لغة البايثون

هي لغة برمجة متعددة الاستعمال ، لغة تنفيذية ، تدعم البرمجة الشيئية

بعض مجالات استخدام لغة البايثون

برمجة الويب

تعلم الالة او الذكاء الاصطناعي

علم البيانات و تمثيل البيانات

متطلبات سابقة

\_\_\_التفكير بالبرمجة

لماذا بايثون ؟

تحتوي ادوات رياضية متقدمة مثل المجموعات والعمليات عليها مثل الاتحاد و ايضا تدعم الاعداد التخيلية

تعمل على كثير من المنصات

Windows | Linux | Mac os | Raspberry Pi

مع انها من اقوى اللغات ولكنها سهلة التعلم وينصح البدء منها

تنوية هذه الدرس يستخدم الاصدار الثالث من بايثون

بمكنك استخدام اي IDE تريد ويمكك تجريب امثلتنا على اللنك تحت المثال حيث سيقوم بارسالك الى صفحة ويب تحتوي على Compiler بسيط

من الجيد معرفة

**بايثون لغه لا تستخم الفاصلة المنقوطة لانهاء امر ففي بايثون كل سطر جديد يعتبر امر جديد**

مثال

Java

Int a =6 ; double c = 1.8;

Python

a=5

c=1.8

لا تشغل بالك باي فرق في المثال السابق الا فرق انهاء الامر فاغلب اللغات تستخدم الفاصلة المنقطوه وبهذا تسمح بوضع اكثر من امر على نفس السطر اما بايثون فانها تسمح فقط بامر واحد على كل سطر بدون فاصة منقوطة

بايثون لغة تميز مجموعات الاوامر المتداخلة من خلال المسافة على عكس اللغات الاخرى التي تميزها من خلال {}

القوس المتعرج

مثال

Java

While(true){

If (true){

}

}

Python

While true :

If true:

Statements inside if

نرى من هذه المثال ان بايثون اعتمد في التفريق بين الاوامر خارج امر while والاوامر داخلها على المسافة بينما اعتمدت اللغة الاخرى على القوسين المتعرجيين

امثلة اخرى متنوعة

**الملاحظات**

يمكنك كتابة الملاحظات بلغة بايثون باستخدام الرمز #

#this is a comment

ولكن انتبة ان بايثون عند تنفيذة للبرنامج اذا راه اول السطر اشاره # سيقوم بتخطي السطر كامل ولا تحاول ان تكتب ملاحظ على سطرين الا اذا كان السطران يبدان ب#

#this is long comment

It comes with two lines

خطا

#this is long comment

#it comes with two lines

صحيح

مثال اخر

X= 5 #this is a variable

في هذه المثال وعند تنفيذ الجملة الاعلى ستقوم اللغة بتنفيذ السطر الى ان تصل الى اشاره # فتقوم بتخطي السطر

X=5 #b=6

هنا ستقوم اللغة بتفيذ الجملة x=5

وستتخطى الجملة

B=6

ولو كانت الاشاره # غير موجوده ستكون الجملة اعلاه جملة خاطئه حيث وكما ذكرنا سابقا ان لغة بايثون تنفذ كل امر على سطر منفرد

**المتغيرات**

لغة بايثون هل لغة داناميكية اي انهو لا يتوجب عليك ذكر نوع المتغير عند تعينة

مثال

Java

Int x =25;

Python

X =25

نلاحظ مما سبق ان اللغة الاولى ليست لغة داناميكيه اي انها احتاجت منك كلمة مفتاحية للعبير عن نوع المتغير اما لغة بايثون فيها اذكى وتستطيع معرفة نوع المتغير لوحدها

تعيين متغير

Var\_Name = “Some\_Value”

انواع المتغيرات في بايثون

**رقم صحيح**

مثال

X=6

**رقم عشري**

مثال

X= 6.5

**رقم تخيولي**

مثال

X=2+5j

اذا كنت لا تعلم ما هو الرقم التخيلي فلا داعي للقلك فهو مصطلع رياضي وبما انك لا تعرفه فلن تحتاجه

رمز

هوا كل ما يحيط به “ “ او ‘ ‘

مثال

X= “String”

او

X= ‘string’

ما الفرق بين

X=5

و

X=”5”

في الامر الاول قام بايثون بتخزين رقم 5 على انه رقم حيث انه يفهم معناه ويمكنك القيام بجميع عمليات الرياضيات عليه اما في المثال الثاني فعندما راه البرنامج الاشارتان "" المحيطتان بالرقم 5 فهو لم ينضر الى المحتوى ولم يحاول فهمه فقط قام بتخزينه على انه من نوع اشاره واذا حاولت استخدام اي من عمليات الرياضيات عليه فيسيظهر خطا لان يعلم ان هذه المتغير ليس من نوع رقم

ملاحظة يدعم بايثون الرمز متعدد الاسطر باستخدام ثلاث اشارات """ mulitline string"""

مثال

“””line one

Line two

Line three”””

شروط اختيار اسم لمتغير

* يجب ان يبدا اسم المتغير حرف او \_
* لا يمكن لاسم المتغير ان يبدا برقم
* تنحصر المقاطع المسموحة عن اختيار الاسم من
* (A-Z) , (a-z) , (0-9)
* اسمع المتغير "Case-sensetive"

**التحويل من متغير الى اخر**

يمكنك التحول من نوع من المتغيرات الى نوع اخر في لغة بايثون من خلا بعض الاقترانات "function" التي توفرها اللغة مثال :

X=int(y)

X = int(3.7)

الاقترانات المتوفرة

Int(x)

Str(x)

Float(x)

مثال غلى كل وحدة

مع العلم ان لغة بايثون تقوم بالتحويل التلقائي للمتغيرات فان في بعض الاحيان سؤدي هذا الى خلل في البرنامج لذا يجب تجنبة واستخدام الاقترانات اعلاه

**المجموعات Arrays**

**تحتوي لغة بايثون على اربع انواع من المجموعات كل منها له خصائص مختلفة و يستخدم بمجالات مختلفة**

* **النوع الاول : lists**

**هذا النوع مرتب (اي يكون لكل عنصر ترتيب محدد) ويمكن التعديل علية بالاضافه او الازالة ويسمح ايضا بتكرار العناصر**

* **النوع الثاني : tuple**

**هذا النوع مرتب ولكنة ثابت اي انه لا يمكن الاضافه عليه او التغير على عناصرة ويسمح ايضا بتكرار العناصر**

* **النوع الثالث: sets**

**هذا النوع غير مرتب اي ان لا يجد لكل عنصر ترتيب معين و يمكن التغيير عليه ولا تسمح بتكرار العناصر**

* **النوع الرابع: dictionary**

**النوع الأول Lists**

هي نوع من الجمعات يكون في كل عنصر فيها ترتيب من المجموعه ويمكن التعديل عليها اما بالاضافه او الازالة او التغيير

نستخم فيها هذه النوع من الاقواس

[]

لتميزها عن غيرها كما يلي

thislist = [“ahmad”,”mohammad”,”salem”]

في هذه المثال قمنا بتعين مجموعة من النوع الأول ووضعنا في ثلاث أسماء لكل اسم ترتيب حيث ان احمد هو الاسم الأول وترتيبة صفر

انتبة =============

**استخراج القيم**

اذا اردنا ان نقوم باستخراج اسم محمد فترتيبة الثاني ولكن كما قلنا لان العد يبدا من الصفر فيكون ترتيبة الأول ونستخرجة كما يلي

print(thislist[1])

**تغيير القيم**

اذا اردت تغير قيمة من مجموعة كل ما تحتاجو هو رقم العنصر كما يلي

thislist[1] = "blackcurrant"

**الدوران على جميع عناصر المجموعة**

في كثير من الأحيان ستحتاج للدوران على جميع عناصر المجموعة لتغيرها او طباعتها او زيادتها بقيمة معينه

تمكنك لغة بايثون بالقيام بهذه بطريقة سهلة جدا باستخدام جملة

For

كما يلي

thislist = ["apple", "banana", "cherry"]  
for x in thislist:  
  print(x)

انتبة

**البحث في المجموعة عن عنصر**

في كثر من الأحيان أيضا ستحتاج الي البحث عن عناصر في مجموعات كبيره عاده ما نقوم بالدوران على المجموعة واستخدام جملة الشرط كما يلي

مثااااااااال

ولكن لغة باثون تحتوي على طريقة اسهل جدا ولا نحتاج لاكثر من سطر واحد باستخدام جملة السطر فقط كما يلي

thislist = ["apple", "banana", "cherry"]  
if "apple" in thislist:  
  print("Yes, 'apple' is in the fruits list")

**إيجاد طول المجموعة**

لمعلافة كم عنصر تحتوى المجموعة يمكنك استخدام اقتران الطول

Len()

يتمر تمرير المجموعة المراد معرفة حجمها للاقتران ويرجع قيمة حجمها بالارقم كما يلي

thislist = ["apple", "banana", "cherry"]  
print(len(thislist))

تنبية عن انو 3 مش 2

**إضافة عناصر**

يمكنك إضافة عنصر في اخر المجموعة باستخدام ال

Append() method

كما يلي

thislist = ["apple", "banana", "cherry"]  
thislist.append("orange")  
print(thislist)

يمكنك أيضا إضافة عنصر في مكان محدد بتحديد رقم المكان باستخدام

Insert() method

كما يلي

thislist = ["apple", "banana", "cherry"]  
thislist.insert(1, "orange")  
print(thislist)

حيث انها تأخذ تأخذ رقم المكان وبعدها العنصر المراد اضافتة وتقوم بازاحه جميع العناصر التي بعده بمقار 1

**إزالة عناصر**

يمكنك ازلة عنصر من مجموعة بعدة طرق

الطريقة الأولى باستخدام العنصر نفسة مثلا اذا كانت مجموعة تحتوى على اسم الطلاب في صف معين واراد محمد الانتقال فيمكنك حذفه باستخدام اسمه وهذه باستخدام ال

Remove() Method

حيث انها تأخذ العنصر المراد حذفة كما يلي

thislist = ["apple", "banana", "cherry"]  
thislist.remove("banana")  
print(thislist)

طرقة أخرى باستخدام رقم ترتيب العنصر من خلال

Pop() method

كما يلي

thislist = ["apple", "banana", "cherry"]  
thislist.pop()  
print(thislist)

ملاحظة اذا تركت فاضية

يمكنك أيضا افراغ المجموعة او مسح جميع عناصرها باستخدام ال

Clear() method

كما يلي

thislist = ["apple", "banana", "cherry"]  
thislist.clear()  
print(thislist)

ملاحظة عن انها اكثر اشي مهم

جدول أشياء اخرى

**النوع الثاني Tuples**

هي نوع من الجمعات يكون لكل عنصر فيها ترتيب من المجموعه ولاكن لا يمكن التعديل عليها لا بالحذف ولا الإضافة ولا التغيير فعناصرها ثابته

نستخدم فيها هذه النوع من الاقواس

()

لتميزها عن غيرها كما يلي

thistuple = ("apple", "banana", "cherry")

في هذه المثال قمنا بتعين مجموعة من النوع الثاني ووضعنا في ثلاث أسماء لكل اسم ترتيب حيث ان احمد هو الاسم الأول وترتيبة صفر

كما في النوع الأول

**استخراج القيم**

اذا اردنا ان نقوم باستخراج اسم محمد فترتيبة الثاني ولكن كما قلنا لان العد يبدا من الصفر فيكون ترتيبة الأول ونستخرجة كما يلي

thistuple = ("apple", "banana", "cherry")  
print(thistuple[1])

**تغيير القيم**

كما ذكرا سابقا فان هذه النوع لا يسمح بالتعديل على محتواه ولو حاولنا فانها لا تتغير كما في المثال

thistuple = ("apple", "banana", "cherry")  
thistuple[1] = "blackcurrant"  
# The values will remain the same:  
print(thistuple)

**الدوران على جميع عناصر المجموعة**

في كثير من الأحيان ستحتاج للدوران على جميع عناصر المجموعة لتغيرها او طباعتها او زيادتها بقيمة معينه

تمكنك لغة بايثون بالقيام بهذه بطريقة سهلة جدا باستخدام جملة

For

كما يلي

thistuple = ("apple", "banana", "cherry")  
for x in thislist:  
  print(x)

انتبة عشان التبعيد

**البحث في المجموعة عن عنصر**

في كثر من الأحيان أيضا ستحتاج الي البحث عن عناصر في مجموعات كبيره عاده ما نقوم بالدوران على المجموعة واستخدام جملة الشرط كما يلي

مثااااااااال

ولكن لغة باثون تحتوي على طريقة اسهل جدا ولا نحتاج لاكثر من سطر واحد باستخدام جملة السطر فقط كما يلي

thistuple = ("apple", "banana", "cherry")  
if "apple" in thislist:  
  print("Yes, 'apple' is in the fruits list")

**إيجاد طول المجموعة**

لمعلافة كم عنصر تحتوى المجموعة يمكنك استخدام اقتران الطول

Len()

يتمر تمرير المجموعة المراد معرفة حجمها للاقتران ويرجع قيمة حجمها بالارقم كما يلي

thistuple = ("apple", "banana", "cherry")  
print(len(thislist))

تنبية عن انو 3 مش 2

**إزالة عناصر**

مع ان هذه النوع يمنع إزالة او التعديل او الإضافة على عناصرة ولكن يمككنا انو نقوم بحذ المجموعة كامله عن طريق استخدام جملة

Del

فهي تقوم بمسح المتغير كامل من الذاكرة كما يلي

thistuple = ("apple", "banana", "cherry")  
del thistuple  
print(thistuple) #this will raise an error because the tuple no longer exists

هذه الجملة سوف تنتج خطا لانه لم يعد هناك متغير بهذه الاسم بعد ان قمنا بمسحة

جدول