

الوحدة 03: التعليمات التكرارية في لغة البايثون

المحاضرة 05:

التعليمات التكرارية for / while في بايثون

التعليمات التكرارية (الحلقة) هي طريقة في البرمجة لتكرار تنفيذ مجموعة من التعليمات أكثر من مرة طالما أن شرط معين صحيح. تساعدنا على عدم كتابة نفس الكود مرات كثيرة وتوفير الوقت وتقلل الأخطاء.

مثال: إذا أردنا طباعة الأرقام من 1 إلى 100، بدون استخدام التعليمات التكرارية سنكتب 100 سطر (print()، لكن باستخدام التعليمات التكرارية ننجز ذلك بعدد قليل من الأسطر.

أنواع التعليمات التكرارية في بايثون

بايثون توفر نوعين رئيسيين من الحلقات:

- التعليمات **for**: نستخدمها للتكرار على عناصر قائمة أو مجموعة أو نص، أو على أرقام من نطاق محدد.
- التعليمات **while**: تستمر في التكرار طالما أن الشرط صحيح.

كل نوع يُستخدم حسب الحاجة وطبيعة المشكلة.

التعليمات التكرارية for:

هي واحدة من أكثر التعليمات استخدامًا في بايثون. تُستخدم لتكرار عناصر متسلسلة من قائمة أو أي شيء يمكن التكرار عليه (iterable)، وتنفيذ مجموعة من التعليمات لكل عنصر في التسلسل.

الصيغة الأساسية لfor

```
for element in sequence:
```

```
    instructions # التسلسل من التسلل
```

element هو متغير يأخذ قيمة كل عنصر من عناصر التسلسل واحدًا تلو الآخر.

sequence هو التسلسل الذي نريد المرور عليه (قائمة، نص، إلخ).

مثال:

```
word = "Python"

for letter in word:

    print(letter)
```

الشرح:

- **Word** هو النص الذي نريد المرور عليه.
- **Letter** يمثل كل حرف في النص بشكل متتابع.
- كل حرف يطبع باستخدام `print()` أثناء التكرار.

الحل:

```
P
Y
t
h
o
n
```

كل حرف من نص "Python" يظهر في سطر جديد لأنه يتم طباعته داخل حلقة `for`

استخدام الدالة `range()`

الدالة `range()` هي دالة مدمجة في بايثون تُنشئ تسلسلاً من الأرقام حسب المعطيات التي نحددها. غالباً ما

تُستخدم مع حلقة `for` لتحديد عدد التكرارات.

أمثلة على استخدام الدالة `range()`

فيما يلي بعض الأمثلة التي توضح طرق مختلفة لاستخدام الدالة `range()` مع الحلقات في بايثون:

(1) استخدام range() لإنشاء تسلسل من الأرقام

```
for i in range(5):  
    print(i)  
.....:الناتج  
0  
1  
2  
3  
4
```

في هذا المثال، range(5) تنشئ تسلسلاً من الأرقام من 0 إلى 4 (العدد 5 غير مشمول)، وحلقة for تمر على كل رقم وتطبعه.

(2) استخدام range() مع تحديد بداية ونهاية

```
for i in range(2, 6):  
    print(i)  
.....:الناتج  
2  
3  
4  
5
```

هنا، range(2, 6) تنشئ تسلسلاً من 2 إلى 5 (العدد 6 غير مشمول)، وحلقة for تطبع كل رقم من هذا التسلسل.

(3) استخدام range() مع خطوة محددة

```
for i in range(1, 10, 2):  
    print(i)  
.....:الناتج  
1  
3  
5  
7  
9
```

في هذا المثال، `range(1, 10, 2)` تنشئ تسلسلاً من 1 إلى 9 (العدد 10 غير مشمول) مع تخطي رقمين في كل مرة، وحلقة `for` تمر على كل رقم وتطبعه.

4. استخدام `range()` بالعد التنازلي

```
for i in range(4, 0, -1):  
    print(i)  
.....:الناتج  
4  
3  
2
```

1

هنا، `range(4, 0, -1)` تنشئ تسلسلاً من 4 إلى 1 بالعد التنازلي، وحلقة `for` تطبع كل رقم من هذا التسلسل.

تعليمات التحكم `break` و `continue`

1) `break` التعليمة

التعليمة `break` تستخدم للخروج المبكر من الحلقة عندما يتحقق شرط محدد. بمعنى آخر، توقف تنفيذ الحلقة فوراً حتى لو لم تصل الحلقة إلى شرط نهايتها الطبيعي بعد.

مثال: الخروج من حلقة عند الوصول للقيمة 5

```
for i in range(1, 11):  
    print(i)  
    if i == 5:  
        print("الخروج من الحلقة")  
        break  
.....:الناج  
1  
2  
3  
4  
5  
الخروج من الحلقة
```

الشرح:

الحلقة تبدأ من 1 إلى 10، وعندما تصل قيمة i إلى 5 يتم تنفيذ break والخروج من الحلقة فوراً.

(2) التعليمة continue

التعليمة **continue** تستخدم لتخطي ما تبقى من تعليمات الحلقة في هذه الدورة والانتقال مباشرة إلى الدورة التالية للحلقة. تُستخدم لتجاهل بعض التكرارات وفقاً لشرط معين.

مثال: تجاهل الأرقام الزوجية

```
for i in range(1, 11):  
    if i % 2 == 0:  
        continue  
    print(i)
```

.....:الناتج

1
3
5
7
9

الشرح:

الحلقة تمر من 1 إلى 10، وعندما تكون قيمة i زوجية، يتم تنفيذ `continue` فتتجاوز الحلقة هذا الرقم وتنتقل إلى الرقم التالي. و بالتالي يتم طباعة الأرقام الفردية فقط.

أمثلة تطبيقية:

مثال 1:

قم باستخدام الحلقة `for` و الدالة `range()` للقيام بحساب مجموع الأرقام من 1 إلى 5، ثم طباعة النتيجة

الحل:

```
total = 0  
for i in range(1, 6):  
    total += i  
print("مجموع الأرقام من 1 إلى 5 هو", total)
```

مثال 2:

أكتب برنامج يقوم بحساب الراتب الشهري بعد زيادة 50 دينار لكل شهر و لمدة 5 أشهر

```
salary = 1000
for month in range(1, 6):
    salary += 50
    print(f"الراتب بعد الشهر {month} هو: {salary}")
.....:النتائج
الراتب بعد الشهر 1 هو: 1050
الراتب بعد الشهر 2 هو: 1100
الراتب بعد الشهر 3 هو: 1150
الراتب بعد الشهر 4 هو: 1200
الراتب بعد الشهر 5 هو: 1250
```

المحاضرة 06:

التعليمة التكرارية while

الحلقة while هي حلقة تستخدم لتكرار تنفيذ مجموعة من التعليمات مادام أن شرطاً معيناً صحيح. بمعنى آخر، تستمر الحلقة في العمل حتى يصبح الشرط خطأ.

الصيغة الأساسية ل while

```
while condition:
    تنفيذ هذه التعليمات طالما الشرط صحيح # instructions
```

- Condition هو الشرط الذي يجب أن يكون صحيحاً لتستمر الحلقة.
- عندما يصبح الشرط خطأ، تتوقف الحلقة عن التكرار.

مثال 1: طباعة الأرقام من 1 إلى 5

```
i = 1 # متغير يبدأ من 1
while i <= 5: # شرط التكرار
    print(i)
    i += 1 # زيادة المتغير في كل دورة
.....:الناتج
1
2
3
4
5
```

الشرح:

- i يبدأ من 1.
- مادام $i \leq 5$ ، تطبع الحلقة قيمة i ثم تزيدها بمقدار 1.
- عندما يصبح i أكبر من 5، تتوقف الحلقة.

مثال 2: حساب مجموع الأرقام من 1 إلى 5

```
total = 0
i = 1
while i <= 5:
    total += i
    i += 1
print ("مجموع الأرقام من 1 إلى 5 هو:", total)
.....:الناتج
مجموع الأرقام من 1 إلى 5 هو: 15
```

الشرح:

نبدأ بمجموع صفر.

الحلقة تضيف في كل مرة رقم من 1 إلى total.

بعد انتهاء الحلقة، نطبع المجموع النهائي.

مثال 3: أكتب برنامجا يطلب من المستخدم إدخال رقم موجب و لا ينتهي البرنامج مادام المستخدم ال يعطيه عدد موجب

الحل:

```
nombre = int(input(" أدخل رقمًا موجبًا "))
# حلقة while للتأكد من أن الرقم موجب
while nombre <= 0:
    print(" يجب أن يكون الرقم موجبًا ")
    nombre = int(input(" أدخل رقمًا موجبًا "))
print(" لقد أدخلت الرقم:", nombre)
```

الشرح:

في هذا المثال، تطلب حلقة while من المستخدم إدخال رقم موجب.

مادام أن المستخدم يدخل رقمًا صفرًا أو سالبًا، تستمر الحلقة في الطلب.

عندما يدخل المستخدم رقمًا موجبًا، تتوقف الحلقة وتُعرض الرسالة مع الرقم المدخل.

مثال 4:

أكتب برنامجا يطلب من المستخدم الإجابة على السؤال: "كم عدد أشهر السنة؟"

إذا كانت الإجابة غير صحيحة، اطبع "إجابة خاطئة! حاول مرة أخرى."

استمر في طلب الإجابة حتى يجيب المستخدم الإجابة الصحيحة "12"، ثم اطبع "إجابة صحيحة."

الحل:

```
answer = int(input("كم عدد أشهر السنة؟"))  
  
while answer != 12:  
    print("إجابة خاطئة! حاول مرة أخرى")  
    answer = int(input("كم عدد أشهر السنة؟"))  
  
print("إجابة صحيحة!")
```