

# PYTHON COURSE

Python course , from scratch  
to professionalism ...

Made by : Taha Khaldoun Amro

## While loop

سنتكلم الان عن أحد أهم الفاهيم في عالم البرمجة , ألا و هو مفهوم ال loop أو الحلقة التكرارية , حيث يعبر هذا المفهوم عن عملية اختصار تكرار الكود أكثر من مرة , و القيام به بشكل مكرر , حيث يتم بذلك اختصار الوقت و الجهد في صناعة الكود .

في لغة بايثون , و كما هو الحال في كل لغات البرمجة , يتم صياغة عملية ال while loop بالصيغة العامة الآتية :

```
while condition_is_True :
```

```
do the code here
```

نلاحظ أن العملية بحاجة الى كلمة مفتاحية لبدءها , ألا وهي كلمة while , ما يوضع بعد كلمة while هو شرط , حيث يعتمد مبدأ عمل الحلقة على كون الشرط True , كيف ؟

ما تكلمنا سابقاً عن القيم المنطقية , فان أي شكل من البيانات له قيمة منطقية , و هكذا الأمر هنا , و لكن عندما تتغير قيمة الشرط الى False تتوقف الحلقة التكرارية , و سنتكلم عن هذا الأمر لاحقاً .

تخضع ال while loop الى قواعد ال indentation أي أن الكود تحتها يجب أن يكون موضوعاً على بعد tab من بداية السطر كما هو الحال في الجمل الشرطية .

و الان سنبدأ بالتعرف على عمل الحلقات التكرارية .

## While loop

سنبدأ الآن بتمرين بسيط , ألا وهو طباعة شيء دون توقف , أو بالأحرى unbrakeable loop أو الحلقات المستمرة , و هي الحلقات التي لن تنتهي الا في حالة ايقاف الterminal :

```
a = 10
```

```
while a < 20 :  
    print(a)
```

كما رأينا في الكود السابق , فان العملية لم تتوقف , لأن المتغير a أصغر من 10 , وهو لا يتغير طوال الكود , و لهذا السبب فان جواب الشرط يبقا True الى الأبد .

و الآن سوف نعمل على التحكم في عدد مرات تكرار الحلقة , حيث يمكننا بكل بساطة , زيادة قيمة a في كل مرة تتم في الloop , حيث سيعمل الكود على فحص ما اذا كانت a أقل من 20 و في اللحظة التي تصل فيها a الى 20 بفعل كود الزيادة , سيتوقف الكود , دون طباعة القيمة 20 , و ذلك ببساطة لأنها ليست أقل من 20 بل تساويها .

## While loop

```
a = 10

while a < 20 :
    print(a)
    a += 1 # the counter
```

بهذه البساطة تتم العملية , فالموضوع ليس معقداً , و يمكننا أيضا اضافة رسالة تعلمنا بانتهاء ال loop بعد ال loop داخل الكود , وذلك ببساطة لأن ال loop توقف عمل الكود حتى تنتهي , و بمجرد انتهائها فان الكود يستمر في العمل .

```
a = 10

while a < 20 :
    print(a)
    a += 1 # the counter
print('Loop is over')
```

## While loop

عندما نريد اضافة كود ليتم تنفيذه بعد انتهاء ال loop , علينا أن لا نشمله بال indentation الذي تركناه قبل الكود داخل ال loop .

و الان , و بعد أن تمكنا من فهم مبدأ عمل ال loop , سننتقل الى بعض التطبيقات العملية عليها , و سنعمل على بناء مجموعة من التطبيقات التي ستختصر ال loop فيها علينا الكثير .

## Print my friends !

في هذا التمرين , لديك list تحتوي على اسماء أصدقائك , طلب منك بكل بساطة أن تقوم بطباعتهم جميعاً , و أن تقوم بطباعتهم مرقمين كالتالي :

```
#1 Ahmed  
#2 Samer  
#3 Wa'el  
>>>
```

سنبدأ بالطبع الان بتعريف قائمة الأصدقاء , و الذين سوف يتم ادخالهم من خلال المستخدم , اذا فانا بحاجة الى تكرار عملية ادخال , و لكن في البداية , سنقوم بسؤال المستخدم , عن عدد أصدقائه الذين يريد ادخالهم , و سنكرر ال loop على عدد الأصدقاء , حيث سيعمل الكود داخل ال loop على طلب ادخال اسم الصديق , ثم اضافته الى ال list .

سيتم عمل بعض التغييرات على اسم الصديق قبل ان يتم ادخاله في ال list مثل جعله خاليا من ال spaces و جعل أول حرف منه capital .

## Print my friends !

```
myfriends = [] # our empty list of friends

counter = 0 # the counter to count the number of friends

number_of_friends = int(input("please , how many friends do you have ? "))
while counter < number_of_friends : # if counter == number_of_friends : loop is over

    friend = input(f'enter your #{counter + 1} friend : ').capitalize().strip() # friend name

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

    counter += 1 # increase the counter

print(myfriends) # test our work
```

## Print my friends !

شرح للكود السابق :

في البداية انشأنا list فارغة للتعبير عن مجموعة أصدقاءنا , حيث سيقوم المستخدم بتعبئتها فيما بعد .

ثم , و في سبيل جعل ال loop - التي سوف تتم من أجل ادخال الاصدقاء- تنتهي , فاننا قمنا بعمل counter يقوم بعد مرات تكرار ال loop و قمنا في نهاية كل تكرار لل loop بزيادته بمقدار واحد .

ثم عرفنا المتغير الذي سيحدد المستخدم من خلاله عدد أصدقائه , و الذي هو منطقيا عدد مرات تكرار ال loop , لأنه سيقوم بادخال العدد الذي أدخله من الاصدقاء , و بذلك يتم تكرار ال code داخل ال loop على عدد الأصدقاء .

قمنا بعد ذلك ببدء loop يكون شرطها أن تعمل الى أن يساوي العداد عدد الأصدقاء , و من ثم كسر ال loop و ايقافها .

داخل ال loop , يعمل المتغير الأول على طلب ادخال اسم الصديق , من خلال رسالة بسيطة تخبره بادخال الصديق رقم س حيث يتم تعريف س على انها رقم العداد + 1 و ذلك لان العداد يبدأ من العدد 0 , و نحن سنطلب من المستخدم ادخال الصديق 1 ثم 2 ثم 3 و هكذا , و تعمل عملية الادخال أيضا على وضع التنسيقات المطلوبة سابقاً على الاسم المدخل .

و من ثم يتم ادخال الاسم المدخل في list الأصدقاء بكل بساطة .

يتم في نهاية ال loop زيادة عدد ال counter حتى يتم ايقاف ال loop عندما يصل الى عدد الأصدقاء .

و في النهاية يتم طباعة ال list للتأكد من صحة الكود .



## Print my friends !

سنعمل الان على عملية طباعة الاصدقاء واحدا تلو الآخر , و ذلك من خلال نفس ال loop السابقة , ولكن مع استبدال الكود داخلها , بعملية طباعة نص منسق , يخبرنا برقم الصديق و اسمه كما وضعنا سابقا :

```
counter = 0 # make the counter 0 again to reuse it

while counter < number_of_friends : # the same loop as before

    print(f'#{counter + 1} {myfriends[counter]}') # calling the friend by index

    counter += 1 # increase the counter

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

و بكل بساطة , تمكنا الان من انتهاء المهمة .

## While loop

مرفق لكم فيديوهات شرح لما جاء في الدرس :

<https://youtu.be/A0oBGPSUbel?si=mwJJaiMMoYD9Mpg->

بعض التدريبات الخارجية التي ستساعدكم على فهم الموضوع بشكل أكبر :

<https://youtu.be/9rU2fImqSR4?si=2KMDCh5rCF8C0pAz>

[https://youtu.be/jRGJjckgSlA?si=hl\\_YBobYbs8tfHx6](https://youtu.be/jRGJjckgSlA?si=hl_YBobYbs8tfHx6)

[https://youtu.be/7NIcsmfHIrg?si=Te30NI7qWW3W\\_jO](https://youtu.be/7NIcsmfHIrg?si=Te30NI7qWW3W_jO)