# الجمل الشرطية

## أنواع الجمل الشرطية في بايثون (Python)

كما تعلمنا في اللغة العربية واللغة الإنجليزية هناك جمل من نوع جمل شرطية، وهي عبارة عن جملة مركبة تحتوي على جملتين متلازمتين مسبوقتين بأداة شرط، حيث أن تحقق أحدهما يعتمد على تحقق الثانية.

وفي عالم الكمبيوتر،أيضا هنالك جمل شرطية، والتي تمثل مجموعة من الأوامر تُمكّن البرنامج من اتخاذ قرارات بناءً على شروط معينة .تُستخدم لتنفيذ جزء معين من الكود إذا كان الشرط صحيحًا، ويمكن تنفيذ جزء آخر إذا كان الشرط خاطئًا.

وفيما يلي شرحاً توضيحياً حول كل نوع من أنواع هذه الجمل الشرطية:

### أولًا :الجملة الشرطية (if statement) if

الصيغة العامة لكتابة الجملة الشرطية if هي كما في الشكل التالي:

## أولًا: الجملة الشرطية ( if (if statement

على النحو الآتي: ( if ) تُكتَب الصيغة العامة للجملة الشرطية

حيث :

**If** : هي كلمة محجوزة في لغة البرمجة بايثون (Python)

```
if condition:
    statements1
```

**Condition** : يمثل الشرط، وهو عبارة عن تعبير منطقي سوف نتعرف عليه في نهاية الدرس

**Statements**: محموعة أوامر برمجية يتم تنفيذها إذا كان الشرط صحيحاً (أي تم تحقق الشرط)

(**:**) النقطتان الرئسيتان وهما علامة يجب وضعها بعد الشرط ( Condition ) كي يتمكن مفسر لغة البرمجة (Python) من تنفيذ البرنامج.

## وفيما يلي أمثلة على الجملة الشرطية (if)

**مثال (1)**

اكتب برنامجاً بلغة البرمجة بايثون (Python) يطبع عبارة ( **y is greater than x**) في حال كانت قيمة المتغير (**y**) أكبر من قيمة المتغير (**x**).

**الحل:**

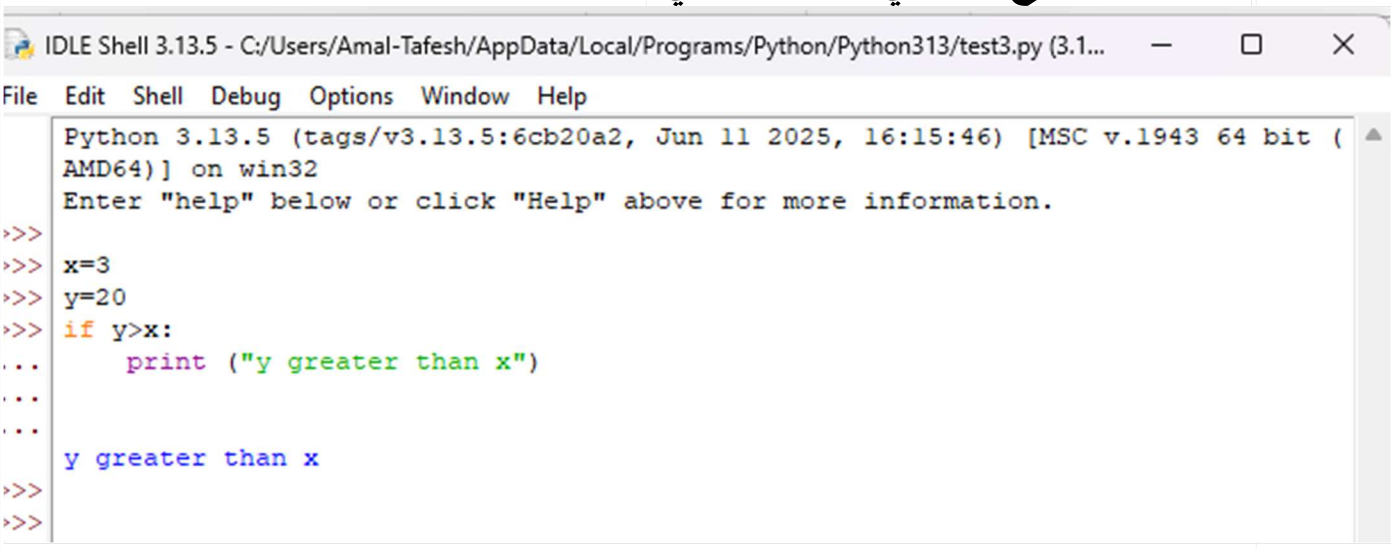
```
x = 3
y = 20
if y > x:
print("y is greater than x")
```

**إضاءة**

**يمكن كتابة الجملة الشرطية السابقة بصورة مختصرة كما يلي:**

```
y  if x:> print ("y is greater than x")
```

**عند تشغيل البرنامج سوف يقوم البرنامج بطباعة العبارة (y is greater than x) كما في الشكل التالي:**



**نشاط عملي:**

أُعدِّل المقطع البرمجي (**الكود**) في المثال السابق لتعيين قيمة (30) للمُتغيِّر (**x**) وأتتبَّع النتيجة المُتوقَّعة من دون تشغيل البرنامج، ثمَّ أتحقَّق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (**Python**) ما الذي يجب تعديله في البرنامج لطباعة جملة "**y is greater than x**" بدلًا من جملة "**x is greater than y**"؟

**الحل:**

نكتب البرنامج عبى النحو التالي، بحيث نعدل جملة الطباعة لتصبح كما يظهر في الشكل التالي:

```
x =30
y = 20
if x > y: print("x is greater than y")
```

```
x is greater than y
```

## ثانيًا: الجملة الشرطية (if else)

تُكتَب الصيغة العامة للجملة الشرطية ( if else ) على النحو الآتي:

```
if condition:
```

```
  statements1
else:
  statements2
```

حيث:

if, else : كلمتان محجوزتان في لغة البرمجة بايثون (Python)

statements1 : أوامر برمجية تُنفَّذ إذا كان الشرط صحيحًا (تحقُّق الشرط).

statements2 : أوامر برمجية تُنفَّذ إذا لم يكن الشرط صحيحًا (عدم تحقُّق الشرط).

: (:) علامة يجب أنْ توضع بعد الشرط ( condition ) و بعد جملة else.

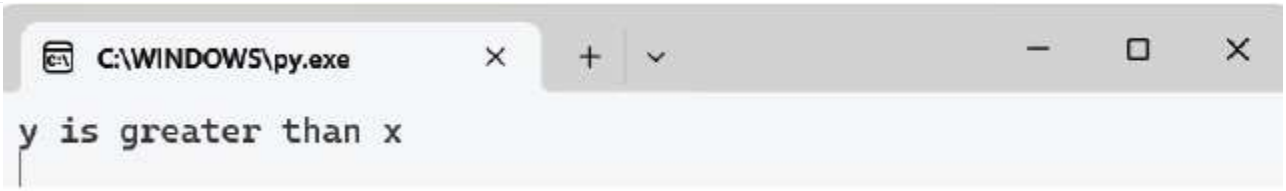
مثال:

إذا كانت قيمة المُتغيّر ( x ) أقل من قيمة المُتغيّر (y) يطبع البرنامج الآتي عبارة " y is less than x "

إذا كانت قيمة المُتغيّر ( x ) أقل من قيمة المُتغيّر (y) ويطبع البرنامج عبارة " x is less than y "

```
x = 80
y = 100
if y < x:
  print("y is less than x")
else:
  print("x is less than y")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:




## إضاءة

يُمكِن كتابة الجملة الشرطية السابقة بصورة مختصرة كما يأتي:

```
x = 80
y= 100
print("y is less than x") if y < x else print("x is less than y")
```

**نشاط عملي**

**أُجرِّب بنفسي:** أُعدِّل المقطع البرمجي (الكود) في المثال السابق لتعيين قيمة (100) للمُتغيِّر (x)، وأتتبَّع النتيجة المُتوقَّعة من دون تشـــغيل البرنامج، ثمَّ أتحقَّق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).

```python
x = 100
y = 100
if y < x:
 print("y is less than x")
else:
  print("x is less than y")
```

```
x is less than y
```

## ثالثًا: الجملة الشرطية (if elif)

تُكتَب الصيغة العامة للجملة الشرطية ( if elif ) على النحو الآتي:

```
if condition1:
   statements1
elif condition2:
   statements2
```

حيث:

if, elif : كلمتان محجوزتان في لغة البرمجة بايثون (Python)

statements1 : أوامر برمجية تُنفَّذ إذا تحقَّق الشرط (condition1)

statements2 : أوامر برمجية تُنفَّذ إذا تحقَّق الشرط (condition2)

مثال:

إذا كانت قيمة ( x ) أكبر من قيمة المُتغيِّر ( y ) المُتغيِّر يطبع البرنامج الآتي عبارة " y is greater than x "

إذا كانت قيمة المُتغيِّرَ ( x ) تساوي قيمة المُتغيِّر (y) ويطبع البرنامج عبارة " x equals y "

```
x = 20
y = 20
if y > x:
print('y is greater than x')
elif x == y:
print('x equals y')
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:

نشاط عملي

**أُجرِّب بنفسي:** أُعدِّل المقطع البرمجي (الكود) في المثال السابق لتعيين قيمة (30) للمُتغيِّر (x)، وأتتبَّع النتيجة المُتوقَّعة من دون تشغيل البرنامج، ثمَّ أتحقَّق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).

```
[10] x = 30
     y = 20
     if y > x:
      print('y is greater than x')
     elif x == y:
      print('x equals y')
```

0s

نشاط فردي

أكتب جملًا برمجية بلغة البرمجة بايثون (Python) لحساب قيمة المُتغيِّر (z) وَفقًا لكل معادلة ممّا يأتي وطباعته:

$z = x^2 * y \quad , x >= 5$

$z = x * (x + y) \ , x == 5$

```
x=5
y=2
if x >= 5:
    z = x**2 * y
    print(z)
elif x == 5:
    z = x * (x + y)
    print(z)
```

```
50
```

```
x=10
y=2
if x >= 5:
    z = x**2 * y
    print(z)
elif x == 5:
    z = x * (x + y)
    print(z)
```

```
200
```

## رابعًا: الجملة الشرطية (if elif else)

تُكتَب الصيغة العامة للجملة الشرطية ( if elif else ) على النحو الآتي:

```
if condition1:
statements1
elif condition2:
```

```
statements2
else:
statements3
```

حيث:

if, else, elif : كلمات محجوزة في لغة البرمجة بايثون (Python)

statements1 : أوامر برمجية تُنفَّذ إذا تحقَّق الشرط (condition1)

statements2 : أوامر برمجية تُنفَّذ إذا تحقَّق الشرط (condition2)

statements3 : أوامر برمجية تُنفَّذ إذا لم يتحقَّق أيٌّ من الشرطين (condition2, condition1)

مثال:

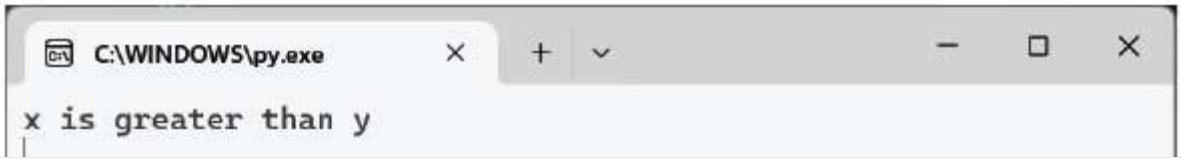
إذا كانت قيمة ( x) أكبر من قيمة المُتغيّر ( y) المُتغيّر يطبع البرنامج الآتي عبارة " y is greater than x "

إذا كانت قيمة المُتغيّر ( x) تساوي قيمة المُتغيّر ( y)، ويطبع البرنامج عبارة " x equals y "

إذا كانت قيمة المُتغيّر ( x) أصغر من قيمة المُتغيّر (y) ويطبع البرنامج عبارة " x is greater than y "

```
x = 50
y = 25
if y > x:
print("y is greater than x")
elif y == x:
print("x equals y")
else:
print("x is greater than y")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:




نشاط عملي

**أُجرِّب بنفسي:**

استنادًا إلى المثال السابق، أُجيب عن الأسئلة الآتية:

1. ما النتيجة المُترتِّبة على تشـــغيل البرنامج إذا كانت قيمة (x) تساوي (40)؟ أُعدِّل المقطع البرمجي (الكود) في المثال لتعيين قيمة (40) للمُتغيِّر (x)، وأتتبَّع النتيجة المُتوقَّعة من دون تشغيل البرنامج، ثمَّ أتحقَّق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).
2. ما النتيجة المُترتِّبة على تشـــغيل البرنامج إذا كانت قيمة (x) تساوي (10)؟ أُعدِّل المقطع البرمجي (الكود) في المثال لتعيين قيمة (10) للمُتغيِّر (x)، وأتتبَّع النتيجة المُتوقَّعة من دون تشغيل البرنامج، ثمَّ أتحقَّق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).
3. ما النتيجة المُترتِّبة على تشـــغيل البرنامج إذا كانت قيمة (x) تساوي (25)؟ أُعدِّل المقطع البرمجي (الكود) في المثال لتعيين قيمة (25) للمُتغيِّر (x)، وأتتبَّع النتيجة المُتوقَّعة من دون تشغيل البرنامج، ثمَّ أتحقَّق من الناتج عن طريق تنفيذ البرنامج في بيئة بايثون (Python).

```
x = 10
y = 25
if y > x:
  print("y is greater than x")
elif y == x:
  print("x equals y")
else:
  print("x is greater than y")
```

```
y is greater than x
```

```
if y > x:
  print("y is greater than x")
elif y == x:
  print("x equals y")
else:
  print("x is greater than y")
```

```
greater than y
```

```
 is greater than x")
= x:
 equals y")

 is greater than y")

y
```

**أبحث**

**أبحث:** أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن لغات برمجة أُخرى، ثمَّ أُقارِن طريقة كتابة الجمل الشرطية فيها بطريقة كتابة الجمل الشرطية في لغة البرمجة بايثون (Python).

الحل : مقارنة بين طريقة كتابة الجملة الشرطية في لغة Python ولغة C++

في Python كتابة الاوامر تعتمد على المسافات
في C++ تستخدم الأقواس المتعرجة {}
الكلمة المفتاحية else اختيارية في Python و إلزامية في C++ إذا أردت تنفيذ تعليمات بديلة

الجملة الشرطية في Python: تتميز Python ببساطتها وقابليتها للقراءة، وتستخدم الكلمة المفتاحية if لبدء الجملة الشرطية.
مثال:

```
Python
x = 10
```

```
if x > 5:
    print("القيمة أكبر من 5")
else:
    print("القيمة أصغر من أو تساوي 5")
```

الجملة الشرطية في C++ تستخدم الكلمة المفتاحية if أيضًا لبدء الجملة الشرطية، ولكن الصيغة تختلف قليلاً عن Python:

مثال:

```
C++
int x = 10;
if (x > 5) {
    cout << "القيمة أكبر من 5" << endl;
} else {
    cout << "القيمة أصغر من أو تساوي 5" << endl;
}
```

## المُعامِلات المنطقية (Logical Operators)

لربط التعابير ( or ) والمُعامِل المنطقي (and)يُستعمَل المُعامِل المنطقي المنطقية البسيطة، وتكوين جمل منطقية مُركَّبة، في حين يُستعمَل المُعامِل المنطقي ( not ) لنفي التعابير المنطقية.

### 1. المُعامِل المنطقي(and):

قد يتوقَّف تنفيذ أمر برمجي مُعيَّن في البرنامج على تحقُّق مجموعة من الشروط مُجتمِعةً.

مثال:

إذا كان مُعدَّل الطالب/ الطالبة أكبر من ( 100 ) أو يساوي ( 90 ) وأقل من أو يساوي ( 100 ) يراد طباعة كلمة " Excellent ".

الحَلُّ:

للدلالة على تحقُّق الشرطين معًا، يجب استعمال المُعامِل المنطقي ( and ) وتُكتَب الأوامر البرمجية على النحو الآتي:

```
if Avg >= 90 and Avg <=100:
print("Excellent")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:

نشاط عملي

**أُجرِّب بنفسي:** أكتب المقطع البرمجي (الكود) السابق، ثمَّ أتحقَّق من النتيجة عن طريق تنفيذ البرنامج.

أُعدِّل قيمة المُتغيِّر (Avg) لتصبح (80)، ثمَّ أُنفِّذ البرنامج. ما الناتج الظاهر على شاشة جهاز الحاسوب؟

أحذِف المسافة البادئة قبل جملة الطباعة، ثمَّ أُنفِّذ البرنامج. ما ناتج تنفيذ البرنامج؟

```
Avg = 80
if Avg >= 90 and Avg <=100:
 print("Excellent")
```

```
Avg = 80
if Avg >= 90 and Avg <=100:
print("Excellent")
```

```
  File "<ipython-input-4-4fa7933b631b>", line 3
    print("Excellent")
    ^
IndentationError: expected an indented block after 'if' statement on line 2
```

## 2. المُعامِل المنطقي (or)

قد يتوقَّف تنفيذ أمر مُعيَّن في البرنامج على تحقُّق شرط من مجموعة شروط.

مثال:

إذا كانت قيمة هذا المُتغيِّر تساوي ( 1( أو ) (x )يراد طباعة قيمة المُتغيِّر (2.

الحَلُّ:

للدلالة على تنفيذ جملة الطباعة في ( or ) يجب استعمال المُعامِل المنطقي حال تحقُّق أحد الشرطين، وتُكتَب الأوامر البرمجية على النحو الآتي:

```
x = 1
if x == 1 or x == 2:
print('you selected a valid number')
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:

## 3. المُعامِل المنطقي (not):

قد يتوقَّف تنفيذ أمر في البرنامج على عدم تحقُّق شرط مُعيَّن.

**مثال:**

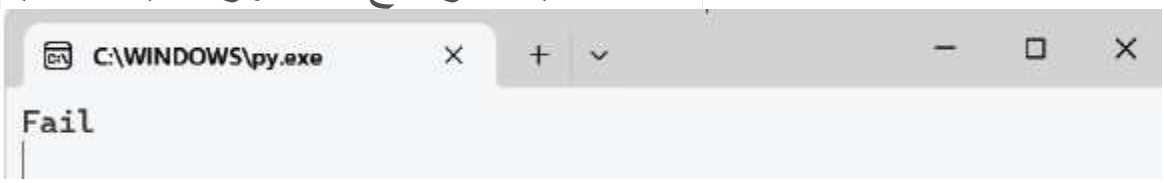
إذا لم يُحقِّق الطالب/ الطالبة شرط النجاح (Avg >= 50) يراد طباعة كلمة " Fail "

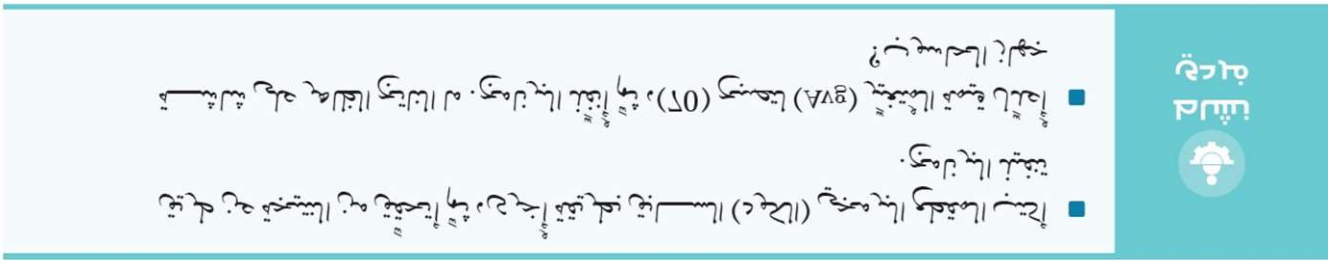
**الحَلُّ:**

للدلالة على عدم تحقُّق الشرط، يجب استعمال المُعامِل المنطقي ( not ) وتُكتَب الأوامر البرمجية على النحو الآتي:

```
Avg = 49
if not Avg >= 50:
print("Fail")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



```
vg = 70
f not Avg >= 50:
print("Fail")
```

```
g = 49
  Avg >= 50:
  print("pass")
lse:
  print("Fail")

il
```

نشاط جماعي

**المُعامِلات المنطقية والجمل الشرطية**

أكتب – بالتعاون مع أفراد مجموعتي– مقطعًا برمجيًّا بلغة البرمجة بايثون (Python) لإدخال قيمة إحدى فواتير الشراء وعدد المشـــتريات، وطباعة عبارة "No discount" إذا كانت قيمة الفاتورة أقل من (100)، أو كان عدد المشـــتريات أقل من (3)، وطباعة قيمة الخصم الذي يساوي 5% من قيمة الفاتورة إذا كانت قيمتها أكبر من (100)، وكان عدد المشتريات أكبر من أو يساوي (3).

```
ill_amount = float(input("أدخل قيمة الفاتورة: "))
um_purchases = int(input("أدخل عدد المشتريات: "))

f bill_amount < 100 or num_purchases < 3:
 print("No discount")
lse:
 discount = bill_amount * 0.05
 print("قيمة الخصم:", discount)
```

```
0.5 :أدخل قيمة الفاتورة
0 :أدخل عدد المشتريات
o discount
```

```
bill_amount = float(input("أدخل قيمة الفاتورة: "))
num_purchases = int(input("أدخل عدد المشتريات: "))

if bill_amount < 100 or num_purchases < 3:
  print("No discount")
else:
  discount = bill_amount * 0.05
  print("قيمة الخصم:", discount)
```

```
أدخل قيمة الفاتورة: 130
أدخل عدد المشتريات: 10
قيمة الخصم: 6.5
```

## الجملة الشرطية المُركَّبة (Nested Conditional Statements)

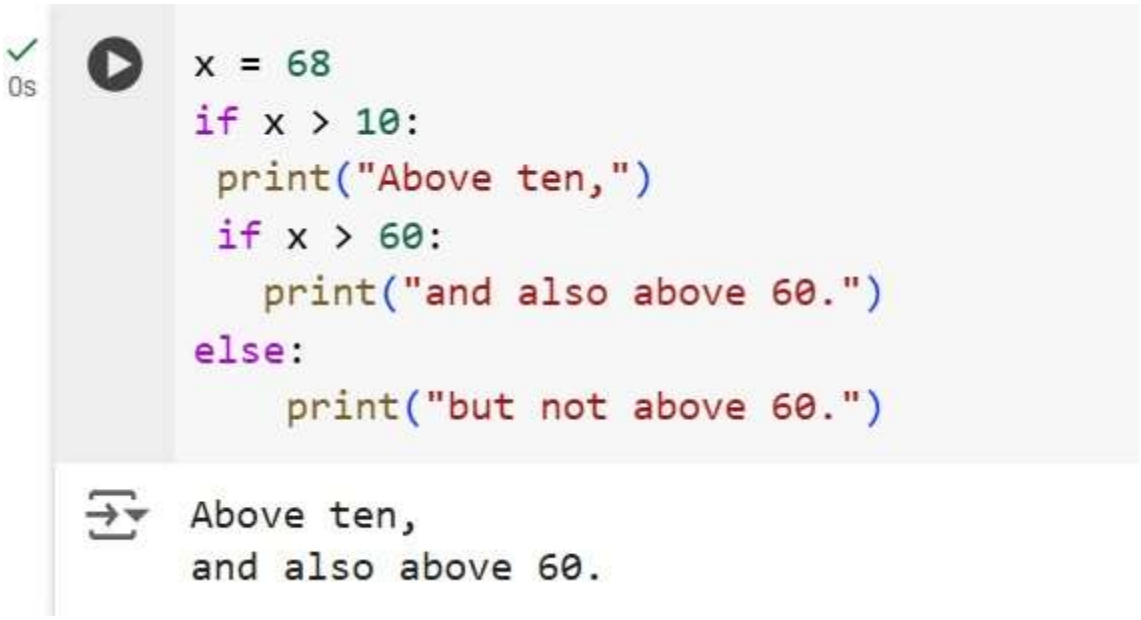
إمكانية كتابة جمل شرطية مُركَّبة ( Python ) تُوفِّر لغة البرمجة بايثون أُخرى ( if ) داخل جملة شرطية ( if )(مُتداخِلة)؛ أيْ وضع جملة شرطية.

مثال:

يطبع البرنامج الآتي عبارة " Above ten " إذا كانت قيمة ( x) أكبر من ( 10 )، ثمَّ يتحقَّق إذا كانت قيمة (x) أكبر من ( 60). فيطبع البرنامج، إضافة إلى العبارة السابقة، عبارة " and also above 60 " و يطبع(x) إذا كانت قيمة ( x) أكبر من ( 60)، وإلا فيطبع عبارة " but not above 60 ".

```
x = 68
if x > 10:
print("Above ten,")
if x > 60:
print("and also above 60.")
else:
print("but not above 60.")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:



نشاط فردي

- أُعدِّل قيمة المُتغيِّر (x) لتصبح (25)، ثمَّ أُنفِّذ البرنامج. ما الناتج الظاهر على شاشة جهاز الحاسوب؟
- أُعدِّل قيمة المُتغيِّر (x) لتصبح (5)، ثمَّ أُنفِّذ البرنامج. ما الناتج الظاهر على شاشـــة جهاز الحاسوب؟
- أحذِف المسافة البادئة قبل جملة الطباعة الأخيرة. ما ناتج تنفيذ البرنامج؟ أُناقِش إجاباتي مع زملائي/ زميلاتي في الصف.

```
[14] x = 5
     if x > 10:
      print("Above ten,")
     if x > 60:
      print("and also above 60.")
     else:
      print("but not above 60.")
```

```
but not above 60.
```

```
[13] x = 25
     if x > 10:
      print("Above ten,")
     if x > 60:
      print("and also above 60.")
     else:
      print("but not above 60.")
```

```
Above ten,
but not above 60.
```

```
[16]  x = 68
      if x > 10:
       print("Above ten,")
      if x > 60:
       print("and also above 60.")
      else:
      print("but not above 60.")
```

```
  File "<ipython-input-16-89be7f78ba93>", line 7
    print("but not above 60.")
    ^
IndentationError: expected an indented block after 'else' statement on line 6
```

مثال:

يُستخدَم البرنامج الآتي في التحقُّق إذا كانت قيمة المُتغيِّر ( y) تقبل القسمة على ( 3)، وتقبل القسمة على (2)، أو تقبل القسمة على أحدهما ، أو لا تقبل القسمة على أي منهما. ثمَّ يطبع البرنامج العبارة الدالَّة على ذلك.

```
y = 9
if y %2==0:
if y %3==0:
print("divisible by 3 and 2")
else:
print("divisible by 2, but not divisible by 3")
else:
if y %3==0:
```

```
print("divisible by 3, but not divisible by 2")
else:
print("not divisible by 2 and not divisible by 3")
```

عند تشغيل البرنامج، ستظهر النتيجة الآتية:

نشاط عملي

**أُجرِّب بنفسي:**

استنادًا إلى المثال السابق، أُجيب عن الأسئلة الآتية:

1. ما نتيجة البرنامج إذا كانت قيمة (y) تساوي (8)؟
2. أكتب المقطع البرمجي (الكود) الوارد في المثال السابق في لغة البرمجة بايثون (Python)، وأتتبَّع النتيجة المُتوقَّعة من دون تشغيل البرنامج، ثمَّ أتحقَّق من الناتج عن طريق تنفيذ البرنامج.
3. ما نتيجة البرنامج إذا كانت قيمة (y) تساوي (5)؟

```
=0:
==0:
int("divisible by 3 and 2")

"divisible by 2, but not divisible by 3")

==0:
"divisible by 3, but not divisible by 2")

rint("not divisible by 2 and not divisible by 3")
```

```
e by 2, but not divisible by 3
```

```
:=0:
3==0:
·int("divisible by 3 and 2")

["divisible by 2, but not divisible by 3")

3==0:
["divisible by 3, but not divisible by 2")

)rint("not divisible by 2 and not divisible by 3")
```

```
sible by 2 and not divisible by 3
```

## إضاءة

إذا أردْتُ كتابة جملة شـــرطية (if) فارغة لا تحتوي على أيٍّ من الجمل، فإنَّني أضع الجملة (pass) داخل هذه الجملة الشرطية (if)؛ لتجنُّب ظهور رسالة تفيد بوجود خطأ في البرنامج.

نشاط عملي

أُجرِّب بنفسي:

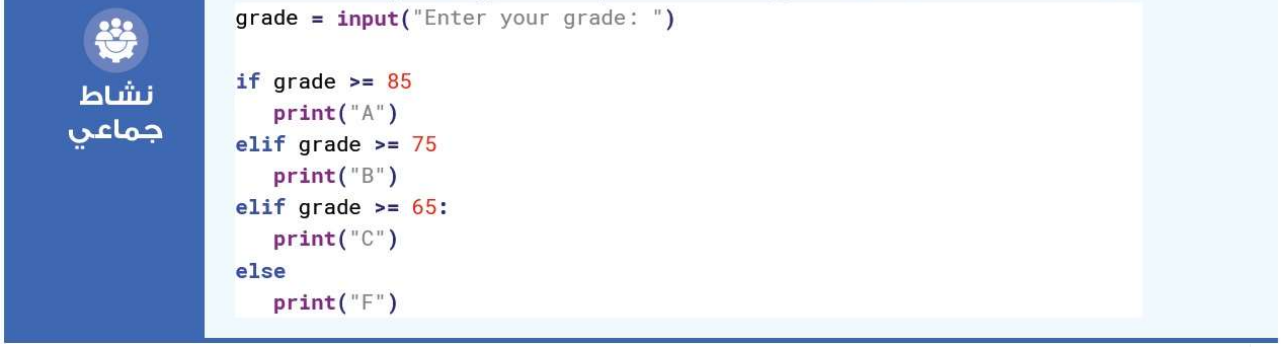
أتتبَّع تنفيذ البرنامج الآتي في بيئة بايثون (Python)، بوجود جملة (pass) تارة، وعدم وجودها تارة أُخرى.

```
x = 4
y = 2
if x > y:
  pass
```

```
x = 4
y = 2
if x > y:

  File "<ipython-input-18-44754a0089d1>", line 3
    if x > y:
           ^
SyntaxError: incomplete input
```

```
x = 4
y = 2
if x > y:
  pass
```




```
grade = int( input("Enter your grade: "))
grade المتغير يجب تعريف

if grade >= 85
              ^
SyntaxError: expected ':'


':' نحتاج بعد جملة الشرط  if  في  جملة
print("A")
   ^
IndentationError: expected an indented block after 'if' statement on line 2


print نحتاج لفراغ قبل الجملة

elif grade >= 75 ^ SyntaxError: expected ':'

':' نحتاج بعد جملة الشرط  if  في  جملة
```

print("B") ^ IndentationError: expected an indented block after 'elif' statement on line 4
print("C")
^
IndentationError: expected an indented block after 'elif' statement on line 6

نحتاج لفراغ قبل الجملة print
else ^ SyntaxError: expected ':'
في جملة else نحتاج بعد جملة الشرط ':'

```
[43]  grade = int( input("Enter your grade: "))
      if grade >= 85:
       print("A")
      elif grade >= 75 :
         print("B")
      elif grade >= 65:
         print("C")
      else:
        print("F")
```

```
Enter your grade: 50
F
```

## المواطنةُ الرقميةُ:

- **التعاون والمشـــاركة**: أتعاون مع الزملاء/ الزميلات، وأُشارِكهم في تنفيذ الأنشطة واكتشاف الأخطاء البرمجية وتحليلها؛ ما يُعزِّز لديَّ مهارة التفكير الناقد والعمل الجماعي، ويُســـهِم في بناء مجتمع تعليمي داعم ومُشارِك.
- **الأخلاقيات الرقمية**: أحترِم حقوق المُلْكية الفكرية عند استخدام المقاطع البرمجية الموجودة في شبكة الإنترنت، أو عند التعديل عليها، وأحرِص على أخذ الموافقة المُسبَّقة على ذلك.