

PYTHON COURSE

Python course , from scratch
to professionalism ...

Made by : Taha Khaldoun Amro

For loop

سنتكلم الان عن أحد أهم المفاهيم البرمجية المتعلقة بالحلقات التكرارية , ألا وهو الfor loop .

حيث يمكن التعبير عن هذه الحلقة بأنها عملية تكرارية تتم على عناصر كائن iterable حسب الصيغة الآتية :

for item in iterable_object :

do something with the item

حيث أن:

1.item هو متغير تقوم أنت بانشائه و تستدعيه داخل الfor loop متى ما تشاء داخل الloop.

2.يتم تعريف item بشكل تلقائي بواسطة الfor loop على أنه موقع العنصر الحالي داخل الiterable , و سيتم تكرار الحلقة على كل عناصر الiterable .

3.الiterable_object هو عبارة عن كائن يحتوي داخله على عناصر -> [list , string , tuple , dictionaries , set]

سنقوم الان بتطبيق عملي , لنفهم ما هو مكتوب في الأعلى .

For loop

سنقوم ببساطة بعملية بسيطة , حيث سننشئ list تحتوي على مجموعة ارقام , و نقوم بطباعتها , أو القيام ببعض العمليات عليها :

```
numbers = [1,2,3,4,5,6]
```

```
for number in numbers :  
    print(number)
```

بهذه البساطة يتم الأمر , جرب الان طباعة الأعداد مضروبة في أي عدد , أو مرفوعة لأي قوة , أو جرب القيام بأي عملية عليها و طباعة ناتج العملية .

سنجرب الان الانتقال الى المستوى التالي , حيث سنعمل على التفريق بين ما اذا كان العدد عدداً فردياً أو زوجياً .

For loop

سنبدأ بإنشاء list بالأعداد من 1 إلى 10 , و سنقوم بعمل loop عليها , بحيث نحدد باستخدام الجمل الشرطية , ما اذا كان العدد فرديا او لا , حيث ان الاعداد الزوجية يكون باقي قسمتها 0 و الاعداد الفردية لا يكون باقي قسمتها 0 , وهذا هو المبدأ الرياضي الذي سنعمل عليه :

```
numbers = [1,2,3,4,5,6,7,8,9,10]
```

```
for number in numbers :
```

```
    if number % 2 == 0:
```

```
        print(f"the number {number} is even")
```

```
    else :
```

```
        print(f"the number {number} is odd")
```

و بكل بساطة , هذا هو كل ما في الأمر .

For loop

ما هي البيانات التي يمكنك عمل ال for loop عليها في لغة بايثون ؟
كنا قد ذكرنا سابقاً أنه يجب أن يكون نوع البيانات iterable حيث أننا لا يمكننا أن نعمل loop على integer لأنه ببساطة ليس iterable , و فيما يخص البيانات التي يمكنك أن تقوم بال loop عليها فهي :

1.strings

2.lists

3.tuples

4.dictionaries

5.sets

6.range function

تكلّمنا فيما سبق عن كل أنواع البيانات السابقة , ما عدا الأخيرة , range function هو اقتران مدمج باللغة , يعمل على انشاء list بأعداد من نقطة بداية الى نقطة نهاية , حيث يمكن استخدامه بشكل كبير جداً في عمليات ال for loop .

سنقوم الان بتصميم برنامج ظريف للتدرب على الموضوع بشكل أكبر .

For loop

هل تذكرون الكود من الدرس السابق , الكود الذي قمنا من خلاله بطباعة أسماء الاصدقاء بعد ادخالهم من قبل المستخدم , الان و قد تعلمنا ال for loop فان الكود السابق يمكن اختصاره .

في ال loop الأولى , يمكننا بكل بساطة استبدال الحلقة التكرارية while ب for loop تقوم بالعملية بعدد مرات محدد , حيث عدد المرات هو عدد الاصدقاء , لذلك سنقوم باستخدام ال range function و الذي سنعمل من خلاله على وضع حد ألا و هو المتغير (عدد الأصدقاء) الذي يقوم المستخدم بادخاله .

```
myfriends = [] # our empty list of friends
number_of_friends = int(input("please , how many friends do you have ? "))
for friend_number in range(number_of_friends) : # if counter == number_of_friends : loop is over
    friend = input(f'enter your #{friend_number + 1} friend : ').capitalize().strip() # friend name

    myfriends.append(friend) # add the friend to the list we made

print(myfriends) # test our work
```

For loop

و الان و كشرح للكود السابق , قد قمنا بالخطوات التالية :

1. الابقاء على المتغيرات الخاصة بقائمة الاصدقاء و عددهم .

2. ازالة المتغير counter لأن الfor loop باستخدام الrange function توفر لنا عدداً افتراضياً تلقائياً .

3. عمل الfor loop عرفنا داخلها متغيراً سميناه friend_number , و ذلك لأنه مع تكرار الحلقة سيأخذ قيمة العدد الذي نقف عليه , ابتداءً من ال0 و انتهاءً بعدد الاصدقاء الذي تم ادخاله - 1 (last index not included)

4. في تعريفنا للrange function في الfor loop , قمنا بوضع قيمة الانتهاء , حيث أن الrange function تقبل قيمتين , قيمة البداية و قيمة النهاية , و لكن عندما نضع قيمة واحدة , فان النظام يفهم أنها قيمة النهاية , و يبدأ من العنصر 0 .

5. في داخل الfor loop قمنا بعملية ادخال اسم الصديق مع رسالة تفيد برقمه , حيث أن رقمه , هو رقم المتغير friend_number + 1 لأن المتغير السابق سيتغير تلقائياً مع تكرار الحلقة .

6. بعدها أضفنا الصديق الى القائمة , تماماً كما كنا قد فعلنا من قبل في الكود السابق .

For loop

والان , و استكمالا لعملنا على الكود , فان المهمة التالية الموكلة الينا في الكود , هي عملية طباعة الاصدقاء , مع رقمهم في رسائل منسقة , و ذلك عبر for loop تتم على عناصر ال friends list ,

```
for friend in myfriends: # for loop on the elements of the list
```

```
friend_number = myfriends.index(friend) # the friend number by the index method
```

```
print(f'#{friend_number + 1} {friend}') # calling the friend by the variable
```

```
print('loop is over') # tells us that the loop is done
```

بالنسبة لما تم داخل ال for loop فهو ببساطة عملية ايجاد رقم العنصر (friend) باستخدام ال index method , و من ثم استخدام المتغير friend number من اجل طباعة رقم الصديق , حيث اننا يجب ان نضيف 1 اليه , لان ال index يبدأ من 0 , بطبيعة الحال يمكنك استخدام counter هنا بدلا من عملية ايجاد رقم العنصر حيث يبدأ العداد من 1 , وهكذا لن تضطر الى زيادة 1 , ولكننا استخدمنا هذه الطريقة استذكارا للمعلومات التي كنا قد درسناها .

Nested for loop

في هذه الجزئية من درس ال for loop سنتكلم عن الحلقات التكرارية المضمنة , و التي تعني وجود حلقة داخل حلقة , تكملان بعضهما البعض , و سنتعلم كيفية استخدامها عبر تمرين بسيط , و لكنه بحاجة الى بعض التركيز .

سنعمل في هذا التدريب على القيام بالمهمة التالية :
انشاء قاموس باسماء ثلاث أشخاص , حيث يكون اسم الشخص هو المفتاح , و تكون قيمته قاموساً اخر , يحتوي بداخله على مفتاح باسم skills و الذي بدوره يحمل قيمة نعبر عنها ب list من تخصصاته البرمجية .
سيطلب منا في هذا التدريب , طباعة اسم الشخص في سطر , ثم طباعة تخصصاته .

Nested for loop

```
people = {  
    "Ahmed" : {  
        'skills' : ['python' , 'Css'] ,  
    } ,  
    "Saeed" : {  
        'skills' : ['Html' , 'Css'] ,  
    } ,  
    "Sawsan" : {  
        'skills' : ['Html' , 'C++' 'Css' , 'JS'] ,  
    }  
}
```

بهذه الطريقة نكون قد عرفنا القاموس الذي يحتوي بداخله على القواميس المضمنة .
سنبدأ الآن بال for loop الرئيسية و التي ستعمل على طباعة المفاتيح في القاموس (أسماء الأشخاص) .

Nested for loop

```
for name in people :
```

```
    print(name)
```

ببساطة , ما قمنا به الان هو طباعة المفاتيح , حيث ان العناصر التي تتم عليها الحلقة , هي اسماء المفاتيح في القواميس , و الان نحن نريد الوصول الى التخصصات الخاصة بكل شخص منهم .
لو تذكرنا دروس القواميس , فاننا نصل عنصر داخل القاموس من خلال ال key الخاص به اذا فان الموضوع سهل , ما عدا اننا سنقوم بالدخول الى قاموس داخل قاموس , لذا فاننا بحاجة وضع اسم القاموس الكبير , ثم اسم القاموس الصغير (اسم الشخص) ثم ال key الخاص بالتخصصات , الا وهو skills .

```
print(people[name]['skills'])
```

المشكلة الوحيدة هنا ان طباعة الموضوع بهذا الشكل ليس عمليا , فشكل القائمة في الطباعة ليس جميلا , لذا فنحن سنقوم بعمل for loop على عناصر القائمة التي تتضمن ال skills , و هذه ستكون ال for loop المضمنة .

Nested for loop

```
for skill in people[name]['skills'] :  
  
    print(f"- {skill}")
```

اضف الحلقة التكرارية الاتية داخل الحلقة التكرارية الكبرى , و ذلك لتحصل على النتيجة النهائية .

ازل عملية طباعة القائمة التي كنا قد وضعناها سابقا للتوضيح , و قم باضافة تنسيقات على النصوص بحيث يصبح الأمر على شكل رسالة ظريفة .

سنتكلم عن هذا الموضوع في وقت لاحق في الدورة , و لكننا نكتفي بهذا القدر الى الان .

For loop

مرفق لكم روابط شرح لما جاء في الدرس :

<https://youtu.be/4YolrVX6f1Q?si=QLWe9427Z5HjNgXa>

https://youtu.be/9JJDDKj_tGA?si=gQcfcLeYd8ycfOl1

https://youtu.be/x_GyjV2Nb6k?si=ITmr-xdLJqm4GIQh