايثون بطريقة إجرائية أو طريقة موجهة للكائنات أو بطريقة

وظيفية.

إعداد بيئة العمل

- 1- تنصيب مفسر اللغة بايثون: قبل الشروع في كتابة برنامج بلغة البايثون يجب تنصيب مفسر اللغة البايثون على الجهاز الذي نعمل عليه، يمكن الحصول على مفسر لغة البايثون من <https://www.python.org>.
- 2- تنصيب IDE: يمكن كتابة برامج بلغة البايثون بأية برنامج محرر نصوص و لكن سنستعمل محرر Visual Studio Code vscode.
- 3- كتابة أول برنامج بلغة البايثون.

Python - Basic Syntax

تعرف Python مجموعة من القواعد تستعمل لكتابة البرنامج، و على الرغم من أنها لديها بعض الشبه مع بعض اللغات الأخرى ك JAVA ، C ولكن هنالك تختلف بشكل كبير عن هذه اللغات.

المعرفات Identifier

(المعرف) Identifier: هو اسم يستخدم لتعريف متتحول أو تابع صنف ...، والـ Identifier يجب أن يبدأ بالأحرف من A-Z أو a-z أو _، متبوعاً بصفر أو أكثر من الحروف والأرقام والـ _ (شرط سفلية).

لا تسمح Python باستخدام أحرف الترقيم مثل @ و \$ و % داخل المعرفات.

لا يمكن استخدام الكلمات المفتاحية لبايثون كمعرفات.

لغة بايثون حساسة لحالة الأحرف و بالتالي Cat, cat معرفان مختلفان.

Identifier

أسماء معرفات صحيحة	أسماء معرفات غير صحيحة
score	@core
return_value	Return
highest_score	highest score
name1	1name
convert_to_string	convert to_string

Identifier

يجب دائماً اعطاء المعرفات أسماء معبرة، مما يساعد في فهم البرنامج وسهولة صيانته.

صحيح
`c=10`

أكثـر منطقـية، وأكـثر لـلـفـهـمـ، اـسـهـلـ لـمـعـرـفـةـ ماـذـاـ يـمـثـلـ
`count=10`

Python Reserved Words

القائمة التالية تتضمن كلمات بايثون الأساسية، هذه كلمات ممحوزة ولا يمكنك استخدامها كثابت أو متغير أو أي أسماء معرفات أخرى.

جميع الكلمات المفتاحية لـ Python تحتوي على حرف صغيرة فقط.

Python Reserved Words

and	as	assert	break	class	continue
def	del	elif	else	except	False
finally	for	from	global	if	import
in	is	lambda	None	nonlocal	not
or	pass	raise	return	True	try
while	with	yield			

Indentation in Python

```
j = 1
while(j <= 5):
    print(j)
    j = j + 1
```

لا تستعمل باليثون الاقواس لتحديد كتل التعليمات عند تعريف التوابع أو عند التحكم في مسار البرنامج .

تستعمل باليثون المسافات البدائة للإشارة إلى كتل التعليمات, حيث أن جميع التعليمات التي تنتمي إلى نفس الكتلة البرمجية يجب أن يكون لها نفس المسافات البدائة.

Comments

التعليقات: هي شرح أو تعلیقات توضیحیة يمكن إضافتها إلى البرنامج.
يتم إضافتها بغرض تسهیل فهم الكود المصدري على البشر ، ويتم تجاهلها بواسطة مترجم بايثون. تعزز التعليقات قابلیة قراءة الكود وتساعد المبرمجین على فهم الكود بسهولة فائقة ، إضافة شرح عن البرنامج.

Comments

تعليقات السطر واحد: حيث يستعمل الرمز # للإشارة على بدء التعليق و اعتبار كل النص الذي يأتي بعده هو تعليق يتم تجاهله من المترجم حتى نهاية السطر.

```
6
7 # This is one line comment, and will be ignored by interpreter.
8
9 name='jone' # all text after hash sign will be ignored
```

Comments

```
12      #this is comment line 1.  
13      #this is comment line 2.  
14      #this is comment line 3.  
15      #this is comment line 4.  
16  
17  
18      '''thi is multi line comment  
19          using triple-qouted string  
20          python intepreter will ignore it'''
```

تعليقات عدة أسطر : يمكن استعمال الهاش # عند بداية كل سطر من التعليق.

أو استعمال نص مع ثلاث علامات تنصيص قبله وبعده حيث سيتم تجاهله من قبل المفسر أيضاً.

المتحولات Variables

```
24 counter=100 # define integer variable
25 miles=10.0 #define floating variable
26 greeting ='Hellow World' #define string variable.
```

- المتغيرات هي أمكان في الذاكرة يتم تخزين البيانات فيها أثناء تشغيل البرنامج.

لكل متغير اسم خاص يتم تحديده باستخدام قواعد المعرفات المذكور سابقا.

يتم اسناد قيم على المتغيرات باستعمال علامة المساواة = .

يتم تحديد نوع بيانات المتغير تبعاً لنوع البيانات التي تم تخزينها فيه.

المتحولات Variables

تسمح بـايثون بإسناد قيم واحدة إلى عدة متغيرات بنفس الوقت

```
28     a = b = c =100
```

كما تسمح بـايثون بإسناد عدة قيم لعدة متغيرات باستعمال تعليمة واحدة

```
name ,age ,weight = 'jone' , 10 ,70.5
```

المتحولات Variables

لمعرفة نوع المتتحول يمكن استخدام الدالة type()

```
39 name='my name'  
40 type(name)
```

يمكن حذف متتحول باستخدام التعلیم del

```
42 del(name)
```

تمت المحاضرة وبحمد الله