

## أوامر التكرار البرمجية



## وصف الدرس

الغرض العام من هذا الدرس هو أن يتعرف الطلبة على ضرورة أهمية التكرارات في البرمجة، وعلى أنواع التكرارات البرمجية التي تقدمها Python مثل تكرار **while**، وحالات استخدامه. سيتعرفون أيضاً على أنواع الأخطاء التي نقع بها أثناء كتابتنا لبرامج الحاسوب.

## ما سيتعلمه الطالب

- < أوامر التكرار.
- < بناء وتتبع الخوارزميات لحل المشكلات.
- < الفرق بين الأخطاء المنطقية واللغوية.

## نتائج التعلم

- < أنواع الأخطاء البرمجية وكيفية تتبعها وتصويبها.

## المصطلحات

اللغة الإنجليزية	اللغة العربية
Syntax error	خطأ لغوي
Logical error	خطأ منطقي
Loops	أوامر التكرار
Bug	خلل برمجي



## التحديات المتوقعة



< قد يجد بعض الطلبة صعوبة في ملاحظة الاختلاف بين تكراري **for** و **while**. ساعدهم بتوضيح أن تكرار **for** يستخدم في حال معرفة عدد التكرارات منذ البداية، أي عند تنفيذ الأوامر، أما تكرار **while** فهو يُستخدم عندما يعتمد التكرار على تحقق شرط معين.

< قد لا يتمكن بعض الطلبة في بعض الأحيان من فهم ماهية التكرار اللانهائي بسهولة. قم بطرح بعض الأمثلة حول هذا التكرار من برامج **Scratch** و **Alice** التي درسها الطلبة في السنوات السابقة.

< قد يواجه الطلبة صعوبة في فهم الأنواع المختلفة من الأخطاء التي يمكن أن يتعرض لها المبرمج أثناء كتابة البرنامج. استعن بأمثلة كتاب الطالب لتوضيح الفرق بين الأنواع المختلفة من الأخطاء.



## التمهيد

< قد يمتلك الطلبة بعض المعرفة الأساسية حول **Python** من السنوات السابقة. من الضروري الربط بين تلك الخبرات السابقة والمعرفة الجديدة من خلال التذكير ببعض المصطلحات والمفاهيم الأساسية في **Python** كالمغيرات والشروط والتكرارات.

< يمكنك بعد ذلك المتابعة باستخدام استراتيجية التعلم القائم على حل المشكلات، وطرح مسألة بسيطة ودعوة الطلبة لكتابة التعليمات البرمجية لحلها.

< يمكنك مثلا طرح مسألة النجاح في المساق الدراسي، والتي يمكن التنبؤ بها بكتابة برنامج يقوم بقراءة درجة طالب ومعرفة فيما إذا كانت أكبر من 50 للنجاح.

< قم بالتوسع في هذه المشكلة بدعوة الطلبة إلى التفكير في الأوامر التي يحتاج الطالب إلى إضافتها إلى البرنامج السابق في حال كان الفصل يضم 20 طالبًا.

< بعد التأكد من الربط بين المفاهيم المختلفة، يمكن تقديم التكرار الجديد، وهو تكرار **while**، وذلك من خلال استراتيجية التعليم المباشر.



## التلميحات الخاصة بالتنفيذ

- < يمكنك الاعتماد على المثال السابق والتعديل عليه لاستخدام تكرار **while**. على سبيل المثال، على الطلبة التفكير في كيفية إيقاف البرنامج من خلال إدخال رقم سلبي في حالة عدم المعرفة المسبقة بعدد الطلبة في الفصل.
- < ساعد الطالب في إضافة التعليمات البرمجية الجديدة في برامجهم وتشغيلها لمعرفة النتيجة.
- < يتعين عليك أن تشرح للطلبة سهولة ارتكاب الأخطاء. قم بعرض بعض أنواع الأخطاء التي يمكن للمبرمج أن يقع بها أثناء كتابته للبرنامج، واستعن بالأمثلة من كتاب الطالب لتوضيح الاختلافات بين تلك الأخطاء.
- < يمكنك أيضًا أن تطلب منهم إضافة أو إزالة بعض الأوامر البرمجية من البرنامج السابق، وذلك للتعرف على بعض الأخطاء التي ارتكبوها.



## استراتيجيات غلق الدرس

- في نهاية الدرس تأكد من تحقيق الطلبة لجميع أهداف الدرس وتقييم معرفتهم من خلال أسئلة على سبيل المثال لا الحصر:
- < هل تستطيع أن تتذكر:

• متى نستخدم تكرار **while**؟

• ما هو الفرق بين تكرار **while** وتكرار **for**؟

• ما هي أنواع الأخطاء المختلفة التي تواجه المبرمج أثناء كتابة البرامج؟

- < ذكّر الطلبة بالمصطلحات الهامة وكرّرها معهم.
- < يمكنك الاستعانة بتدريبات الكتاب ضمن الاستراتيجيات التي ستستخدمها لغلق الدرس.

## التدريبات المقترحة لخلق الدرس



يمكنك استخدام التمرين السابع ضمن استراتيجية خلق الدرس لتقييم وتعزيز قدرة الطلبة على تطبيق المهارات التي تم تقديمها في هذا الدرس.

الصف الحادي عشر | الفصل الأول | كتاب الطالب | صفحة 22

## الفروق الفردية

### تمارين إضافية للطلبة ذوي التحصيل المرتفع

< يمكنك التوسع في المهام المطلوبة في التمرين 7 في الصفحة 22.  
< اطلب من الطلبة عرض عدد المحاولات المتبقية للمستخدم على الشاشة.

## الإجابات النموذجية للتدريبات:



في أي الحالات يفضل استخدام كل من:

جملة التكرار While:

يستخدم تكرار **while** عندما لا يكون عدد مرات التكرار معروف سلفاً، حيث يستمر التكرار في العمل طالما كان شرط التكرار صحيحاً، وتستخدم المسافة البادئة في بداية الجمل البرمجية التي يحتويها مقطع التكرار تحت الأمر **while**، وذلك لتميزها أثناء قراءة المقطع.

جملة التكرار For:

يستخدم تكرار **for** عند الحاجة إلى تكرار تركيب برمجي لعدد محدد من المرات معروف مسبقاً.



اختر الإجابة الصحيحة:

سيتم تكرار الأوامر داخل جملة التكرار While في البرامج التالية:

<input checked="" type="radio"/>	6 مرات	1. i=1 while i<6: i=i+1 if i == 3: print("Hello") print(i)
<input type="radio"/>	حتى تصبح قيمة i = 3	
<input type="radio"/>	5 مرات	
<input type="radio"/>	سيتم تنفيذه 5 مرات.	2. i=5 while i>1: print(i) i=i-1
<input type="radio"/>	عدد مرات التكرار غير معروف.	
<input checked="" type="radio"/>	سيتم تنفيذه 4 مرات.	

1

2

3

4

5

6

1

2

3

4

5



3

ضع إشارة ✓ أمام الإجابة الصحيحة:

1. يتم التحقق من الشرط مرة واحدة على الأقل قبل تنفيذ جملة التكرار While.

خطأ  صحيح

2. تُنفذ الأوامر في تكرر While مرة واحدة على الأقل.

خطأ  صحيح

3. يتم تنفيذ تكرر While طالما كان الشرط الموجود خطأً.

خطأ  صحيح



4

ما ناتج تنفيذ كل من الجمل البرمجية التالية:

25 16 9 4	<pre>for i in range(5,1,-1):     print (i*i)</pre> <p>.1</p>
nothing is printed	<pre>for k in range(5,5):     print (k)</pre> <p>.2</p>
8 ==== 7 ==== 6 ==== 5 ====	<pre>i=6 while i!=2:     print (i+2)     print ("====")     i=i-1</pre> <p>.3</p>
9	<pre>sum=0 i=1 while i&lt;=6:     sum=sum+i     i=i+2 print(sum)</pre> <p>.4</p>



حوّل المقاطع البرمجية التالية باستخدام جملة التكرار While.

```
i=0
while(i<5):
    print (i)
    i=i+1
```

```
for i in range(5):
    print (i) .1
```

```
i=10
while(i>2):
    print (i)
    i=i-2
```

```
for i in range(10,2,-2):
    print (i) .2
```

```
i=3
while(i<9):
    print (i+2)
    i=i+1
print (100)
```

```
for i in range(3,9):
    print (i+2) .3
print (100)
```



اكتب برنامجًا باستخدام جملة التكرار While يطلب من المستخدم إدخال عدد محصور بين 1 و 18 لطباعة جملة ( I Love Qatar ). بحيث يعود البرنامج ويطلب إدخال العدد مرة أخرى إذا تم إدخال عدد من خارج النطاق (1-18).

```
num=int(input("Enter a number: "))
while num<1 or num >18:
    print("Wrong number, enter a number between 1-18.")
    num=int (input("Enter a new number: "))
print ("I Love Qatar ")
```



7

اكتب برنامجاً يطلب إدخال كلمة المرور الصحيحة وبعد أقصى 5 محاولات.

< يطلب البرنامج إدخال كلمة مرور رقمية.

< إذا تمت كتابة الرقم الصحيح فإن البرنامج يطبع "كلمة مرور صحيحة"، وإلا فإنه يطبع "حاول مرة أخرى!".

< بعد المحاولة الخامسة يتوقف ويطلب "تجاوزت عدد المحاولات".

```
myPassword=int(input("Enter your secret password: "))
cnt=1
password=int(input("Enter a password: "))
while password!=myPassword and cnt<5:
    print("Wrong password, try again")
    password=int (input("Enter a new password: "))
    cnt=cnt+1
if cnt<=5 and password==myPassword:
    print("Correct password")
else:
    print ("You run out of tries.")
```



8

اكتب برنامجاً يطبع الأعداد المحصورة بين 1 إلى 10 ، باستثناء العدد 5 باستخدام جملة التكرار.

```
for i in range(1,10):
    if i!=5:
        print (i)
```