

تم تحميل وعرض المادة من

موقع كتيب



www.kottby.net

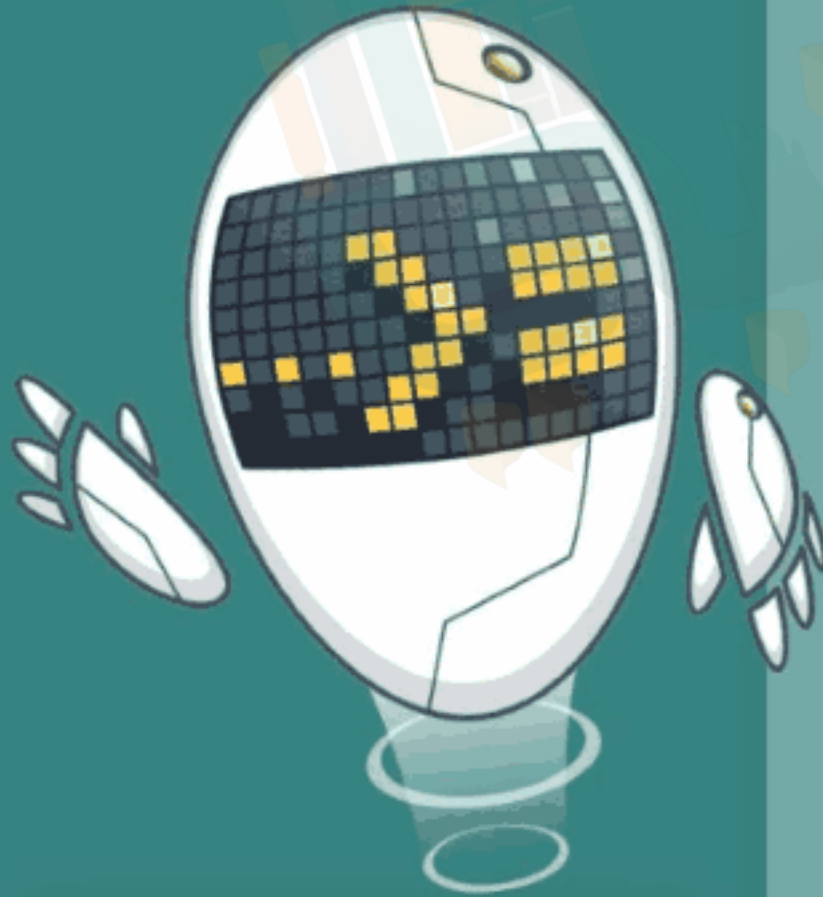
موقع كتيب يعرض لكم الكتب الدراسية الطبعة الجديدة
وحلولها، وشرح للمناهج الدراسية، توزيع المناهج، تحاضير،
أوراق عمل، نماذج إختبارات عرض مباشر وتحميل PDF

حمل التطبيق من هنا



الوحدة الثالثة: البرمجة مع بايثون

ستتعلم في هذه الوحدة كيفية استخدام بيئة التواصل باي تشارم (PyCharm Community) من أجل إنشاء مقاطع برمجية تعالج مشاكل أكثر صعوبة في لغة بايثون (Python). وستتعلم أيضًا كيفية اتخاذ القرار عند البرمجة باستخدام معاملات شرطية ومنطقية جنبًا إلى جنب مع العبارات في بايثون. بالإضافة إلى ذلك ستتعرف على الحلقات (loops) وكيفية استخدامها في بايثون، وستستخدم الحلقات المتداخلة (nested loops) من أجل طباعة الأنماط، كما ستتعلم كيفية تجميع مجموعة من التعليمات البرمجية وإنشاء الدوال. أخيرًا، ستتعلم كيفية التعامل مع ملف إكسل باستخدام بايثون.



الأدوات

إصدار بيئة التواصل باي تشارم
(PyCharm Community Edition)

أهداف التعلم

ستتعلم في هذه الوحدة:

- كيفية إنشاء مقطع برمجي في بيئة التواصل باي تشارم (PyCharm Community).
- كيفية استخدام الأنواع المختلفة من الجمل الشرطية.
- التمييز بين الجمل الشرطية المختلفة.
- استخدام الحلقات في بايثون.
- إيقاف الحلقة باستخدام عبارة الإيقاف.
- آلية استخدام الحلقات المتداخلة.
- إنشاء أنماط باستخدام الحلقات.
- ماهية الدالة في لغة البرمجة.
- بناء دوال جديدة.
- التعامل مع معاملات الدالة.
- تمييز المتغيرات المحلية والعامة.
- استخدام التعليمات البرمجية لمعالجة جداول البيانات.

هل تذكر؟

مراحل إنشاء البرنامج

أولاً: تحديد المشكلة.



ثالثاً: رسم المخطط الانسيابي بناءً على الخطوات الخوارزمية.



ثانياً: كتابة الخطوات الخوارزمية اللازمة لحلها بترتيب متسلسل.



رابعاً: كتابة المقطع البرمجي بلغة البرمجة.



الأشكال المستخدمة في المخطط الانسيابي

الوصف	نوع الصندوق
الإشارة إلى بداية ونهاية المقطع البرمجي.	البداية / النهاية
استقبال وعرض البيانات التي سيتم معالجتها (إدخال وإخراج).	الإدخال / الإخراج
تنفيذ عملية رياضية.	العمليات
اتخاذ قرارات (نعم أو لا) أو اختبارات تحقق (صواب أو خطأ).	اتخاذ قرار
عرض التسلسل الذي يجب تنفيذ الخطوات به.	↓



نوع البيانات	التعريف في بايثون	مثال
الأعداد الصحيحة	int	900000, 0, 999, 12
الأعداد الحقيقية	float	3.0, -90.5, 0.0003, 4.5
النصوص والرموز	str	"\$\$\$", "hello", "Saad"
قيمة منطقية	bool	True, False

أنت تطلب من المستخدم إدخال قيمة للمتغير x.

"الرجاء إدخال قيمة للمتغير x:"
"قيمة x:"

يُدخل المستخدم القيمة 10 ويضغط على .

الرجاء إدخال قيمة للمتغير x:
10
قيمة x: 10

تم إسناد القيمة 10 للمتغير x.

العمليات الرياضية في لغة البايثون

رياضيًا	في بايثون
$2 + 4$	$2 + 4$
$2 - 4$	$2 - 4$
2×4	$2 * 4$
$2 \div 4$	$2 / 4$
x^2	$x ** 2$

المعاملات (operators) في بايثون

المعامل في لغة البرمجة هو رمز يستخدم لإجراء عملية محددة على المتغيرات والقيم. يقدم بايثون فئات مختلفة من المعاملات، الفئات الأربع للمعاملات الأكثر استخدامًا في بايثون هي:

< المعاملات الرياضية.

< معاملات الإسناد.

< المعاملات الشرطية.

< المعاملات المنطقية.

المعاملات (operators) في بايثون:		
المعاملات الرياضية	$+$ $-$ $*$ $/$ $**$	تُستخدم لإجراء العمليات الحسابية: الجمع، الطرح، الضرب، و القسمة وما إلى ذلك.
معاملات الإسناد	$=$ $+=$ $-=$ $*=$ $/=$	تُستخدم لإسناد قيم للمتغيرات.
المعاملات الشرطية	$>$ $<$ $==$ $<=$ $>=$ $!=$	تُستخدم في مقارنة القيم أثناء كتابة الجمل الشرطية.
المعاملات المنطقية	and or not	تُستخدم لفحص أكثر من شرط في جملة شرطية واحدة، وأيضًا لفحص نقيض الشرط. وتُمكن هذه المعاملات من اتخاذ قرارات لجمل شرطية مركبة.





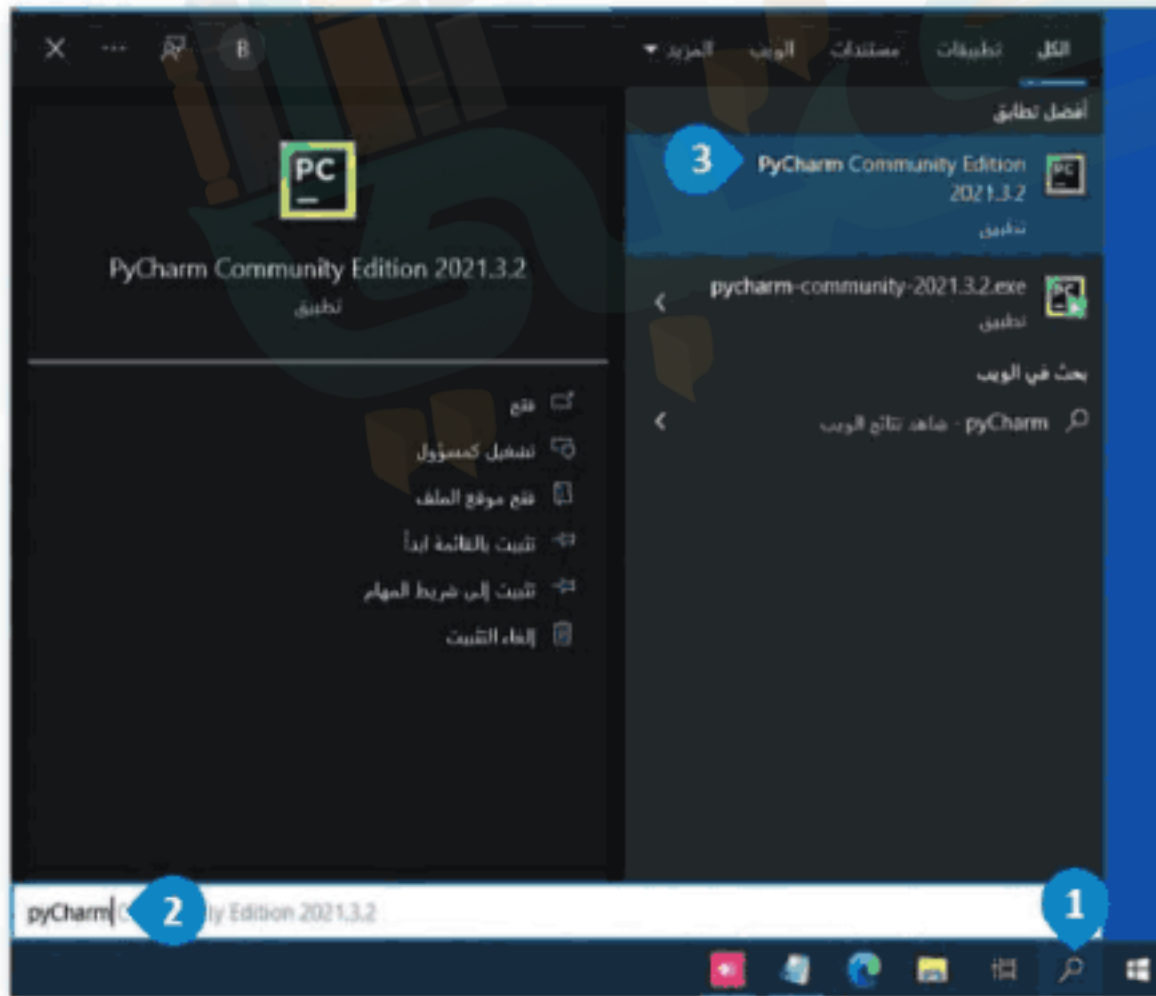
المعاملات الشرطية والمعاملات المنطقية في بايثون

بعد أن تعرفت على لغة برمجة بايثون (Python)، حان الوقت لتتعرف على بيئة برمجية جديدة لكتابة مقاطع البرمجية وهي بيئة التواصل باي تشارم (PyCharm Community).

بيئة التواصل باي تشارم

بيئة التواصل باي تشارم تُستخدم خصيصًا للغة برمجة بايثون. وهي بيئة تطوير متكاملة (Integrated Development Environment - IDE) توفر العديد من الوحدات النمطية والحزم والأدوات لمساعدتك في البرمجة بلغة بايثون. تتوافق بيئة التواصل باي تشارم مع أنظمة تشغيل ويندوز (Windows) ولينكس (Linux) وماك أو إس (macOS). يمكنك تنزيل بيئة التواصل باي تشارم مجانًا من الموقع الإلكتروني:

<https://www.jetbrains.com/pycharm/>



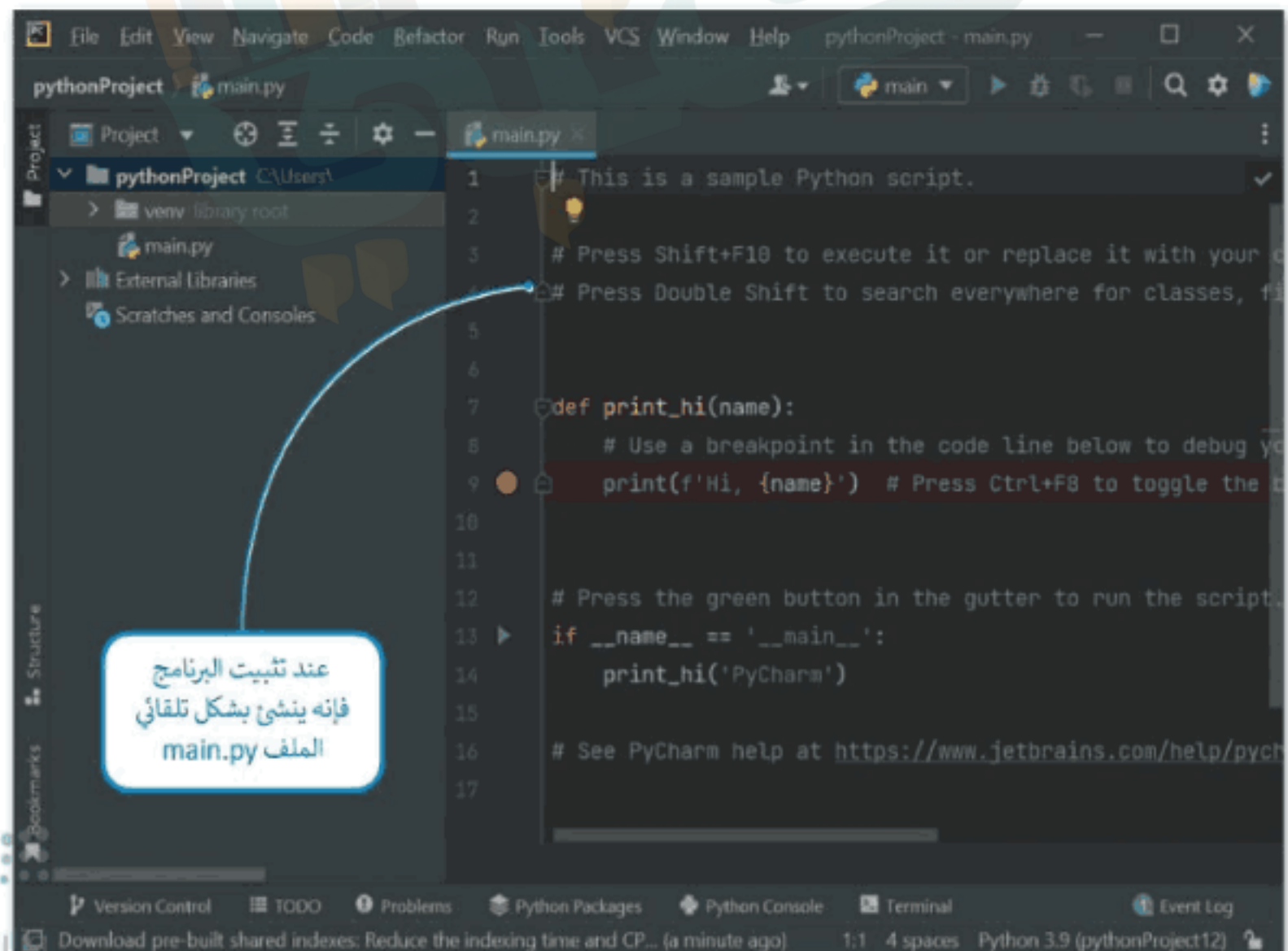
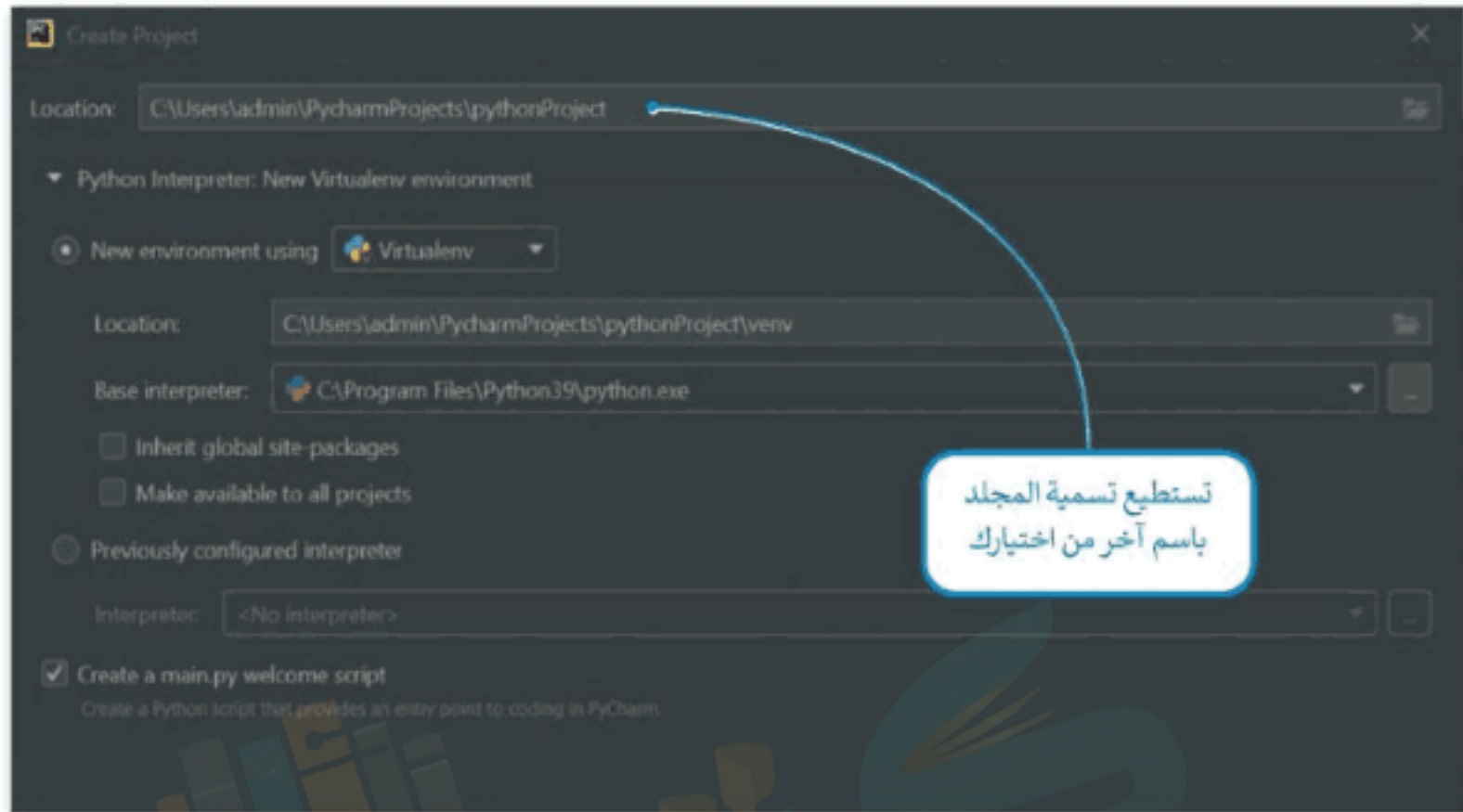
لفتح بيئة التواصل باي تشارم:

- 1 < اضغط على زر Search (بحث).
- 2 < اكتب PyCharm (باي تشارم) في شريط البحث، ثم اضغط عليه.
- 3

معلومة

على الرغم من أن بيئة التطوير المتكاملة (Integrated Development Environment - IDE) تساعد المطور على كتابة المقاطع البرمجية بسهولة إلا أنها أيضًا توفر وظائف وأدوات أكثر تعقيدًا.

من أجل الحفاظ على ملفاتك منظمة، تحتاج إلى مجلد لحفظها. في المرة الأولى التي تقوم فيها بتثبيت بيئة التواصل باي تشارم على جهاز الحاسب الخاص بك، ينشئ البرنامج تلقائيًا مجلد باسم **PythonProject**، وتُحفظ الملفات التي تقوم بإنشائها في هذا المجلد. وبهذه الطريقة يمكنك الوصول بسهولة إلى جميع ملفات بايثون الخاصة بك.

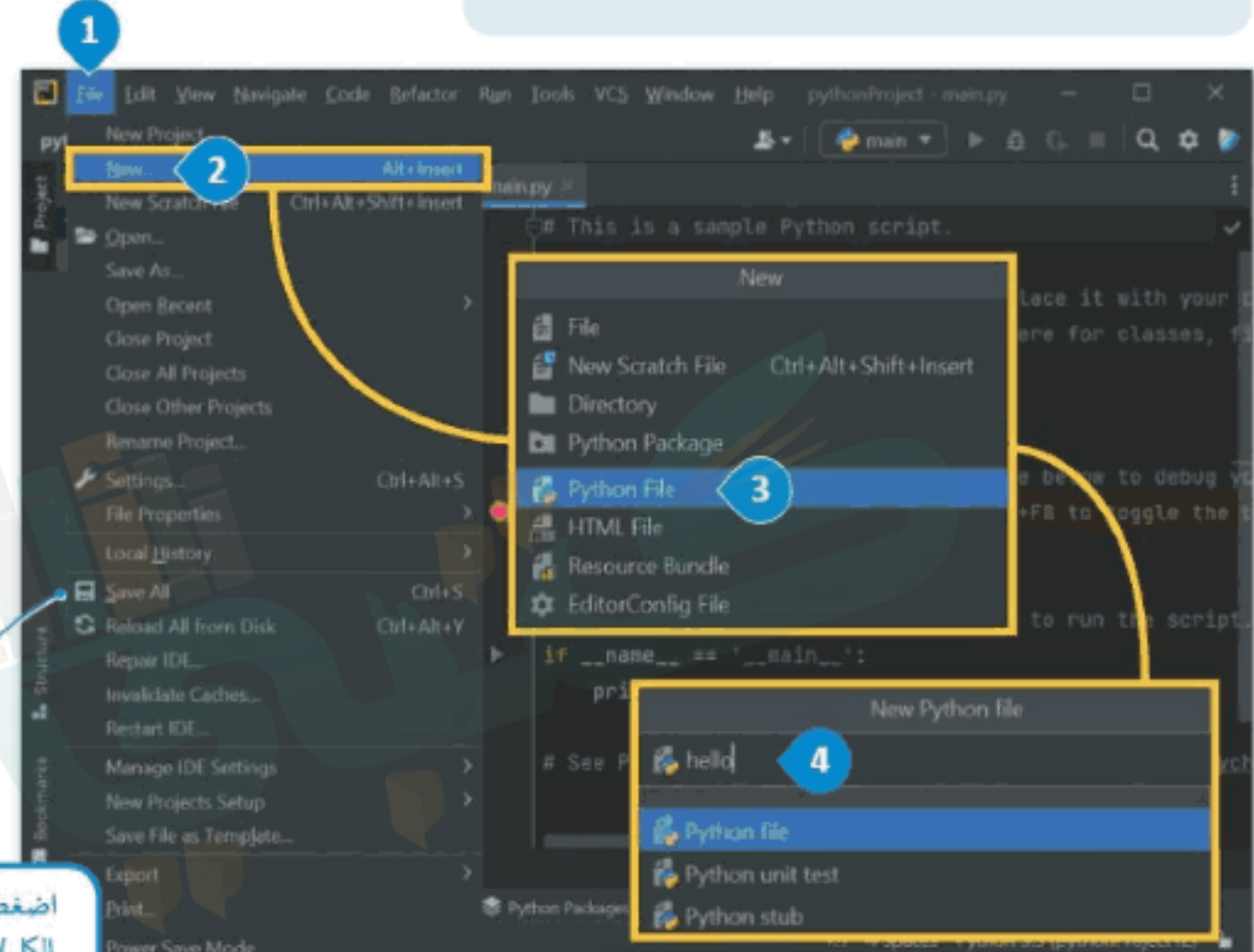


إنشاء ملف بايثون في بيئة التواصل باي تشارم وتشغيله
حان الوقت الآن لإنشاء أول ملف لك بلغة بايثون في بيئة التواصل باي تشارم.

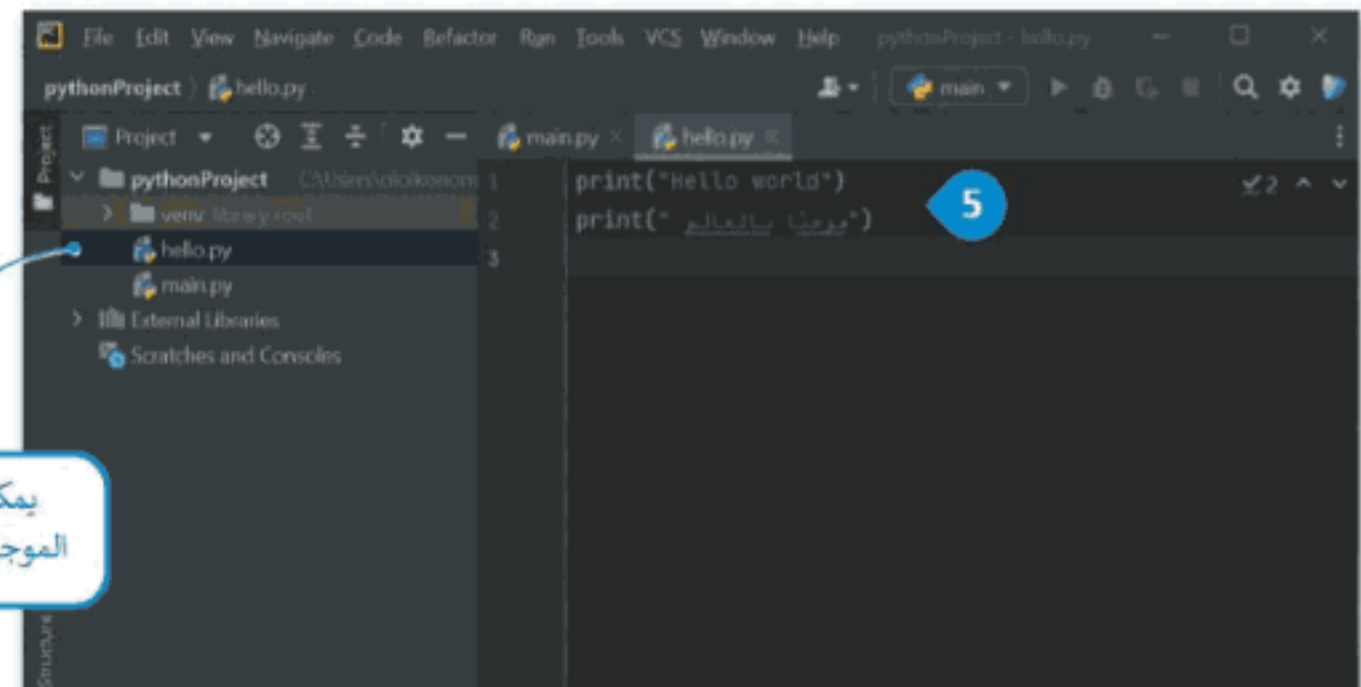
يحفظ باي تشارم بشكل تلقائي التغييرات التي تجريها على ملفاتك.

إنشاء ملف جديد:

- 1 < اضغط على **File** (ملف)، وحدد **New** (جديد).
- 2 < حدد ملف **Python File** (ملف بايثون).
- 3 < اكتب اسمًا للملف. واضغط على زر **Enter**.
- 4 < اكتب أمر بايثون بسيط.

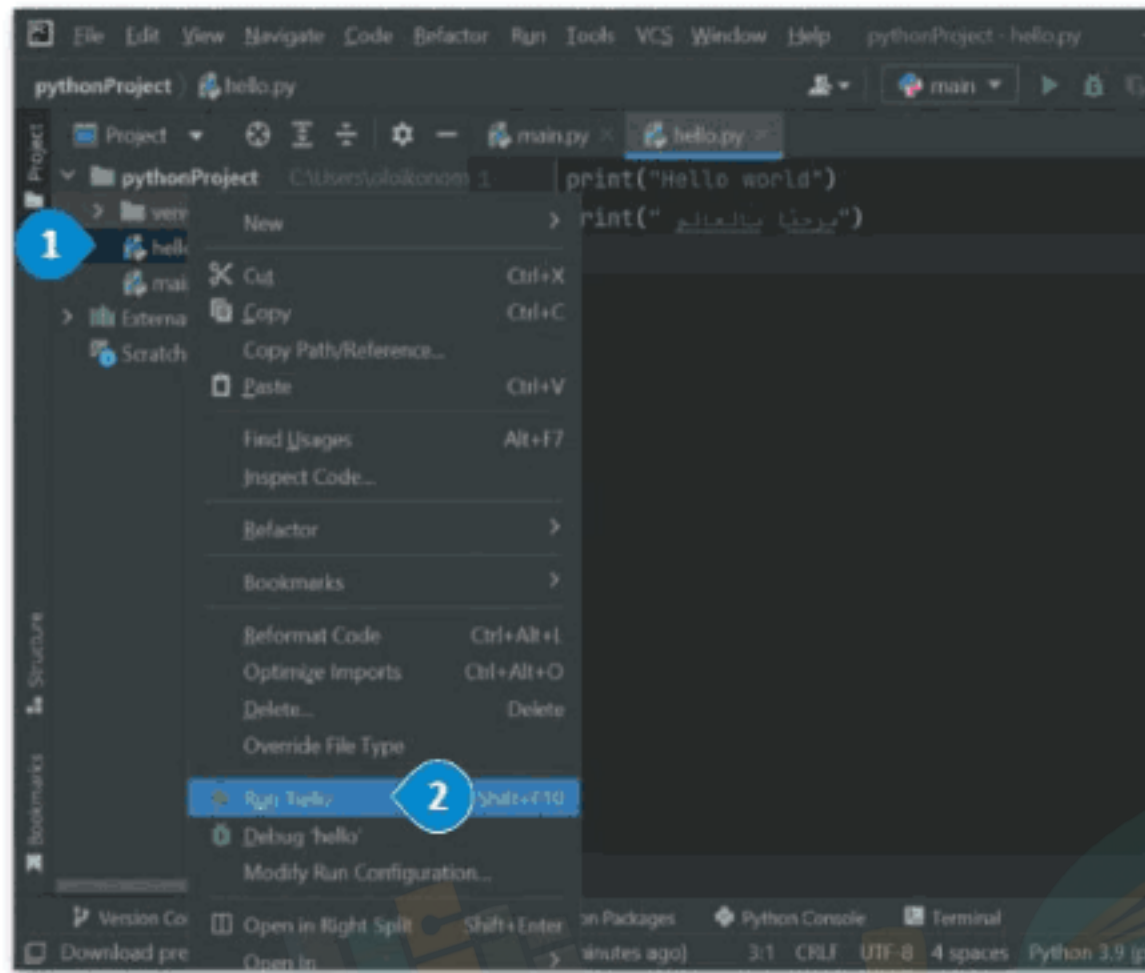


اضغط على **Save All** (حفظ الكل) لحفظ التغييرات يدويًا.



يمكنك هنا رؤية الملفات الموجودة في مجلد مشروعك.

أنت الآن جاهز لتشغيل مقطعك البرمجي الأول في بيئة التواصل باي تشارم .

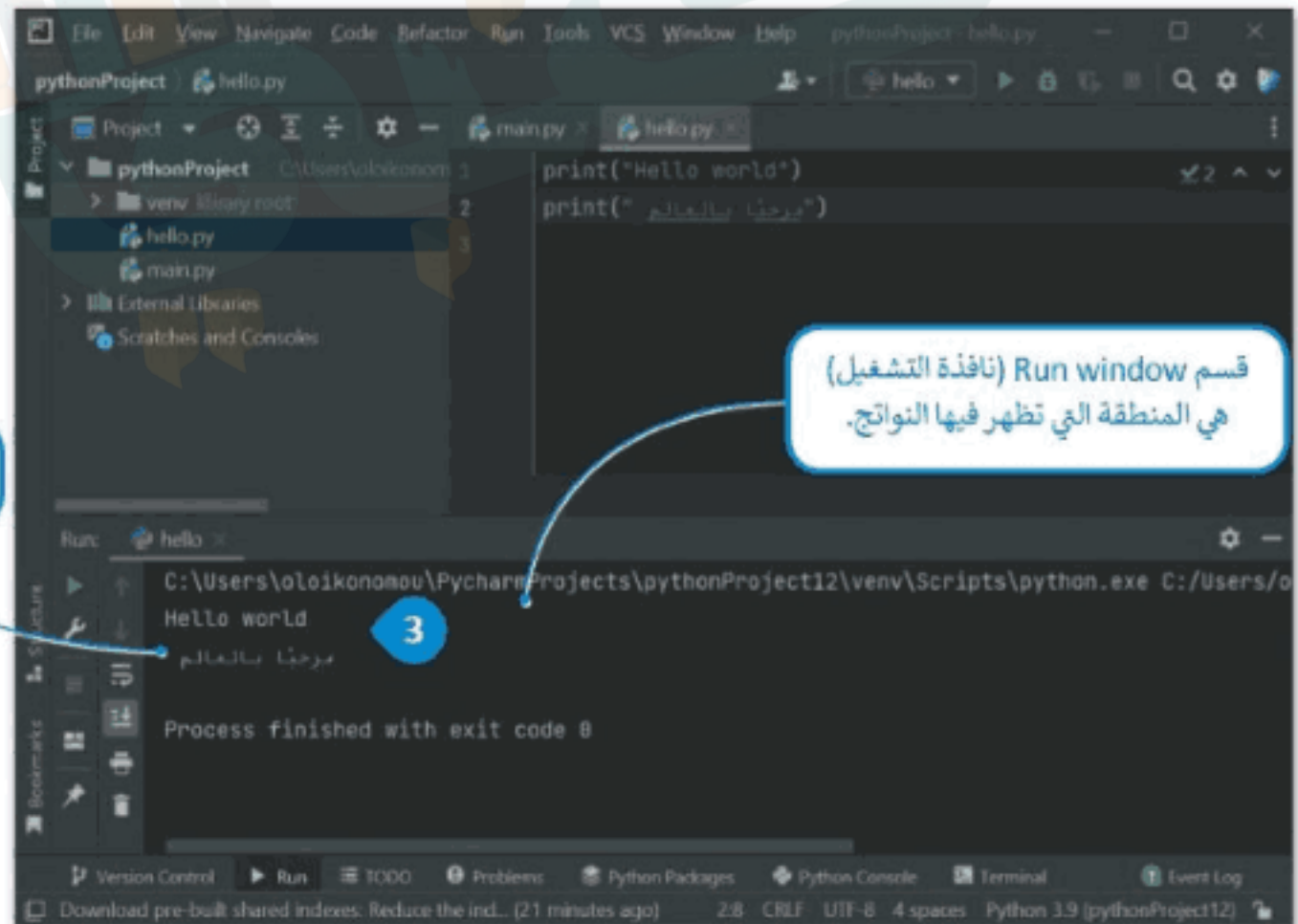


لتشغيل المقطع البرمجي الخاص بك:

< اضغط بزر الفأرة الأيمن على ملفك. 1

< اختر 'file name' Run (تشغيل "اسم الملف"). 2

< تتم معالجة المقطع البرمجي وتظهر النتيجة أسفل الشاشة. 3



قسم Run window (نافذة التشغيل) هي المنطقة التي تظهر فيها النواتج.

النواتج.

معلومة

أثناء عملية المعالجة يتم فحص وكشف الأخطاء في المقطع البرمجي.

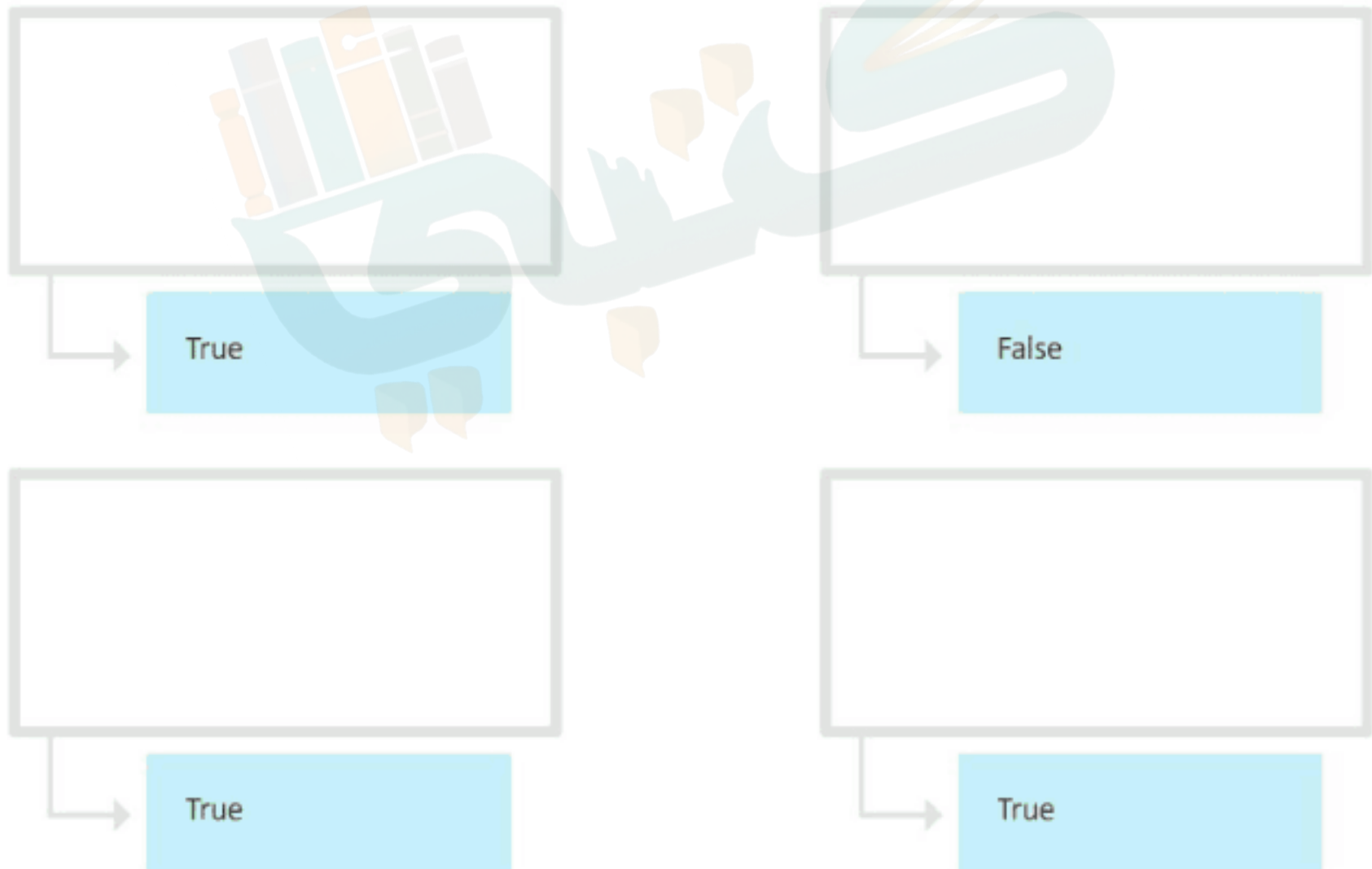
المعاملات الشرطية في بايثون

لاتخاذ قرار في البرمجة، عليك استخدام المعاملات الشرطية. تُستخدم هذه المعاملات للتحكم في مسار المقطع البرمجي. ستجدها عادة كجزء من دالة if أو while، حيث تقارن القيم وتعود بنتيجة صواب (True) أو خطأ (False). وبناءً على نتيجة الفحص، ينفذ المقطع البرمجي العمليات المقابلة.

المعامل	المعنى	المعامل	المعنى
>=	أكبر من أو يساوي	==	يساوي
<=	أصغر من أو يساوي	>	أكبر من
!=	لا يساوي	<	أصغر من

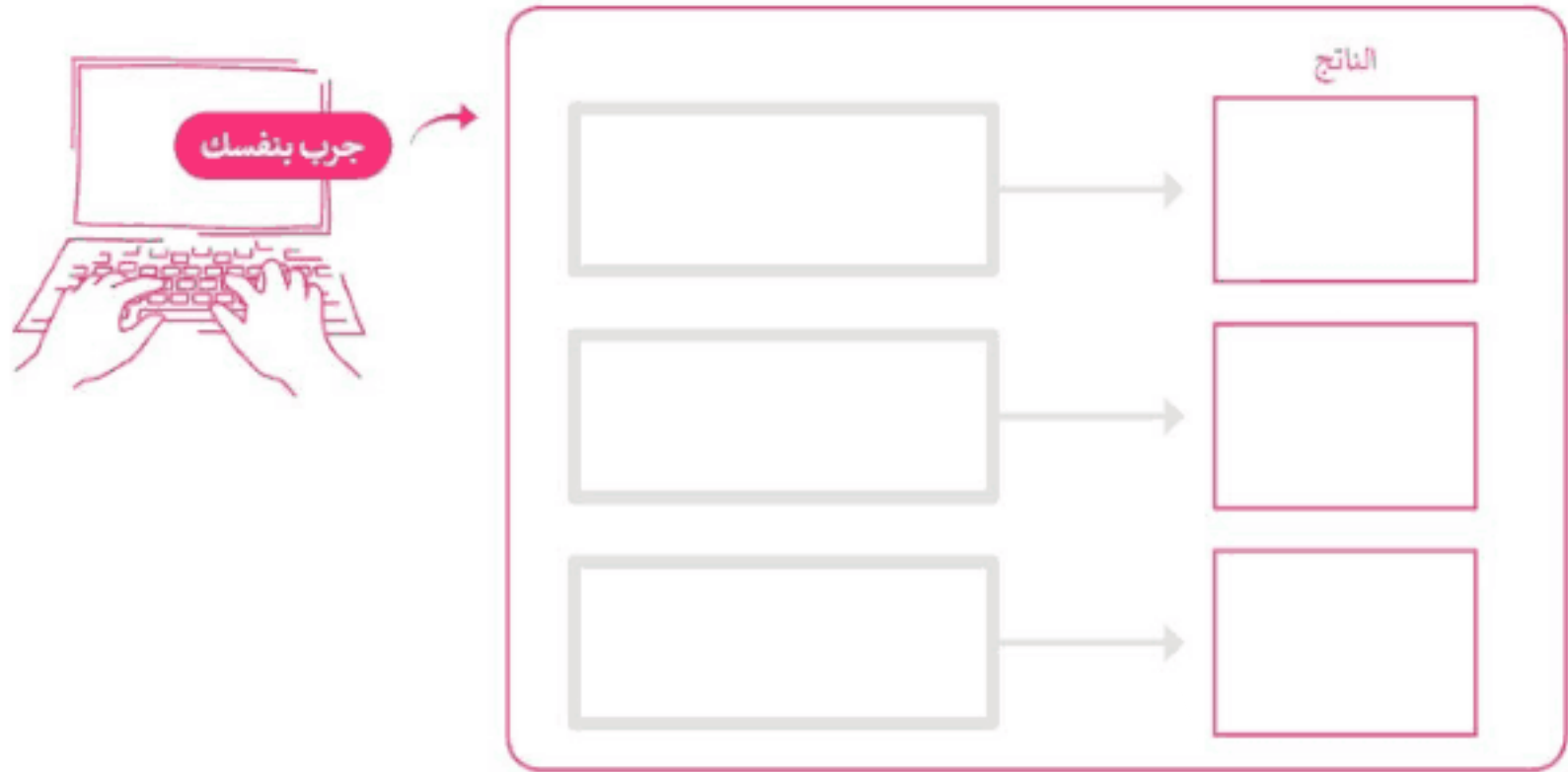
فيما يلي تتم مقارنة قيمتين ويعود المقطع البرمجي بلغة بايثون بالإجابة المنطقية: صواب (True) أو خطأ (False).

مثال 1: المعاملات الشرطية



معلومة

من الأخطاء الشائعة، الخلط بين معاملي الإسناد (=) ومعاملي المساواة (==). وللتفريق بينهما تجد أن معاملي الإسناد سيغير قيمة المتغير، في حين أن معاملي المساواة سيقرأ فقط قيمة المتغير ويرى ما إذا كان يساوي شيئاً ما.



المعامل

المعنى

AND	تعود القيمة صواب (True) إذا كانت كلتا العبارتين صحيحة.
OR	تعود القيمة صواب (True) إذا كانت إحدى العبارات صحيحة.
NOT	تعود القيمة العكسية خطأ إذا كان الناتج صواب، وصواب إذا كان الناتج خطأ.

المعاملات المنطقية في بايثون

قد تحتاج إلى دمج مقارنات متعددة في المعامل الشرطية، في هذه الحالة تُستخدم المعاملات المنطقية، وكذلك تُستخدم لدمج شرطين أو أكثر. من أجل فهم استخدام المعاملات المنطقية بشكل أفضل، يمكنك الرجوع إلى جدول الحقيقة. جدول الحقيقة هو جدول يسرد جميع المدخلات الممكنة للمتغير، ويعرض مخرجاته وفقاً للمعامل المنطقي كما يظهر بالأسفل:

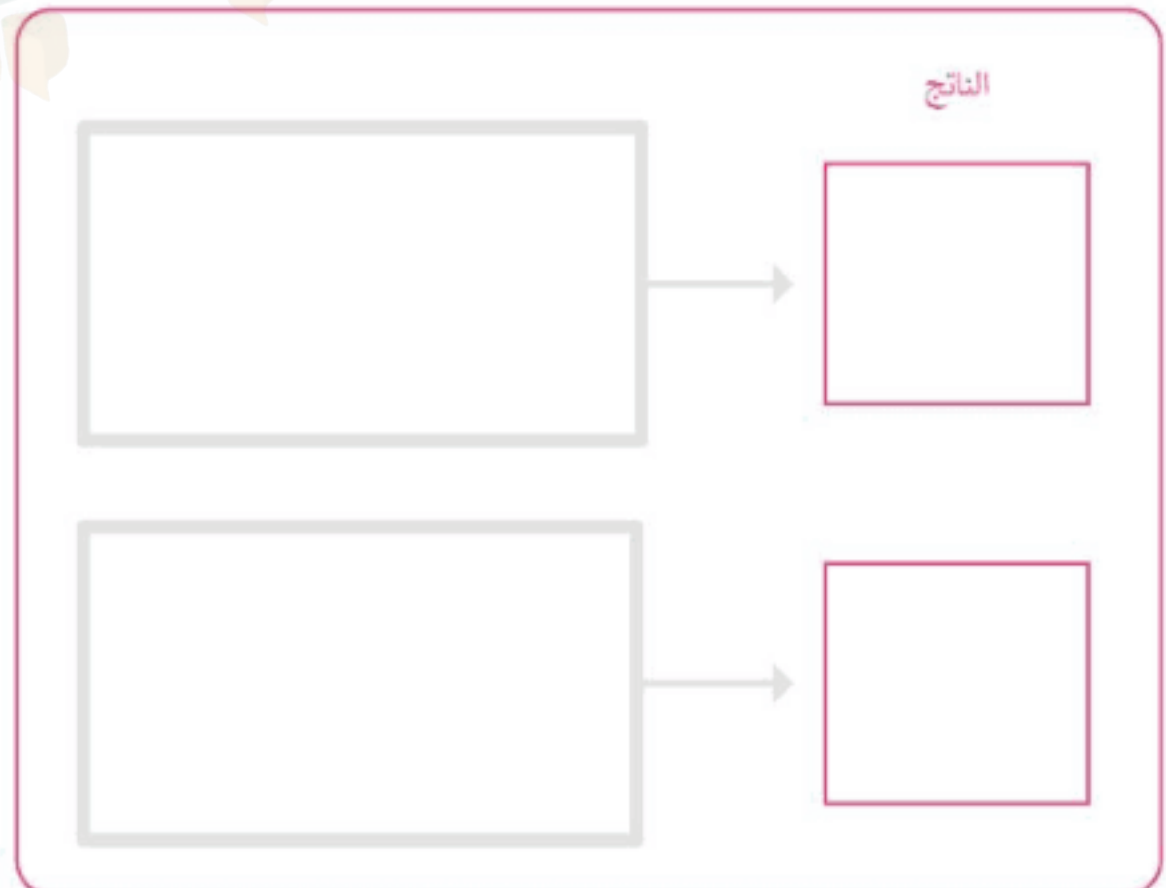
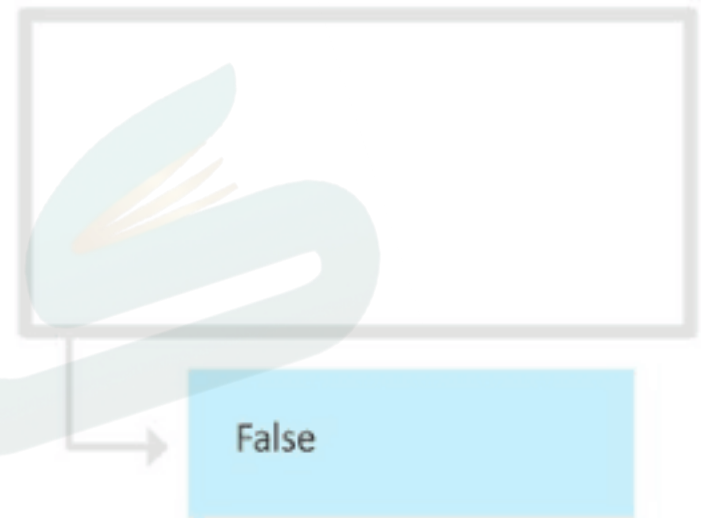
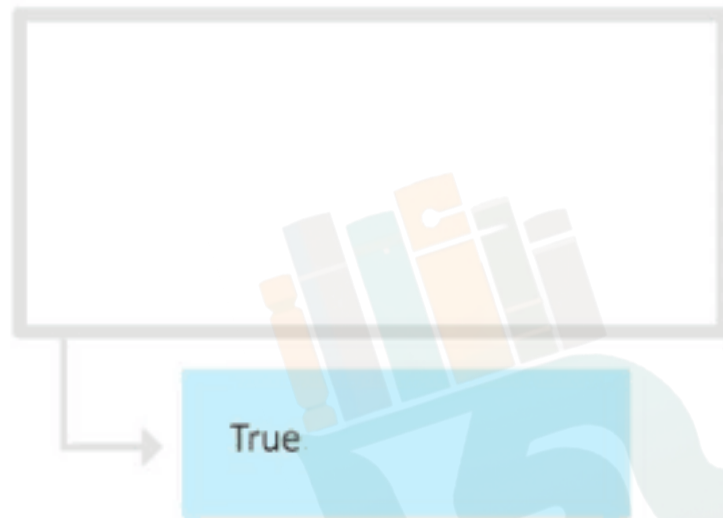
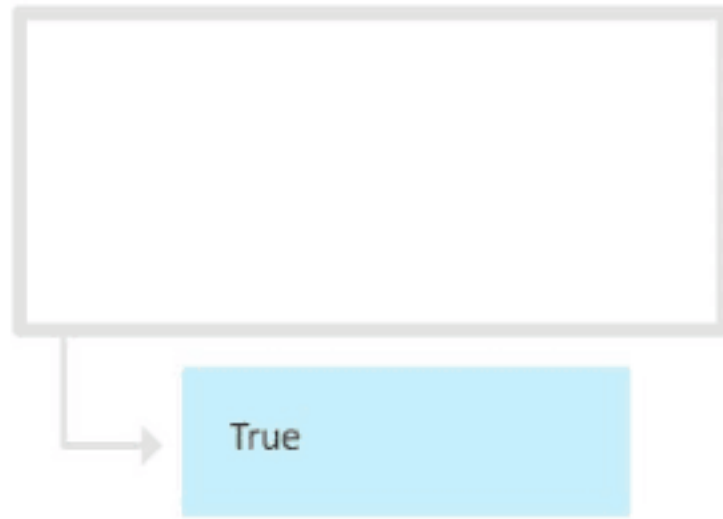
جدول الحقيقة

ترتيب المعاملات في البرمجة



مثال 2: المعاملات المنطقية

في الأمثلة التالية سترى كيفية استخدام العوامل المنطقية في المقاطع البرمجية.



لنطبق معًا

تدريب 1

👉 صل كل معامل مع فئته.

+	1
=	2
<	3
+=	2
==	3
or	4
!=	3
not	4

1 المعاملات الرياضية

2 معاملات الإسناد

3 المعاملات الشرطية

4 المعاملات المنطقية



تدريب 2

اكتب المعامل الصحيح بجانب الوصف الخاص به.

1. يساوي	<code>==</code>
2. أكبر من	<code>></code>
3. أصغر من أو يساوي	<code><=</code>
4. لا يساوي	<code>!=</code>

تدريب 3

اكتب ناتج العمليات التالية:

False	<code>x = a > b and b > c</code>	<code>a=2</code>
True	<code>y = (a+b) < c or (b+c) < a</code>	<code>b=5</code>
False	<code>z = not(a > 0 or b < 0)</code>	<code>c=10</code>



تدريب 4

استخدم بيئة التواصل باي تشارم لكتابة مقطع برمجي يقوم بقراءة رقمين ويحسب حاصل الجمع وكذلك حاصل الطرح وحاصل الضرب لهما ثم اكتبه:

المقطع البرمجي لإيجاد حاصل جمع الرقمين

المقطع البرمجي لإيجاد حاصل طرح الرقمين

المقطع البرمجي لإيجاد حاصل ضرب الرقمين

Num1= 8

Num2= 3

اكتب الأوامر البرمجية هنا.

$x = \text{num1} + \text{num2}$

اكتب الأمر البرمجي الذي يجمع الرقمين ويظهر الناتج

$y = \text{num1} - \text{num2}$

اكتب الأمر البرمجي الذي يطرح الرقمين ويظهر الناتج

$z = \text{num1} * \text{num2}$

اكتب الأمر البرمجي الذي يضرب الرقمين ويظهر الناتج





الجميل الشرطية في البايثون

تعرفت على تنفيذ الأوامر بالتتابع واحدًا تلو الآخر، إلا أن اتخاذ القرار أحيانًا يكون تبعًا لطبيعة المشكلة. إن اتخاذ القرار عملية مهمة وقوية متاحة لكل مبرمج يقوم بتطوير البرنامج. ستتعلم في هذا الدرس كيفية إنشاء مقاطع برمجية تستجيب لمدخلات من المستخدم أثناء تنفيذها وتعطي نتائج مختلفة تبعًا للمدخلات المختلفة. ويتحقق ذلك باستخدام الجمل الشرطية.

الجميل الشرطية في بايثون

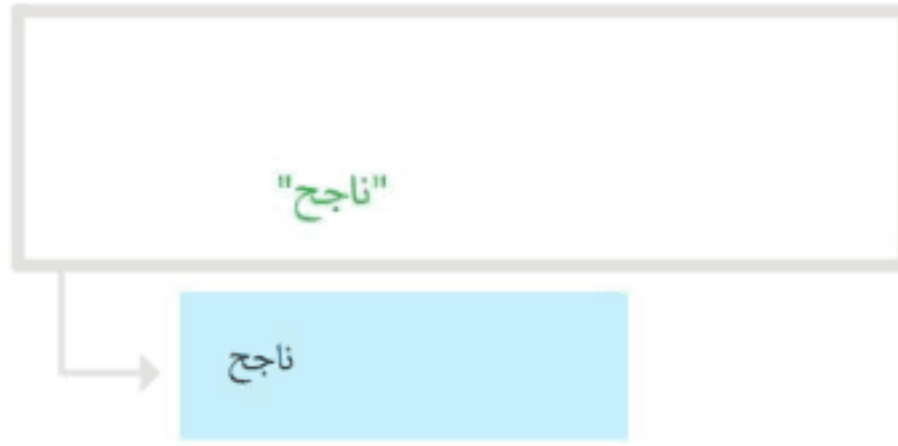
لاتخاذ القرارات في لغة برمجة بايثون: استخدم الجملة الشرطية `if`. ويوجد ثلاثة أنواع من الجمل الشرطية كما في الشكل التالي:

أنواع الجمل الشرطية



أمثلة على أنواع الجمل الشرطية

حان الوقت لتشاهد مثالاً.



الشرط: if

العبارة



الشرط: if

عبارة 1

else:

عبارة 2



الشرط: if

عبارة 1

elif:

عبارة 2

else:

عبارة 3

on: الشرط:

العبارة

لاحظ أن النقطتين (:) اللتين
تليان التعبير الشرطي ضروريتان.



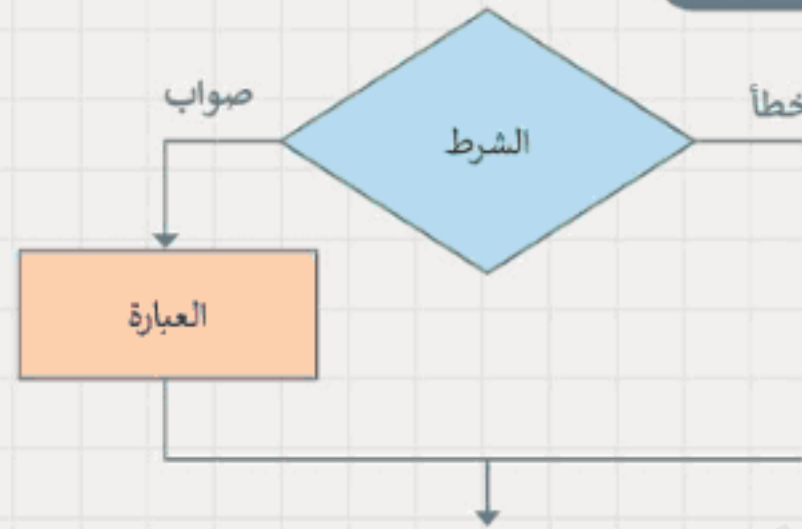
جملة if الشرطية البسيطة

الشرط: if

العبارة

< إذا تحقق الشرط فستنفذ العبارة أو العبارات التي تتبع if.
< إذا لم يتحقق الشرط فلن تنفذ العبارة أو العبارات التي تتبع if.

المخطط الانسيابي للمقطع البرمجي



المسافة البادئة

المسافة البادئة (Indentation) في لغة البرمجة بايثون مهمة جداً، إذ يستخدمها المقطع البرمجي للإشارة إلى العبارات المعتمدة على تحقيق الشرط. إذا لم تترك مسافة بادئة، فستلقى رسالة خطأ. الصفوف غير المزودة بمسافة بادئة والتي لا يعتمد تنفيذها على نتيجة الشرط، ستنفذ حتى إذا لم يتحقق الشرط.



IndentationError: expected an indented block



لا ينتمي أمر الطباعة الثاني إلى جملة if لذلك يُنفذ الأمر دون النظر إلى نتيجة جملة if.

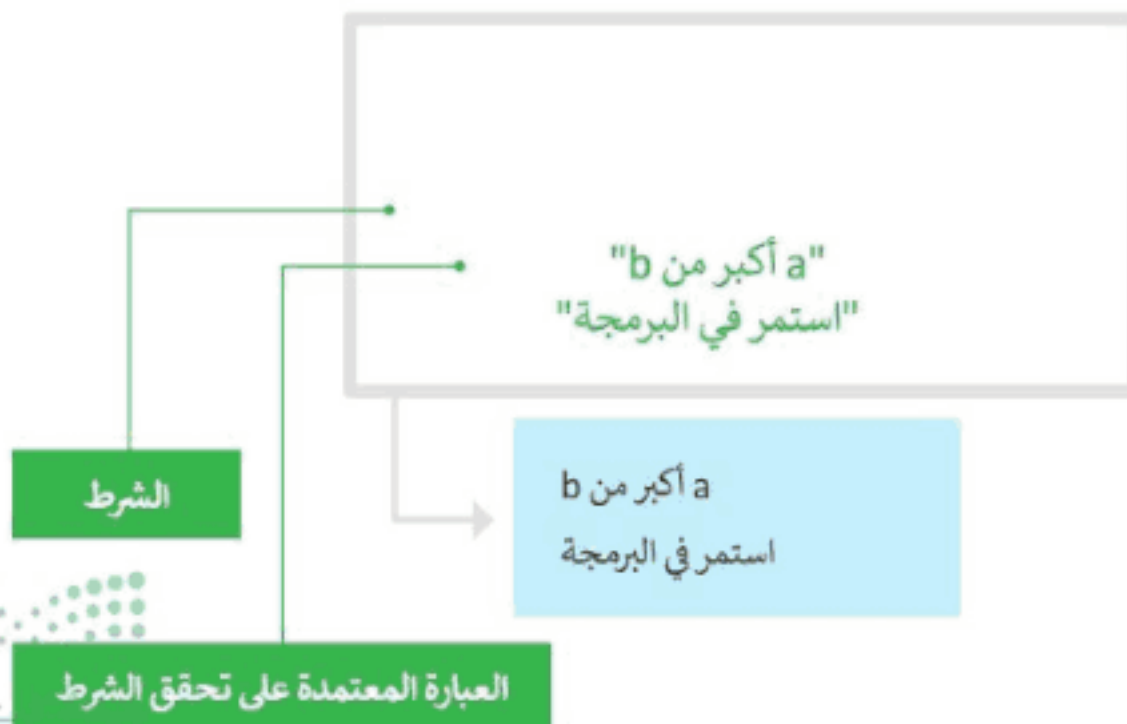
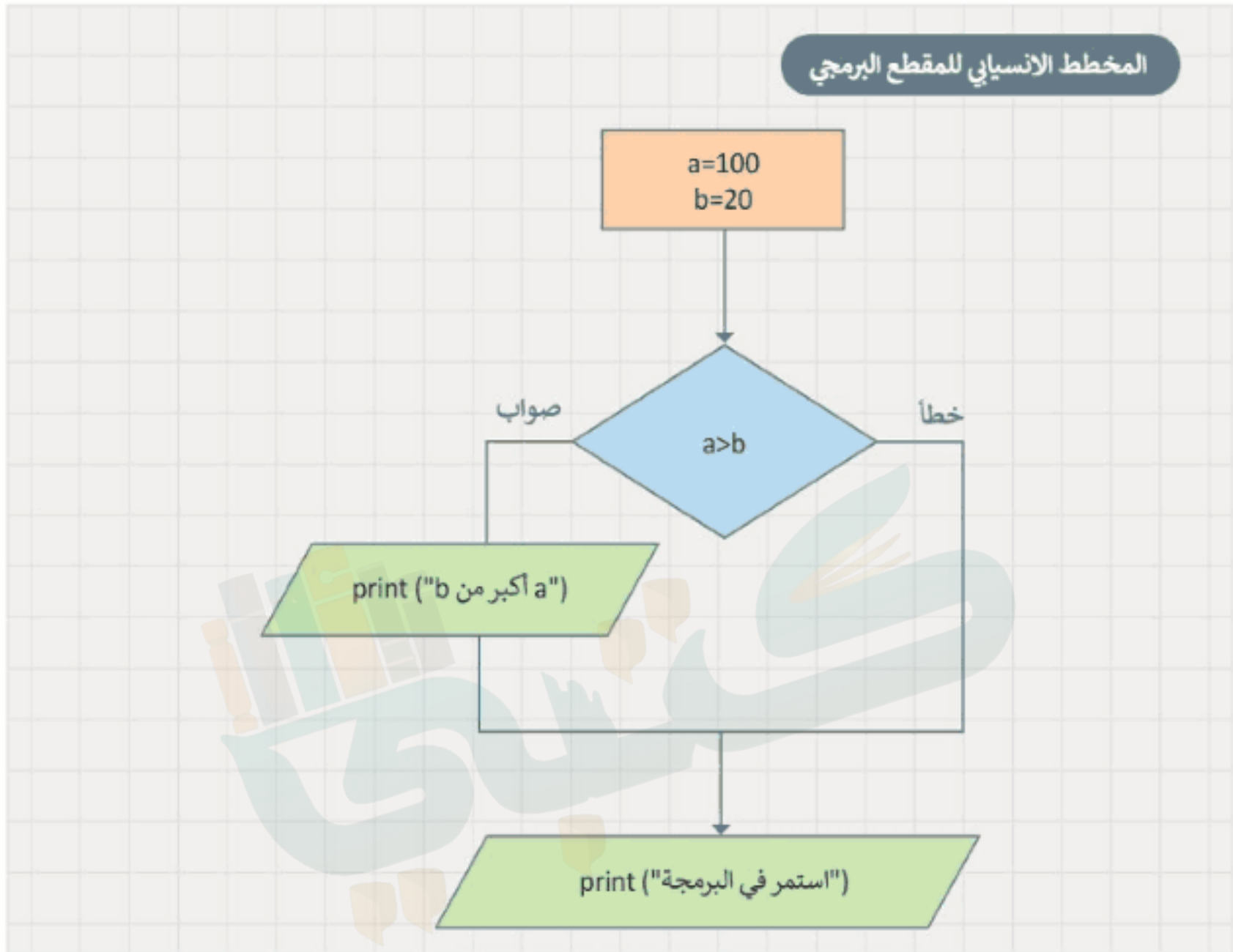
"ممتاز"
استمر في تطوير مهاراتك

ممتاز
استمر في تطوير مهاراتك

حان الوقت لتشاهد بعض الأمثلة.

مثال 1: مقارنة رقمين

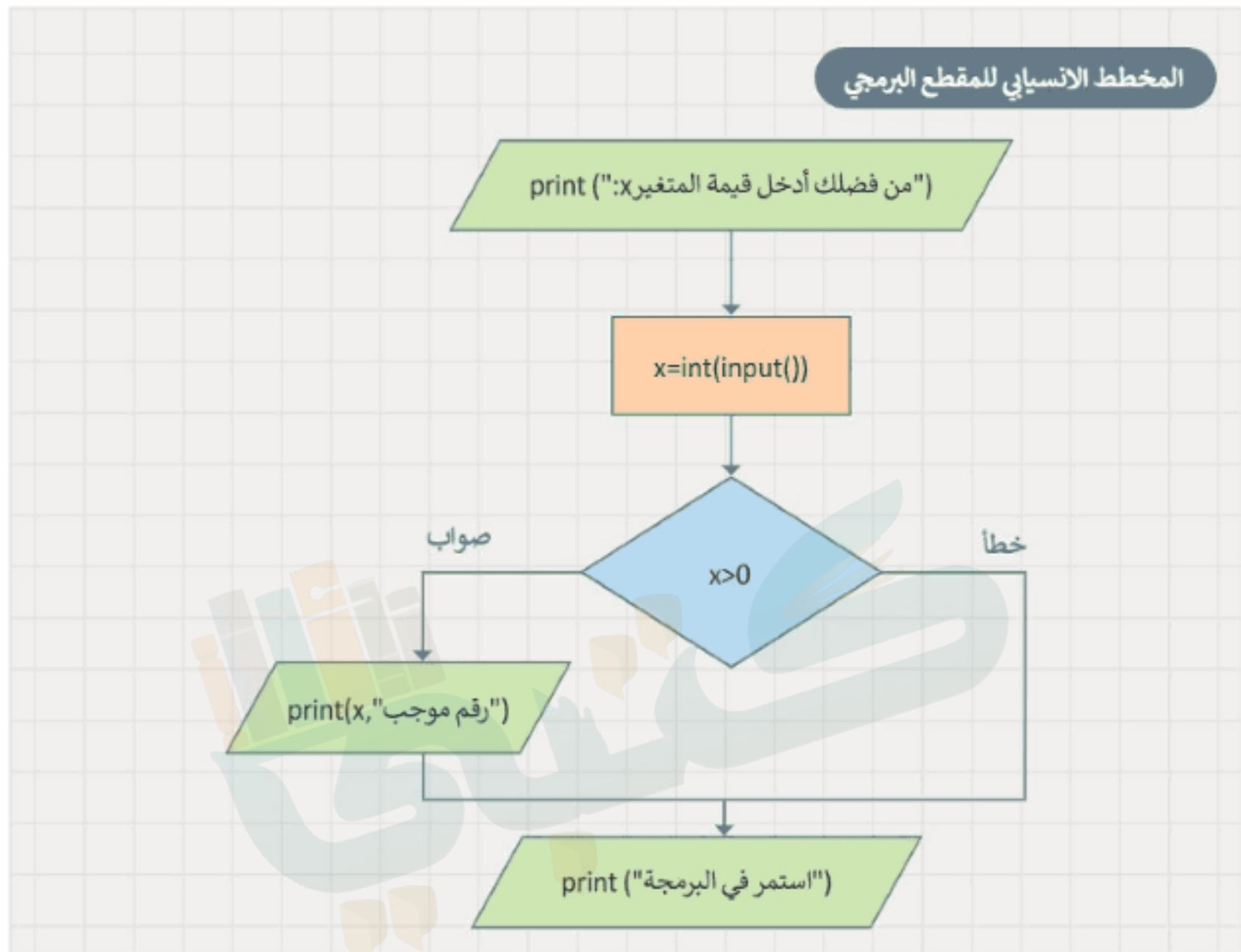
يقارن المقطع البرمجي قيمة الرقمين ثم يطبع أيهما أكبر.



مثال آخر على جملة if البسيطة.

مثال 2: رقم موجب

يقرأ المقطع البرمجي رقمًا ويتحقق مما إذا كان الرقم موجبًا.



لنطبق معًا

تدريب 1

ماذا يحدث عند تشغيل المقطع البرمجي التالي:

اختر الإجابة الصحيحة	
1.	<pre>Number = 12 if Number > 0 : print("رقم موجب")</pre>
☐	لن يعمل المقطع البرمجي لوجود خطأ في صيغة الأوامر.
☐	لن تُعرض أي رسالة على الشاشة لأن الشرط لم يتحقق.
☒	ستعرض الرسالة "رقم موجب" على الشاشة.
2.	<pre>Number != 12 if Number > 0 : print("رقم موجب")</pre>
☒	لن يعمل المقطع البرمجي لوجود خطأ في صيغة الأوامر.
☐	لن تُعرض أي رسالة على الشاشة لأن الشرط لم يتحقق.
☐	ستعرض الرسالة "رقم موجب" على الشاشة.

تدريب 2

اكتب ناتج المقطع البرمجي التالي:

```
a = -7
if a < 0 :
    a = a*(-1)
print(a)
```

الناتج

7



تدريب 3

اكتب ناتج المقطع البرمجي التالي:
مستخدمًا القيم: a=5, b=12, c=9

```
print("من فضلك أدخل قيمة a و b و c:")  
a=int(input())  
b=int(input())  
c=int(input())  
if a > b and a > c :  
    m = a  
if b > a and b > c :  
    m = b  
if c > a and c > b :  
    m = c  
print(m)
```

12

ما وظيفة المقطع البرمجي؟
يطبع الرقم الأكبر

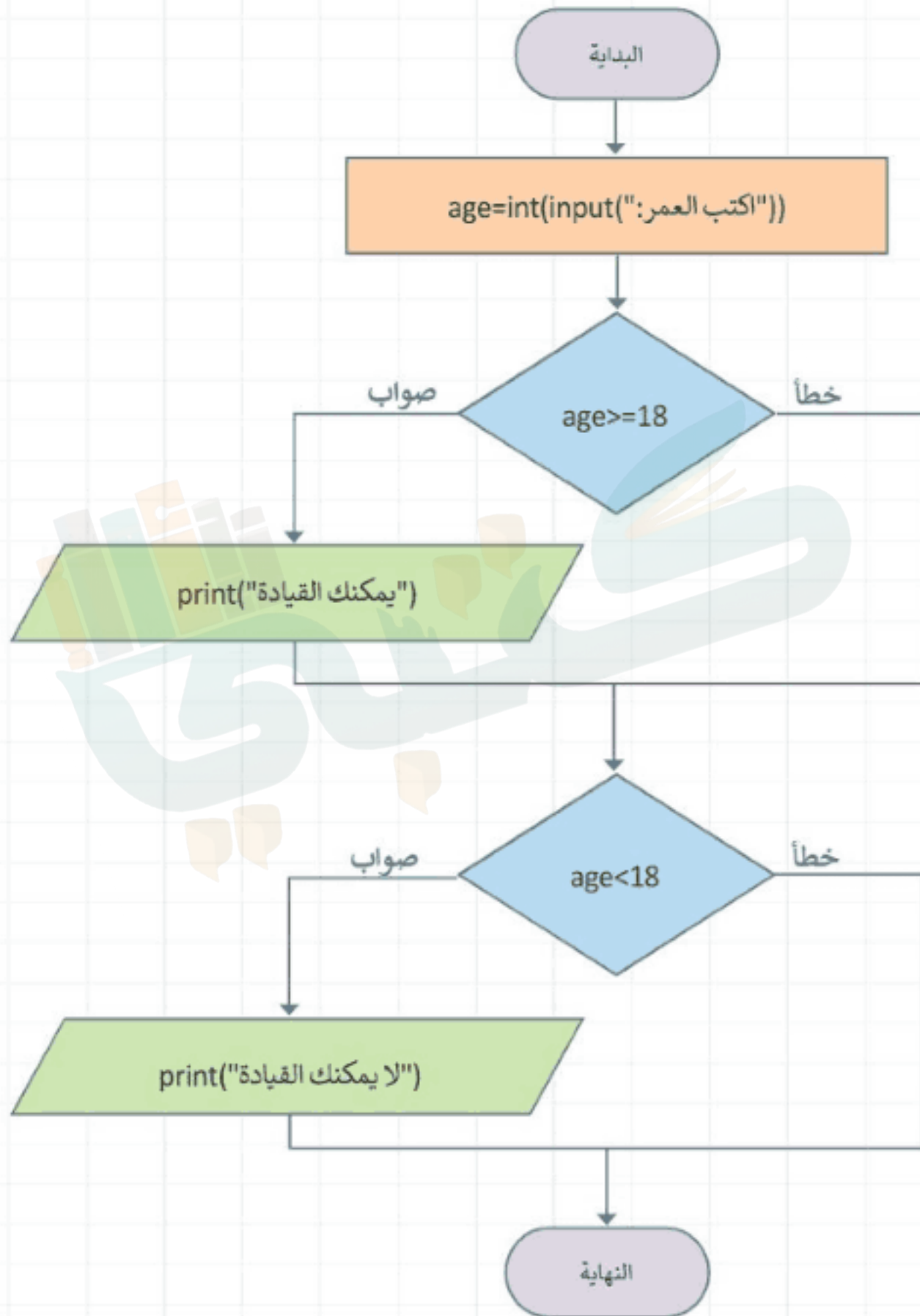
تدريب 4

ارسم المخطط الانسيابي لمقطع برمجي يتحقق من كون عمرك مناسبًا لقيادة السيارة، ثم اكتب المقطع البرمجي.

```
age=int(input("اكتب العمر:"))  
if age>=18:  
    print("يمكنك القيادة")  
  
if age< 18:  
    print("لا يمكنك القيادة")
```



المخطط الانسيابي للبرنامج





الدرس الثالث: اتخاذ القرارات

بعد أن تعلمت كيفية كتابة الجمل الشرطية البسيطة في لغة البرمجة بايثون، وتعرفت على ما سيحدث إذا تحقق الشرط، حان الوقت لتتعلم ما سيحدث عندما لا يتحقق الشرط.

جملة if...else الشرطية

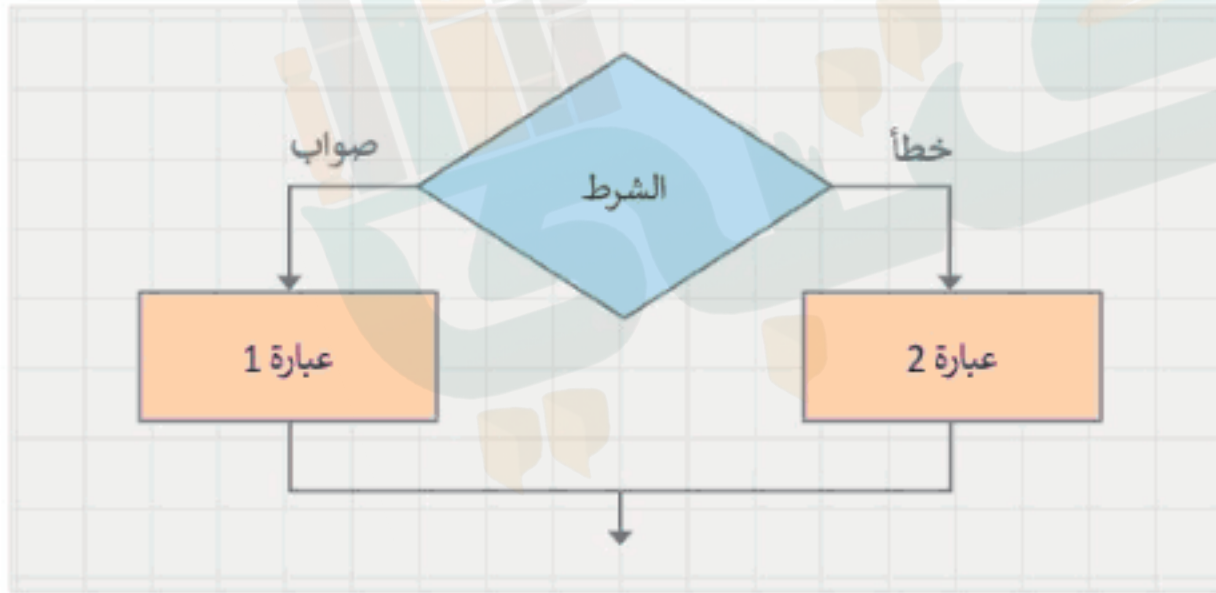
الشرط: if

عبارة 1

else:

عبارة 2

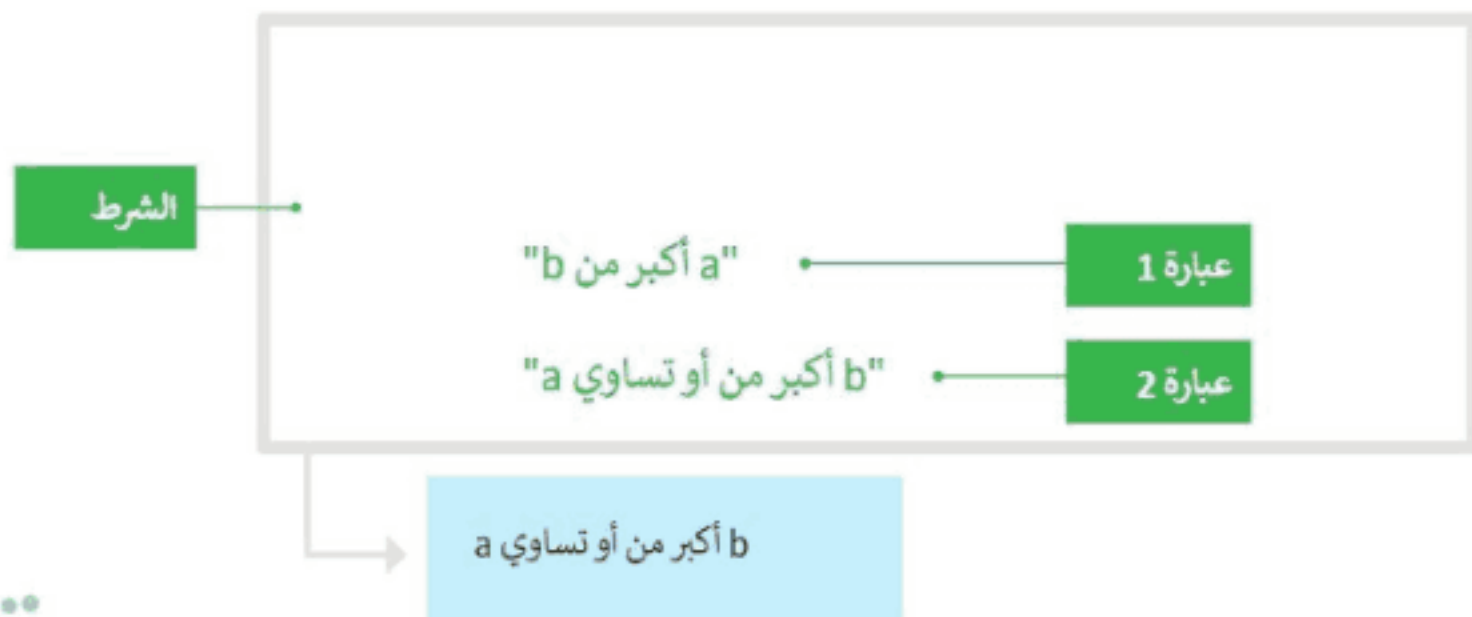
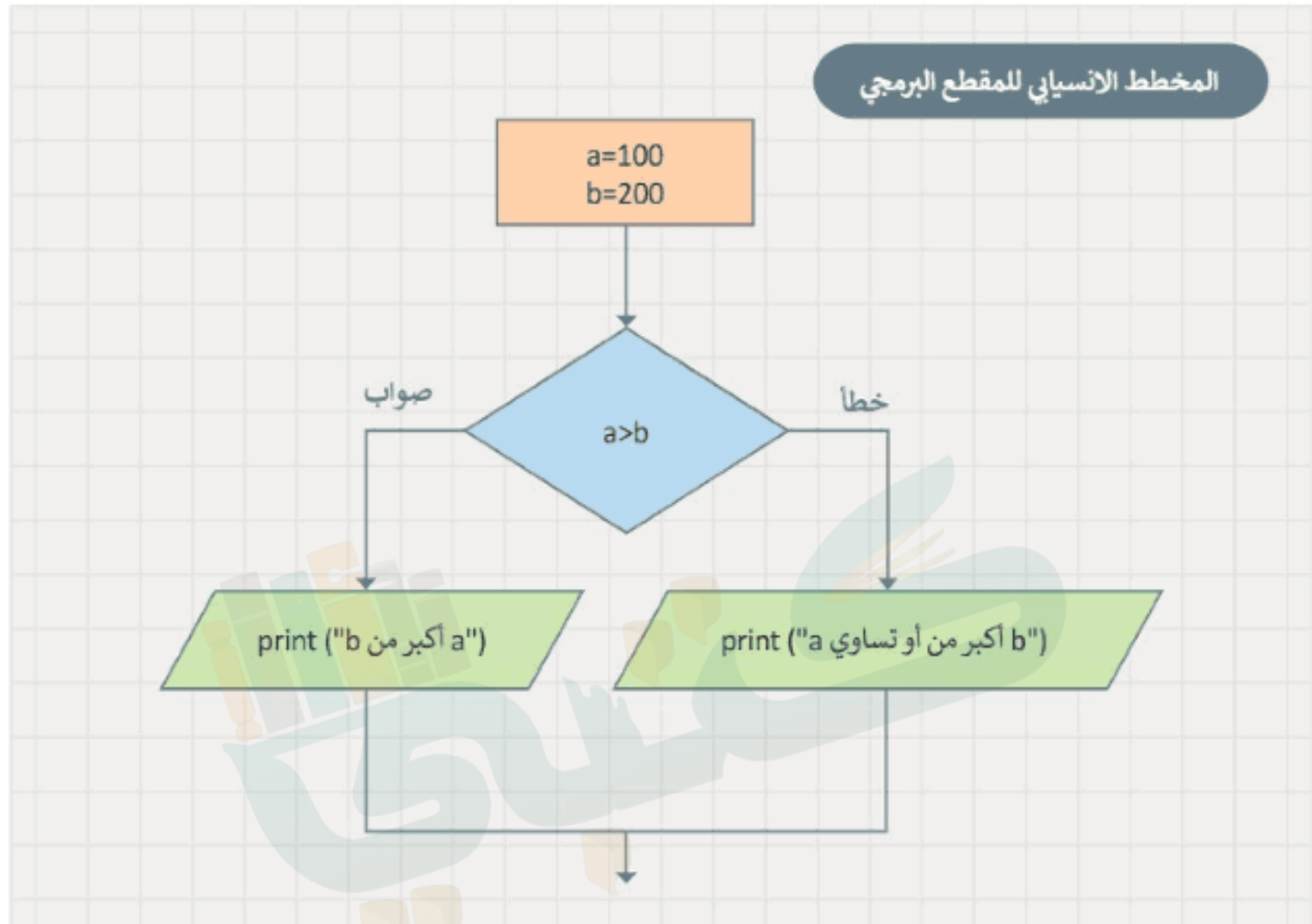
ستتعلم في هذا الدرس كيفية استخدام جملة if ... else الشرطية. إذا تحقق الشرط فسيتم تنفيذ عملية أو عمليات محددة، وإذا لم يتحقق فسيتم تنفيذ عملية أو عمليات أخرى محددة. كما في الحالة السابقة، يتم استخدام المسافة البادئة لتحديد العبارات التي ستنفذها كل مرة.



حان الوقت لتشاهد بعض الأمثلة.

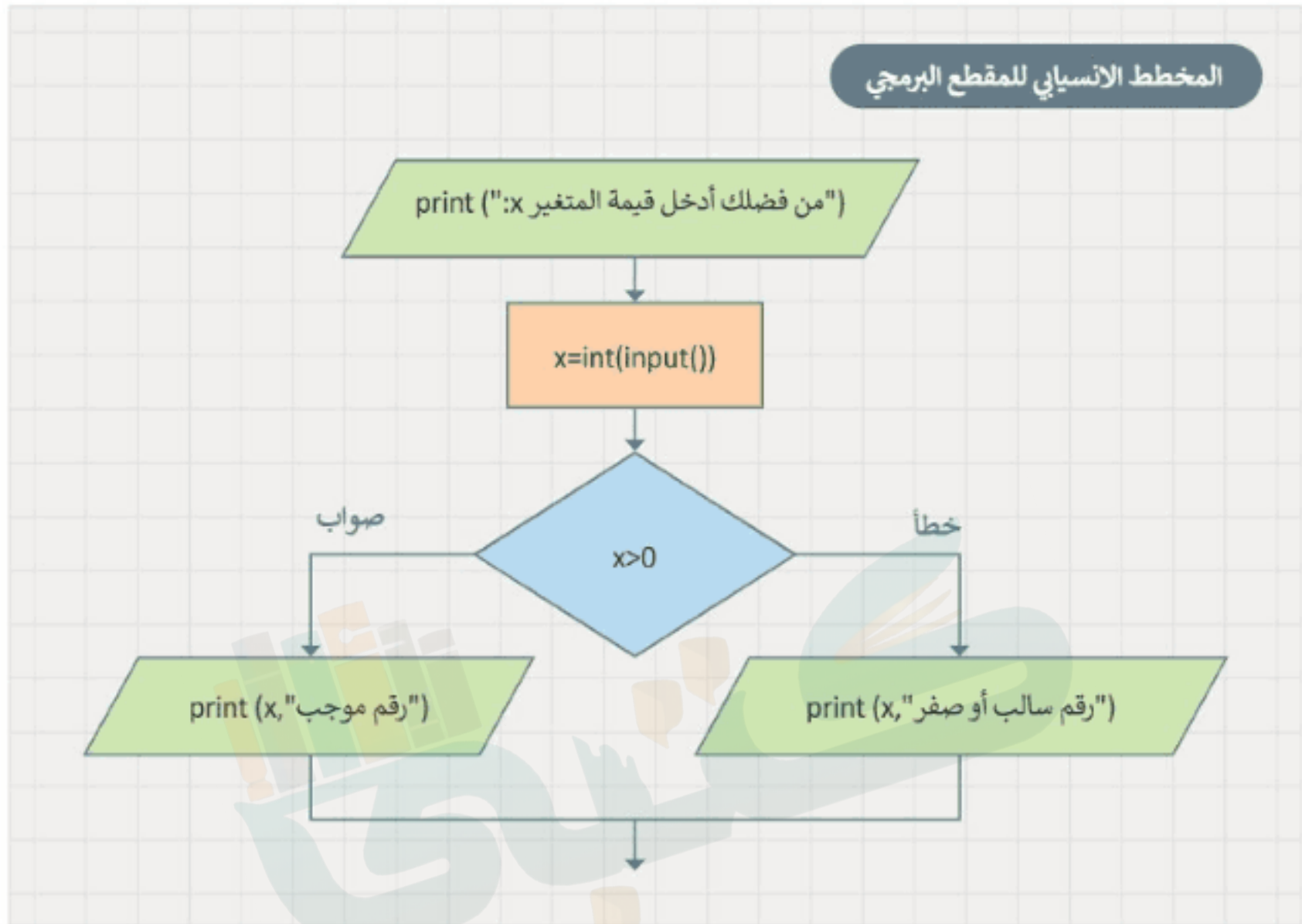
مثال 1: مقارنة رقمين

يقارن المقطع البرمجي قيمة الرقمين ثم يطبع أيهما أكبر.



مثال 2: موجب أو سالب

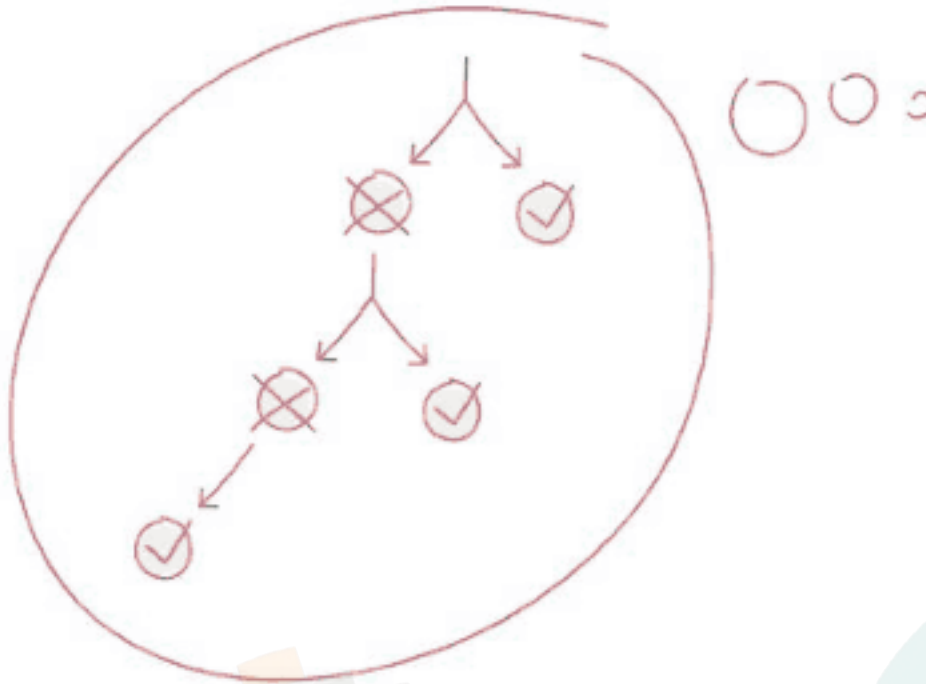
يقرأ المقطع البرمجي الرقم ويتحقق مما إذا كان الرقم موجباً أم سالباً ثم يطبع الرسالة المقابلة.



جملة if...elif

في الجمل الشرطية السابقة كان على المستخدم اختيار أحد خيارين، أما في هذا النوع من جمل if الشرطية، فإن المستخدم يجب أن يحدد خياراً من بين خيارات متعددة، تنفذ عبارات if من الأعلى إلى الأسفل.

يتحقق المقطع البرمجي من الشروط واحداً تلو الآخر، فإذا تحقق أحد الشروط، يتم تنفيذ ما تحت هذا الشرط ويتجاوز باقي الشروط، أما إذا لم يتحقق أيًا من الشروط، فستنفذ جملة else.



الشرط الأول if:

عبارة 1

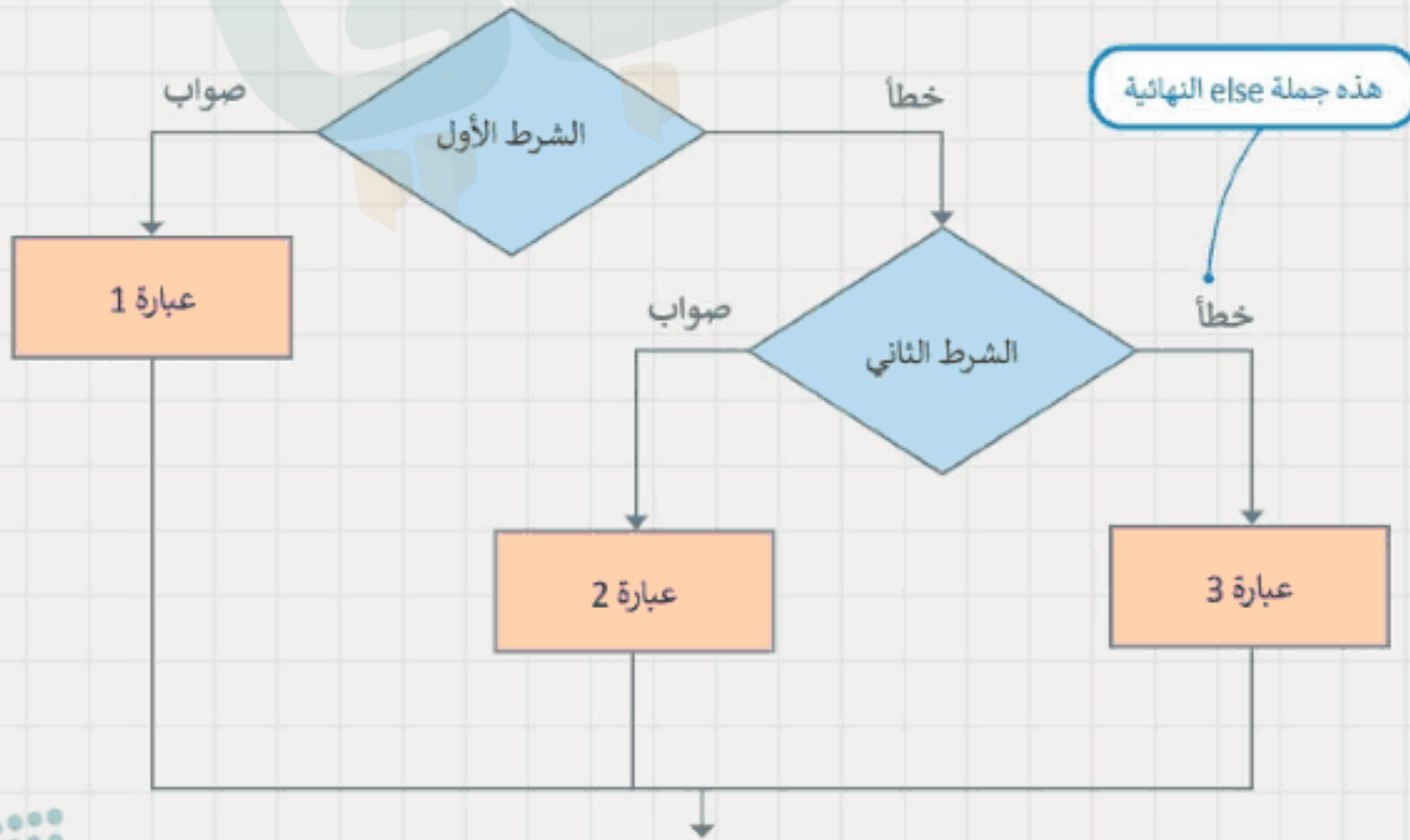
الشرط الثاني elif:

عبارة 2

else:

عبارة 3

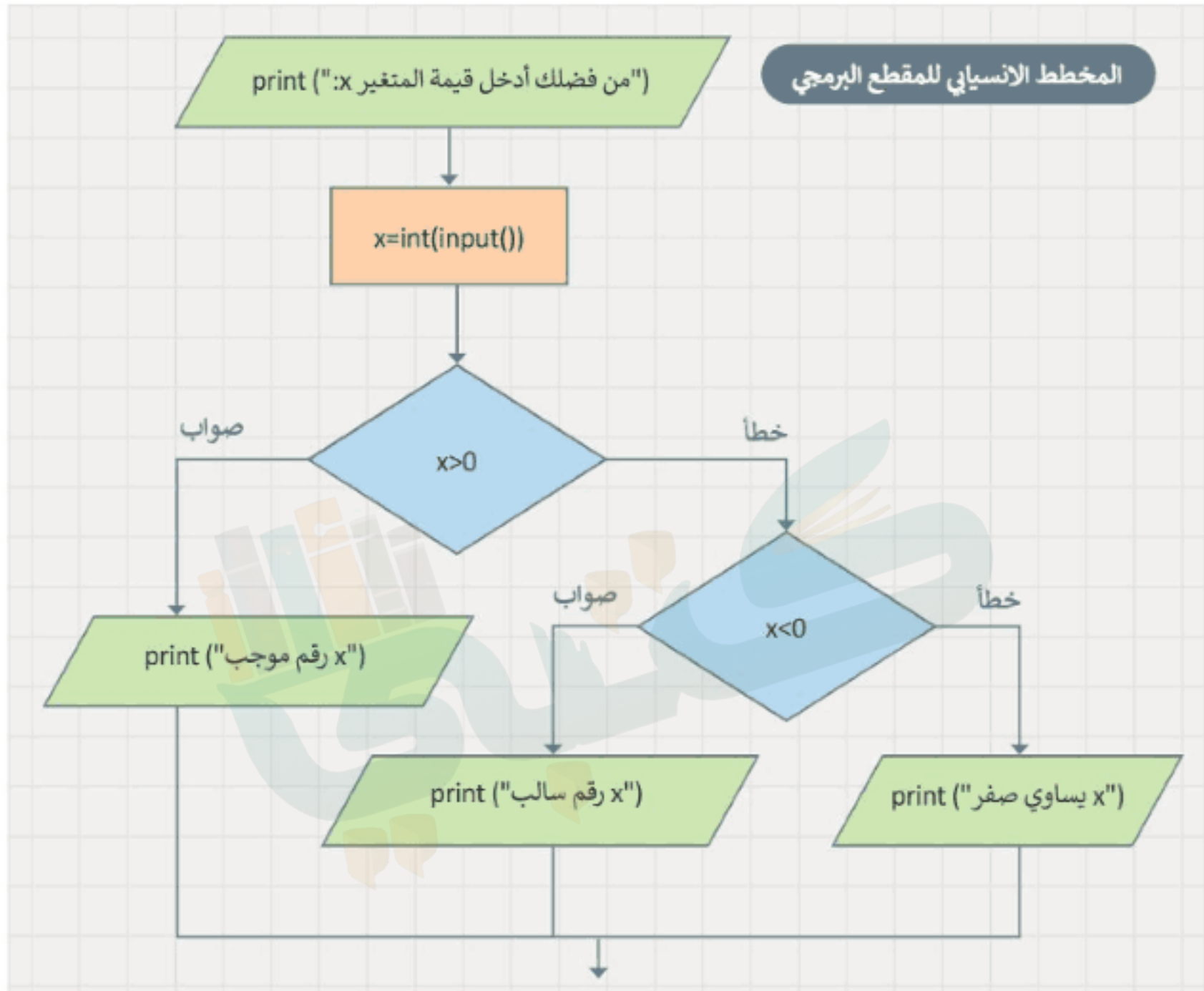
المخطط الانسيابي للمقطع البرمجي



حان الوقت لتشاهد بعض الأمثلة.

مثال 1: موجب أو سالب أو صفر

يستخدم المقطع البرمجي جملة `if ... elif`. ليقراً رقمًا ويتحقق مما إذا كان الرقم موجباً أم سالباً أم صفرًا ثم يطبع الرسالة المقابلة.



"من فضلك أدخل قيمة المتغير x:"

"x رقم موجب"

"x رقم سالب"

"x يساوي صفر"

من فضلك أدخل قيمة المتغير x:

-45

x رقم سالب



مثال 2: درجات الطلبة

يقرأ المقطع البرمجي درجة الطالب ثم يطبع الرسالة المقابلة.

"من فضلك أدخل الدرجة:"

"درجة غير صالحة"

"ممتاز"

"جيد جدًا"

"اجتهد أكثر"

من فضلك أدخل الدرجة:
12
درجة غير صالحة



جرب بنفسك

الدرجة

ما الذي يجب عليك إدخاله حتى
يتم طباعة "جيد جدًا"؟

ما الذي يجب عليك إدخاله حتى
يتم طباعة "اجتهد أكثر"؟

ما الذي يجب عليك إدخاله حتى
يتم طباعة "ممتاز"؟

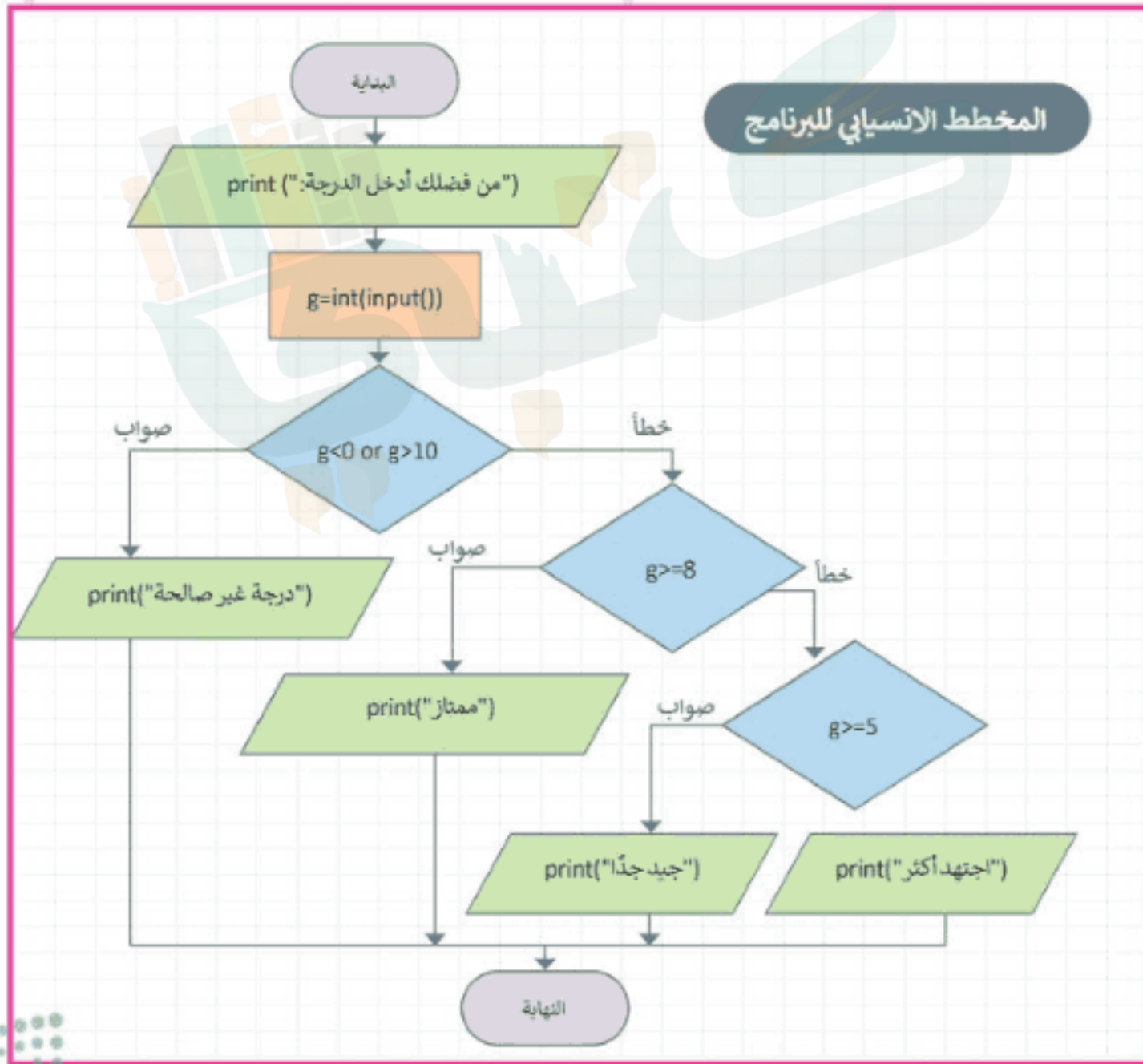


لنطبق معًا

تدريب 1

➤ ارسم المخطط الانسيابي للمقطع البرمجي.

```
print("من فضلك أدخل الدرجة:")
g=int(input())
if g<0 or g>10:
    print("درجة غير صالحة")
elif g>=8:
    print("ممتاز")
elif g>=5:
    print("جيد جدًا")
else:
    print("اجتهد أكثر")
```



تدريب 2

◀ وفقاً للمقطع البرمجي الذي أمامك:

ما نتيجة المتغير (num) إذا كانت:

a: num = 18

b: num = -7

```
num=int(input("أدخل رقم:"))
if num>=0:
    print(num)
else:
    num=num*(-1)
    print(num)
```

a

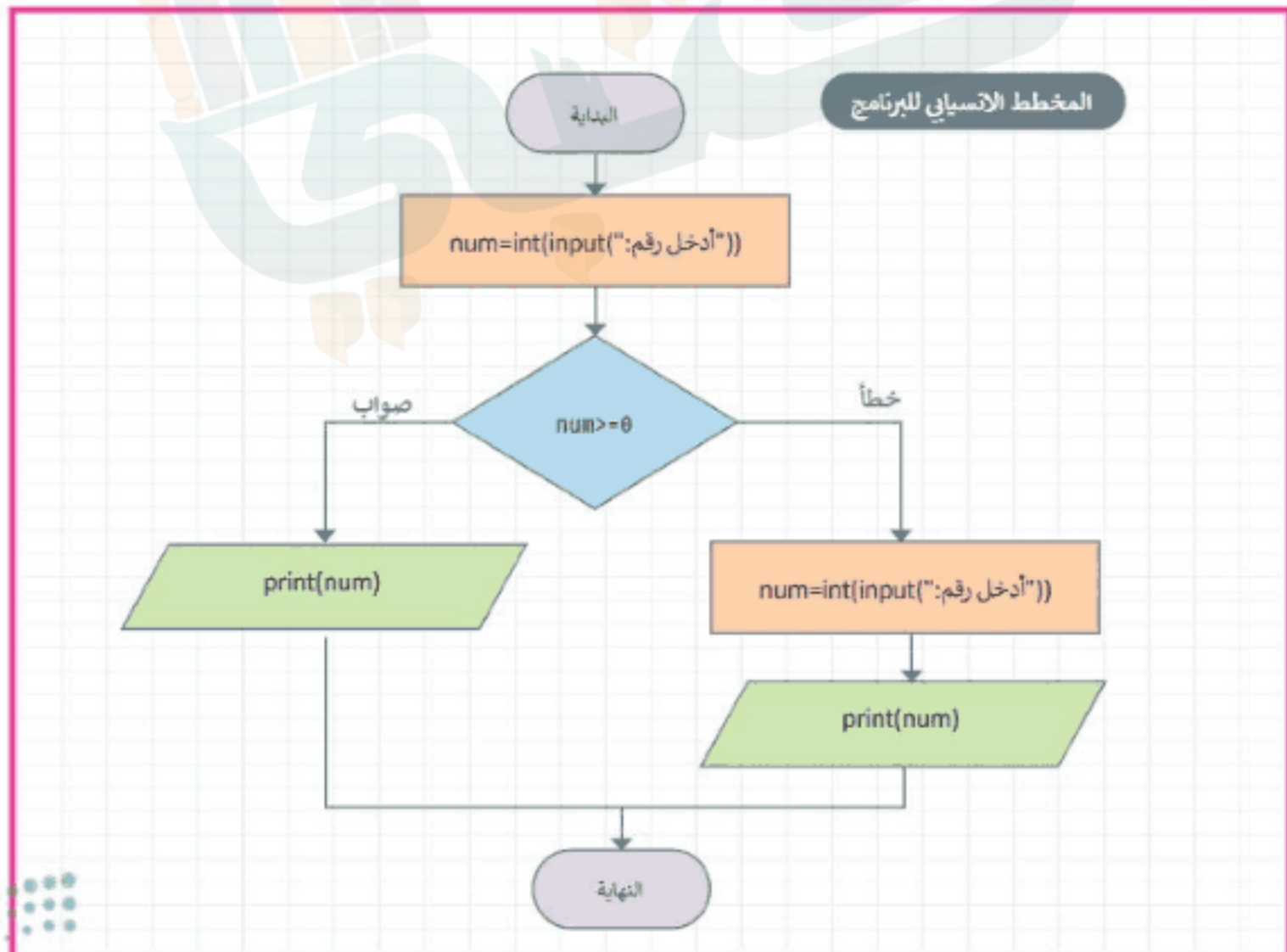
18

b

7

ما وظيفة المقطع البرمجي؟
يحسب المقطع البرمجي ويطبوع
القيمة المطلقة للرقم.

ارسم المخطط الانسيابي للمقطع البرمجي.



تدريب 3

➤ ارسم مخططًا انسيابيًا لإدخال درجة حرارة اليوم ويطبع إحدى الجمل التالية:

< "طقس معتدل"، إذا كانت درجة الحرارة محصورة بين 15 و 30.

< "طقس بارد"، إذا كانت درجة الحرارة أقل من 15.

< "طقس حار"، إذا كانت درجة الحرارة أكبر من 30.

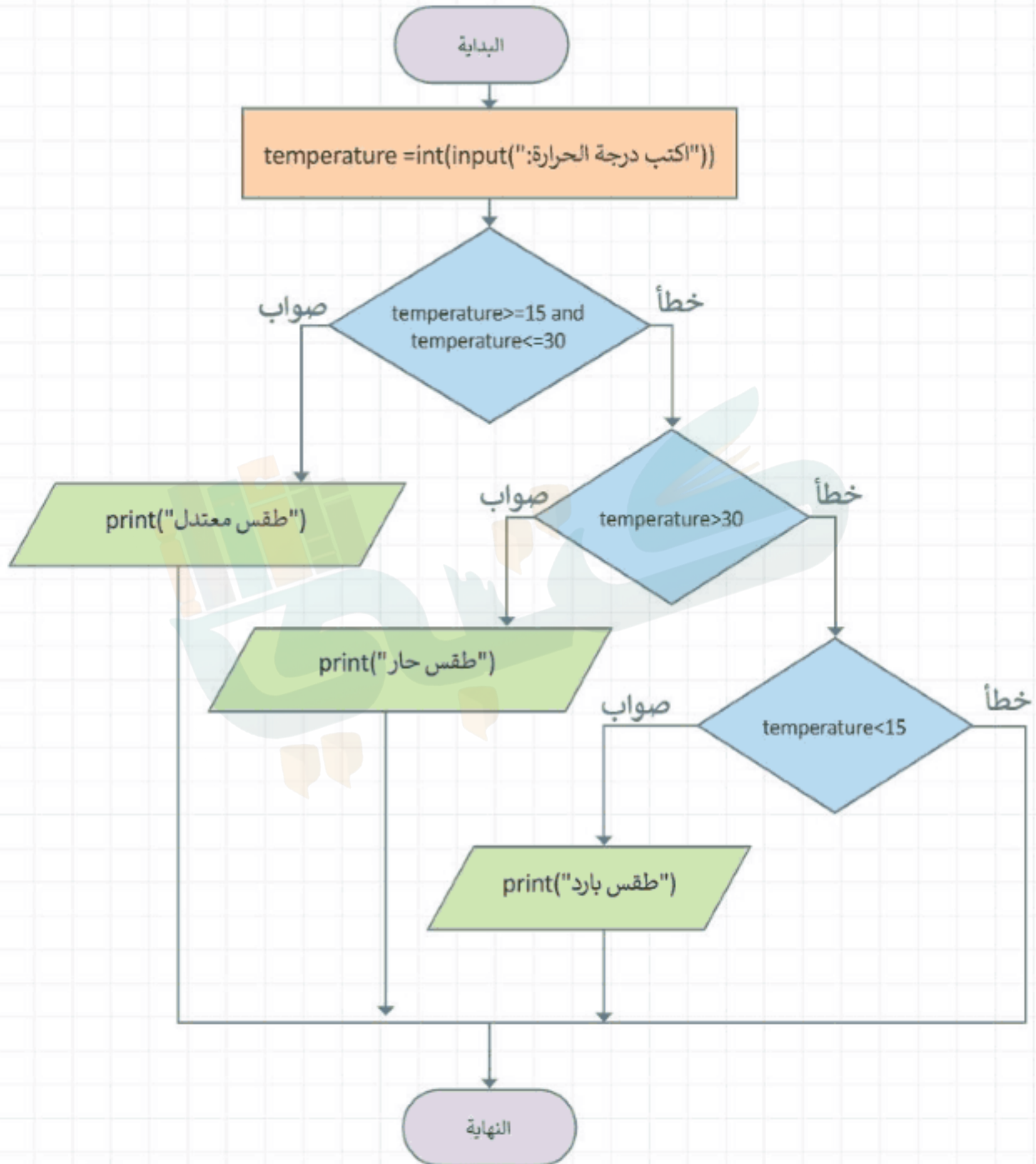
ثم اكتب المقطع البرمجي.

المقطع البرمجي

المخطط الانسيابي للمقطع البرمجي

```
temperature =int(input("اكتب درجة الحرارة:"))
if temperature>=15 and temperature<=30:
    print("طقس معتدل")
elif temperature>30:
    print("طقس حار")
elif temperature<15:
    print("طقس بارد")
```







الدرس الرابع: الشروط المتداخلة

التداخل

التداخل (Nesting) مصطلح يستخدم لوصف وضع كائن أو أكثر داخل كائن آخر. في برمجة جهاز الحاسب، العبارة المتداخلة عبارة موجودة داخل عبارة أخرى في المقطع البرمجي الأساسي للبرنامج. عند استخدام العبارات المتداخلة تُستخدم المسافة البادئة لتحديد الشرط الذي تنتمي إليه العبارة.

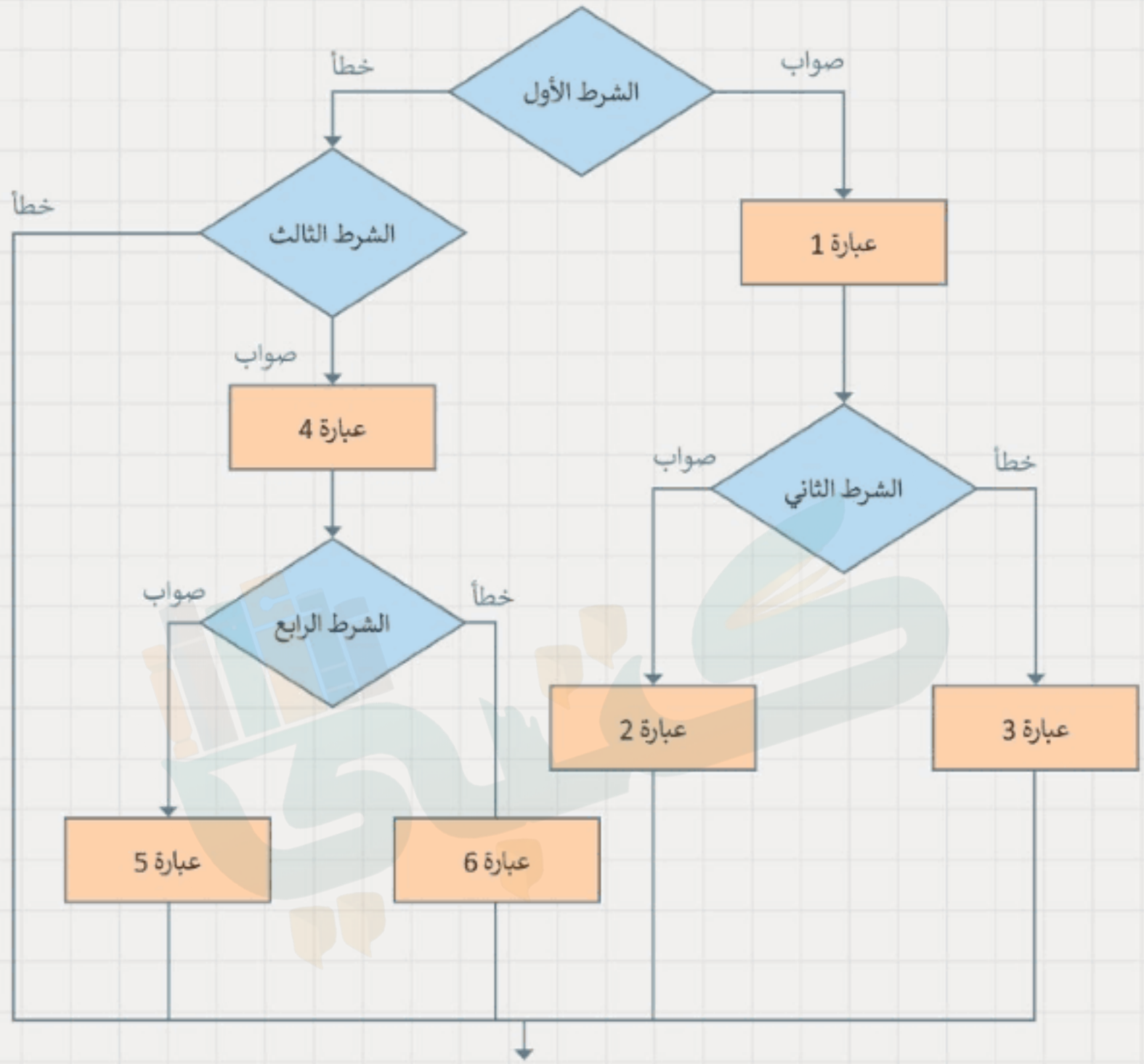
الجملة الشرطية if المتداخلة

الجملة الشرطية if المتداخلة عبارة عن جملة if البسيطة وتكون موجودة داخل (أو متداخلة مع) جملة if الأخرى أو جملة if...else الشرطية.

يمكن دمج أي عدد من العبارات في أي مجموعة داخل بعضها البعض.



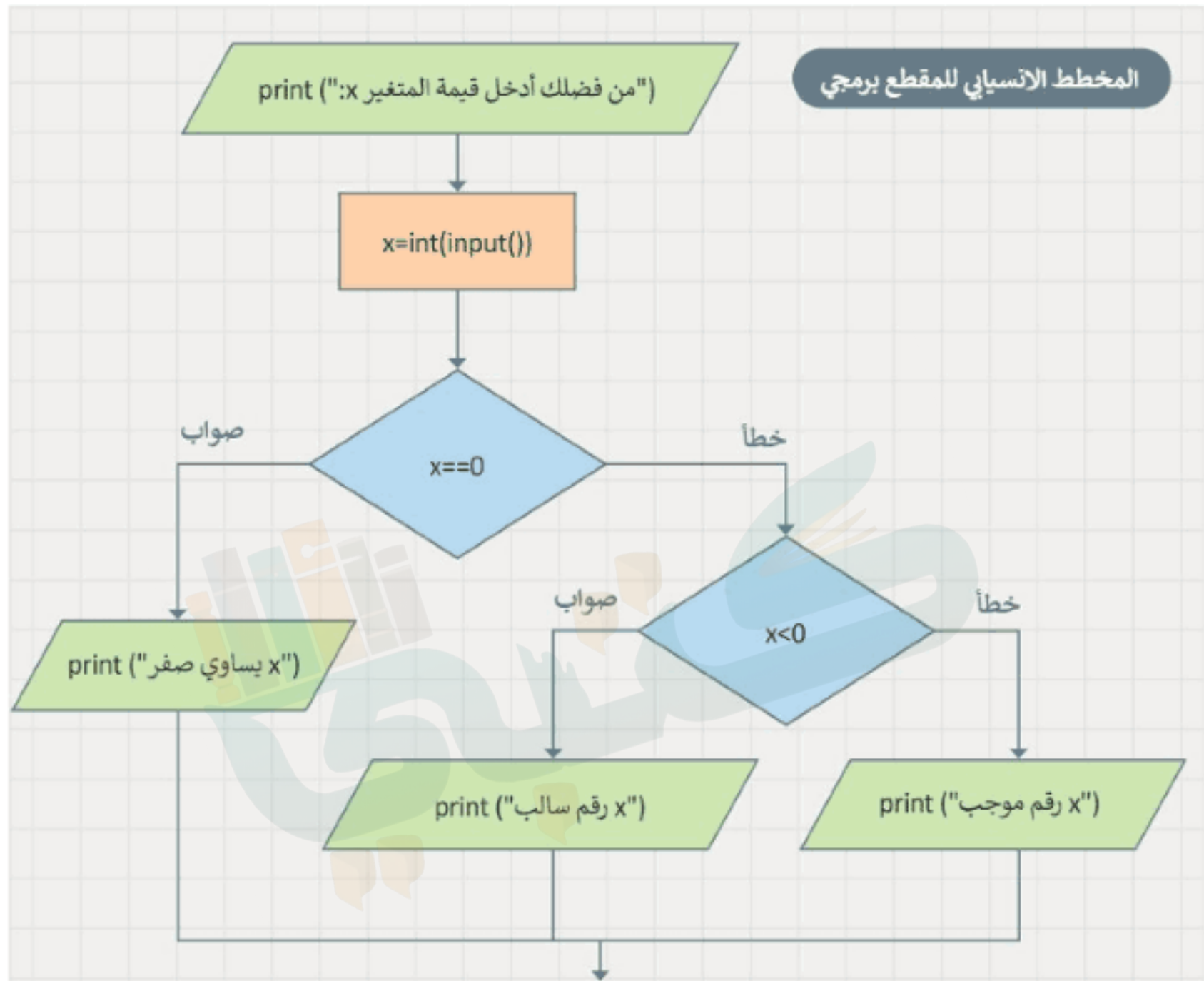
المخطط الانسيابي للمقطع البرمجي



شاهد بعض الأمثلة السابقة باستخدام الشروط المتداخلة.

مثال 1: جملة if المتداخلة

يستخدم المقطع البرمجي عبارة if المتداخلة لطباعة الرقم إذا كان الرقم المدخل موجباً أو سالباً أو صفراً.



"من فضلك أدخل قيمة المتغير x:"

"x يساوي صفر"

"x رقم سالب"

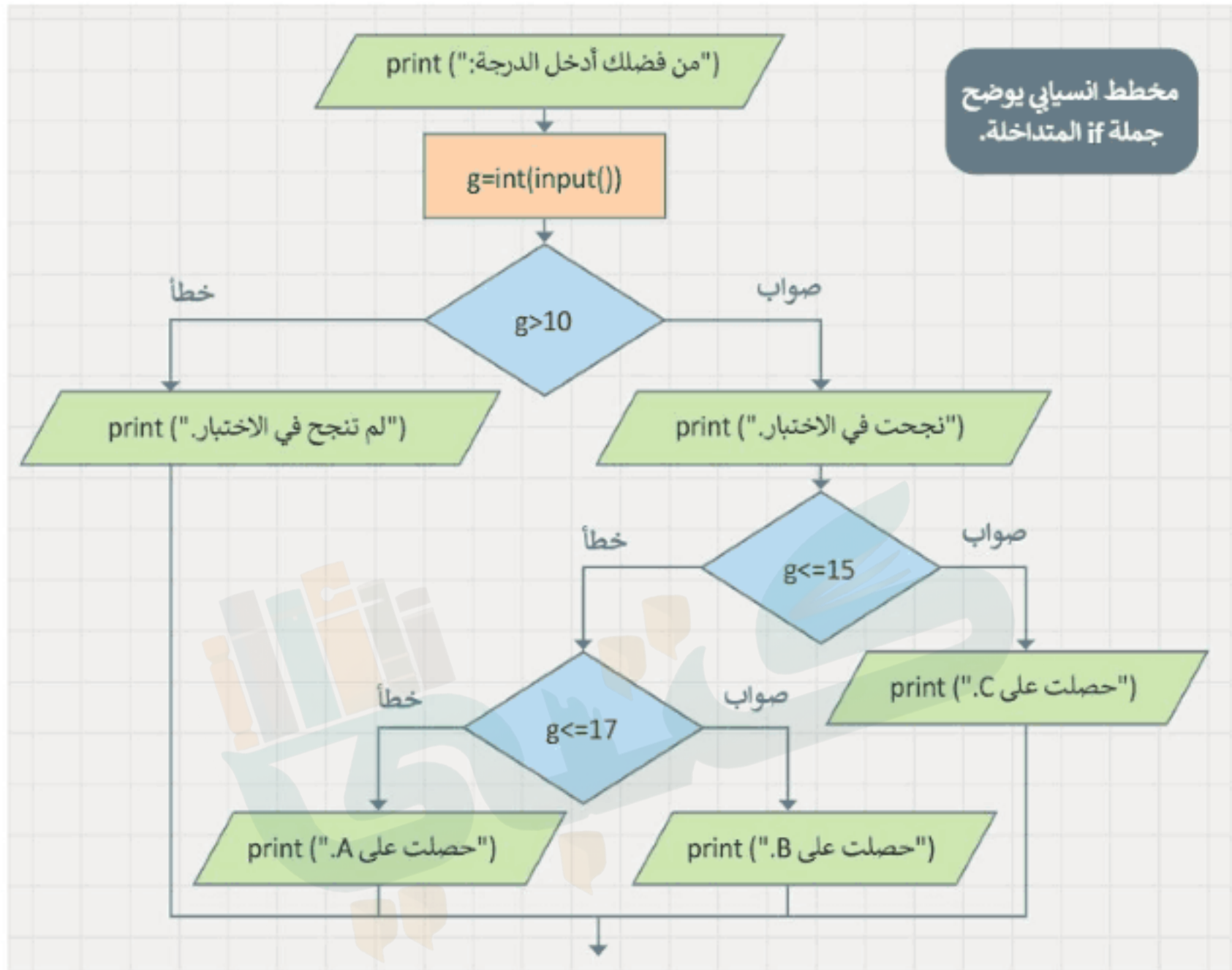
"x رقم موجب"

من فضلك أدخل قيمة المتغير x:

x رقم موجب

مثال 2: تقديرات الطلبة بالأحرف

يحسب المقطع البرمجي تقدير الطالب بالأحرف. لترى كيف يمكنك استخدام جملة **if** المتداخلة لإخبار الطالب إذا نجح في الاختبار والتقدير الذي حصل عليه بالأحرف.



"من فضلك أدخل الدرجة:"

"نجحت في الاختبار."

"حصلت على C."

"حصلت على B."

"حصلت على A."

"لم تنجح في الاختبار."

من فضلك أدخل الدرجة:

نجحت في الاختبار.

حصلت على B.



لنطبق معًا

تدريب 1

◀ ماذا سيعرض المقطع البرمجي على الشاشة إذا أدخلت قيمة الشهر (month) كالتالي:

```
month=int(input("اكتب رقم الشهر:"))
if month<1 or month>13:
    print("رقم الشهر خاطئ")
else:
    if month>=9 and month<=11:
        print("فصل الخريف")
    elif month==12 or month>=1 and month<=2:
        print("فصل الشتاء")
    elif month>=3 and month<=5 :
        print("فصل الربيع")
    else:
        print("فصل الصيف")
```

a 4

b 1

c 25

a

فصل الربيع

b

فصل الشتاء

c

رقم الشهر خاطئ



تدريب 2

يمكن للاعب أن ينضم لفريق كرة السلة إذا كان طوله أكبر من 1.80 متر، ووزنه بين 85 و125 كجم.

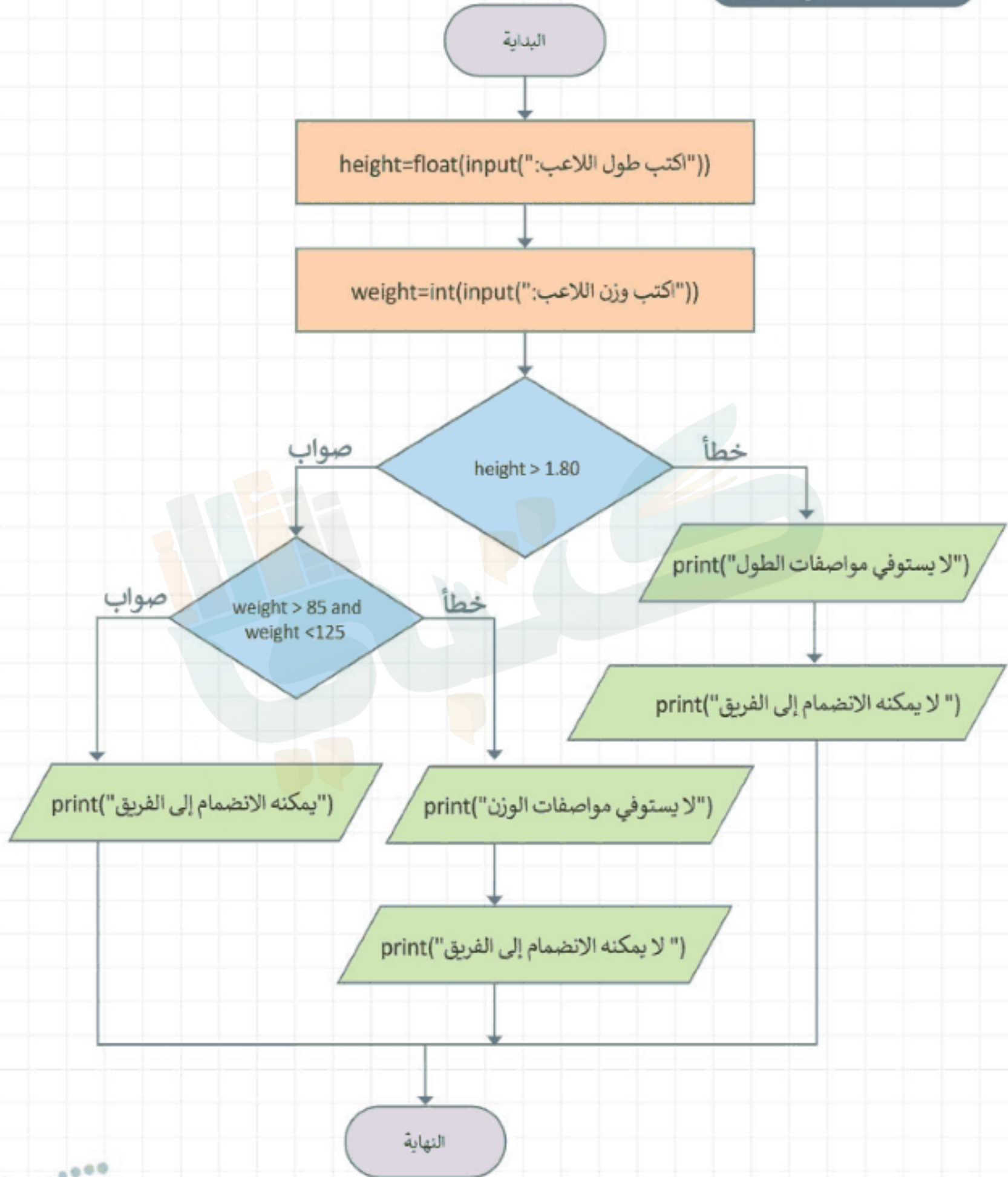
ارسم المخطط الانسيابي لمقطع برمجي يقرأ طول ووزن اللاعب الرياضي، ويعرض إذا كان بإمكانه الانضمام إلى فريق كرة السلة ثم اكتب المقطع البرمجي.

المقطع البرمجي

المخطط الانسيابي للمقطع البرمجي

```
height=float(input("اكتب طول اللاعب:"))
weight=int(input("اكتب وزن اللاعب:"))
if height > 1.80:
    if weight > 85 and weight <125:
        print("يمكنه الانضمام إلى الفريق")
    else:
        print("لا يستوفي مواصفات الوزن")
        print("لا يمكنه الانضمام إلى الفريق")
else:
    print("لا يستوفي مواصفات الطول")
    print("لا يمكنه الانضمام إلى الفريق")
```







ستحتاج في بعض الأحيان إلى تكرار مجموعة من الأوامر عدة مرات في النص البرمجي، وسيطلب هذا التكرار الكثير من الوقت والجهد. لذلك، توفر جميع لغات البرمجة تقريبًا بنية تحكم تسمى **حلقة (loop)**. ستسمح لك هذه الدالة بتنفيذ صف واحد أو مجموعة من المقاطع البرمجية عدة مرات. ويمكنك تحديد عدد التكرارات برقم معين، أو أن تعتمد على شرط معين. يدعم بايثون نوعين من الحلقات: حلقة **for** وحلقة **while**.

الحلقات في بايثون

حلقة **for**

حلقة **while**

حلقة **for**

تُستخدم حلقة **for** لتكرار مجموعة من الأوامر لعدد محدد من المرات، ويكون عدد التكرارات محددًا في قيم دالة النطاق (**range()**).

يجب أن تُسبق البيانات المتكررة بمسافة بادئة.

يمكنك استخدام دالة النطاق **for** لتحديد عدد مرات تنفيذ حلقة **for**.

المسافة البادئة في الحلقات

كما تعلمت سابقًا، فإن استخدام المسافة البادئة (Indentation) لعبارات IF الشرطية، هو أمر مهم جدًا في بايثون. يتعرف المقطع البرمجي على العبارات المضمنة في الحلقة والتي يتم تنفيذها في كل تكرار لها، من خلال المسافة البادئة، لذلك إذا لم تترك مسافة بادئة في مقطعك البرمجي، فستلقى رسالة خطأ.



دالة النطاق

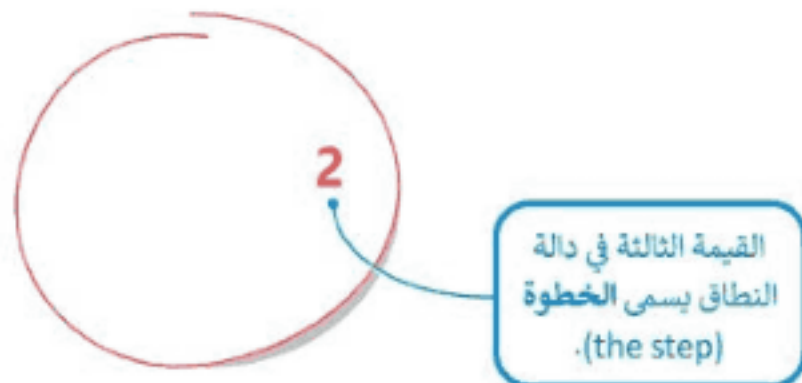
تُستخدم دالة النطاق (range()) مع الحلقة لتحديد عدد التكرارات، و يسمى المتغير الذي يحسب التكرار العدّاد (counter).
في دالة النطاق يبدأ العداد بالعد من 0، يزيد بمقدار 1 وينتهي العد قبل الوصول إلى الرقم المحدد. مثال على ذلك:



يمكن تحديد مقدار البداية بإضافة قيمة. على سبيل المثال، النطاق (2,5)، يعني أن العداد يبدأ من 2 وينتهي عند 4 (القيمة 5 غير مدرجة).



القيمة التلقائية للزيادة في دالة النطاق هي 1، ولكن يمكن تحديد مقدار الزيادة بإضافة قيمة ثالثة. على سبيل المثال، النطاق (1,5,2)، يعني أن العداد يبدأ في العد من 1، وينتهي عند 4 ويزداد بمقدار 2.



يمكن أن تكون الخطوة عددًا سالبًا، ويصبح العدُّ بشكلٍ عكسي.





جرب المقطع البرمجي التالي واكتب القيم التي تظهر على الشاشة.

مثال: درجات تقييم الطلبة

لقد عملت سابقًا على مثال لحساب درجات تقييم الطلبة، حيث يتحقق المقطع البرمجي من اجتياز الطالب للامتحانات. ستتعلم كيف يمكنك تطبيق حلقة **for**. لقد تحققت في المثال السابق من درجات طالب واحد فقط، لتفترض أن عليك التحقق من درجات صف كامل مكون من 15 طالبًا.

"الرجاء إدخال اسم الطالب:"

"الرجاء إدخال درجة الطالب:"

"اجتاز الامتحانات."

"حصل على C."

"حصل على B."

"حصل على A."

"لم يجتز الامتحانات."

لا تنس ترك
مسافة.

لقراءة اسم
كل طالب.



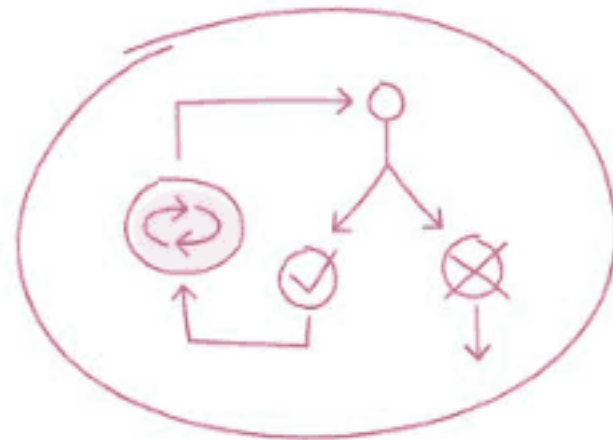
حلقة while الشرطية

تُستخدم حلقة **while** عندما لا يكون عدد التكرارات معروفاً مسبقاً. طالما أن الحالة صحيحة، فإن الحلقة تتكرر وتُفحص بعد كل تكرار للتأكد من صحتها. عندما تُصبح الحالة خطأً، يتوقف التكرار وينتقل المقطع البرمجي إلى الصف التالي بعد الحلقة. أما إذا كانت حالة الشرط خطأً في البداية، فلن يتم تنفيذ الحلقة على الإطلاق.



يجب أن تُسبق الأوامر المتكررة بمسافة بادئة.

انظر إلى مثال حلقة **while** الشرطية، يُدخل المستخدم قيمة للمتغير x ، وتنتهي الحلقة عندما يُدخل المستخدم 0 كقيمة للمتغير x .



يُمكنك استخدام حلقة **while** الشرطية للتحقق من مدخلات المستخدم في متغير معين.

مثال: درجات تقييم الطلبة

في هذا المثال، سيطلب المقطع البرمجي من المستخدم إدخال درجة الطالب، ويجب أن تكون هذه الدرجة أكبر من أو تساوي 0، وأقل من أو تساوي 20. في حال أدخل المستخدم قيمة خارج هذا النطاق، فسيعرض المقطع البرمجي رسالة خطأ ويطلب من المستخدم إدخال درجة صالحة.

يجب أن تكون درجات الطلبة أكبر من أو تساوي 0
وأقل من أو تساوي 20
"أدخل درجة الطالب:"

"درجة غير صالحة، أدخل درجة من ٠-٢٠."
"أدخل درجة صالحة:"
"درجتك هي:"

أدخل درجة الطالب: 67
درجة غير صالحة، أدخل درجة من 0-20.
أدخل درجة صالحة: 18
درجتك هي: 18



جرب المقطع البرمجي التالي واكتب ما يظهر على الشاشة.

"مرحبًا"



حلقة لا نهائية

إذا لم يُصبح شرط حلقة **while** خطأ، فسينتهي بك الأمر بحلقة لا نهائية (Infinite loop)، وهي حلقة لا تنتهي أبداً. عندما تستخدم حلقة **while** الشرطية، يجب أن تتضمن أمراً، أو مجموعة من الأوامر التي ستغير الحالة من صواب (True) إلى خطأ (False).



جرب المقطع البرمجي التالي. ماذا تلاحظ؟

في المثال السابق، قيمة المتغير لا تتغير، لذلك سيتم تكرار المقطع البرمجي إلى الأبد.

لإيقاف تكرار الحلقة، اضغط على **Ctrl + C** في نافذة **Python shell**.



عبارة الإيقاف

في بعض الأحيان تريد إنهاء حلقة قبل أن تصبح حالتها خطأ. في مثل هذه الحالات، ستستخدم عبارة الإيقاف (Break statement). تُنهي عبارة الإيقاف الحلقة التي تحتوي عليها، وينتقل المقطع البرمجي إلى الصف المتواجد بعد الحلقة. يمكن استخدام عبارة الإيقاف في حلقة for أيضًا.

عادة ما يكون هناك العديد من الطرق المختلفة لأداء نفس المهمة.

نُفضل إحدى الطرق على الأخرى بناءً على عدة عوامل، أهمها سرعة تشغيل المقطع البرمجي ومساحة التخزين المطلوبة.

المبرمج يُحدد أفضل طريقة.

"اكتب كلمة: "
"إيقاف"
"لقد استخدمت عبارة الإيقاف."
"اكتب كلمة مختلفة"

اكتب كلمة: سيارة
اكتب كلمة مختلفة
اكتب كلمة: طائرة
اكتب كلمة مختلفة
اكتب كلمة: دراجة
اكتب كلمة مختلفة
اكتب كلمة: دراجة هوائية
اكتب كلمة مختلفة
اكتب كلمة: إيقاف
لقد استخدمت عبارة الإيقاف.

إن السلاسل النصية في بايثون حساسة لحالة الأحرف، لذلك عند التحقق من قيمة متغير، عليك أيضًا التحقق مما إذا كانت هناك مسافات زائدة؛ لأن المسافة تُعتبر حرفًا في بايثون أيضًا.



لنطبق معًا

تدريب 1

◀ كم مرة سينفذ أمر الطباعة (print())؟ اختر الإجابة الصحيحة:

اختر الإجابة الصحيحة		
☐	لن ينفذ؛ لأن تنسيق الأوامر غير صحيح.	<pre>for i in range (0,5,3): print(i)</pre>
☒	ينفذ مرتان.	
☐	ينفذ 3 مرات.	
☒	ينفذ 5 مرات.	<pre>for i in range (10,1,-2): print(i)</pre>
☐	عدد المرات غير معروف.	
☐	ينفذ 4 مرات.	
☐	ينفذ 5 مرات.	<pre>i=5 while i>1: print(i) i=i-1</pre>
☐	عدد المرات غير معروف.	
☒	ينفذ 4 مرات.	



تدريب 2

اكتب مقطعًا برمجيًا يعرض الأرقام من 10- إلى 1- باستخدام حلقة for.

```
for i in range (-10,0) :  
    print (i)
```

تدريب 3

اكتب مقطعًا برمجيًا يعرض الأرقام 100، 95، 90،...، 0 على الشاشة.

```
for i in range (100,-5,-5) :  
    print (i)
```

تدريب 4

اكتب مقطعًا برمجيًا باستخدام حلقة while لعرض أول 13 عدد من مضاعفات العدد 7.

```
n=1  
while (n<14):  
    print (n, "*7", "=", 7*n)  
    n=n+1
```

تدريب 5

اكتب مقطعًا برمجيًا لحساب مجموع جميع الأرقام من 1 إلى رقم معطى.

< اطلب من المستخدم إدخال رقم.

< احسب مجموع جميع الأرقام من 1 إلى الرقم المعطى.

< على سبيل المثال، إذا أدخل المستخدم 8، فيجب أن يكون الناتج $36 (8 + 7 + 6 + 5 + 4 + 3 + 2 + 1)$.

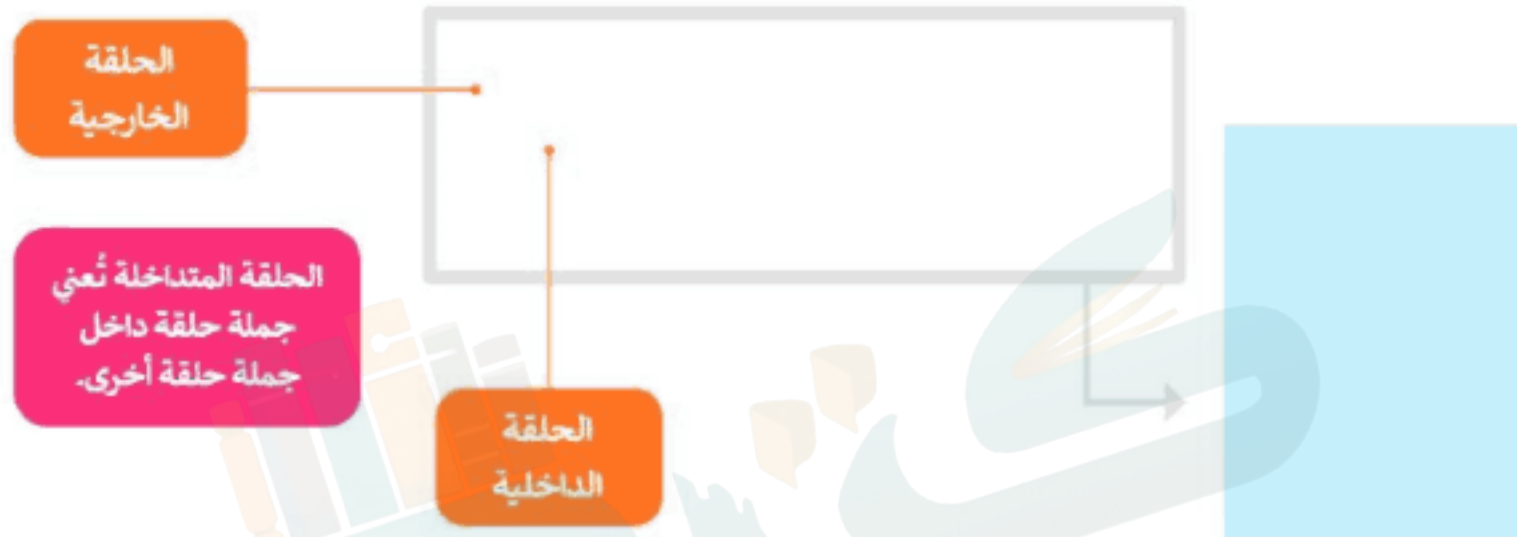
```
n = int(input("Enter a number: "))  
sum = 0  
for i in range (n+1):  
    sum = sum + i  
  
print(sum)
```





الحلقات المتداخلة

لقد تعرّفنا على الحلقات في الدرس السابق وتعلّمت كيفية استخدام أنواعها المختلفة. في هذا الدرس ستتعلم كيفية وضع حلقة داخل حلقة أخرى، ويسمى هذا الإجراء بالتداخل (Nesting) كما تسمى هذه الحلقات بالحلقات المتداخلة (Nested loops)، حيث يمكنك إدخال أي نوع من الحلقات في أي نوع آخر، وعلى سبيل المثال: يمكنك إدخال حلقة for في حلقة while أو العكس. وتعدّ حلقات for هي الحلقات المتداخلة الأكثر شيوعاً.



القيم	
i	j
0	0
	1
1	0
	1
2	0
	1

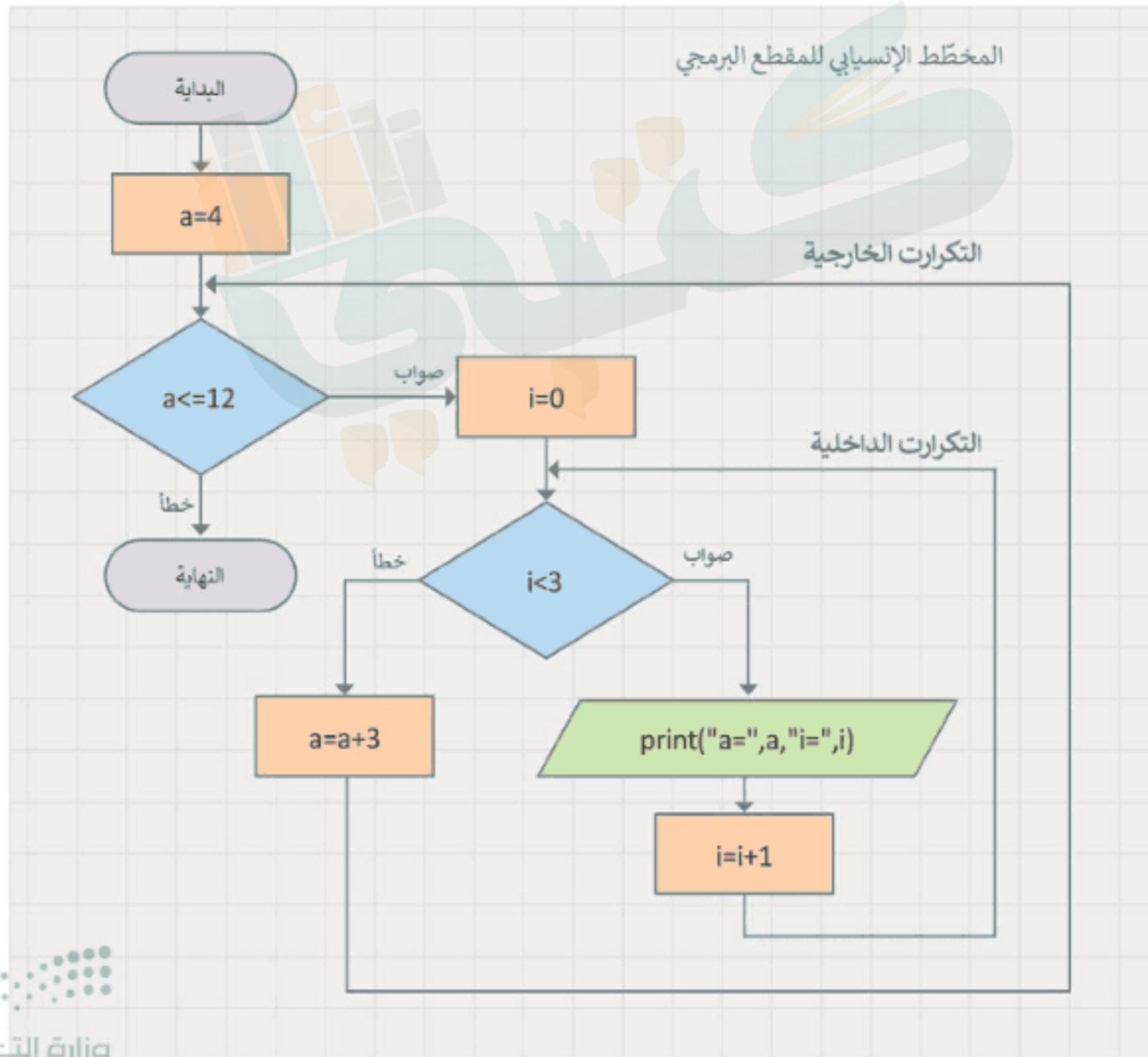
خطوات التنفيذ:

1. i يحصل على القيمة 0، ستتكرر الحلقة الداخلية مرتين: $j = 0$ ، $j = 1$. for
 2. الآن ارفع قيمة i، ثم $i = 1$ ، ستتكرر الحلقة الداخلية مرتين: $j = 0$ ، $j = 1$. for
 3. الآن ارفع قيمة i، ثم $i = 2$ ، ستتكرر الحلقة الداخلية مرتين: $j = 0$ ، $j = 1$. for
- في النهاية، تكرر الحلقة الخارجية 3 مرات والحلقة الداخلية 6 مرات.

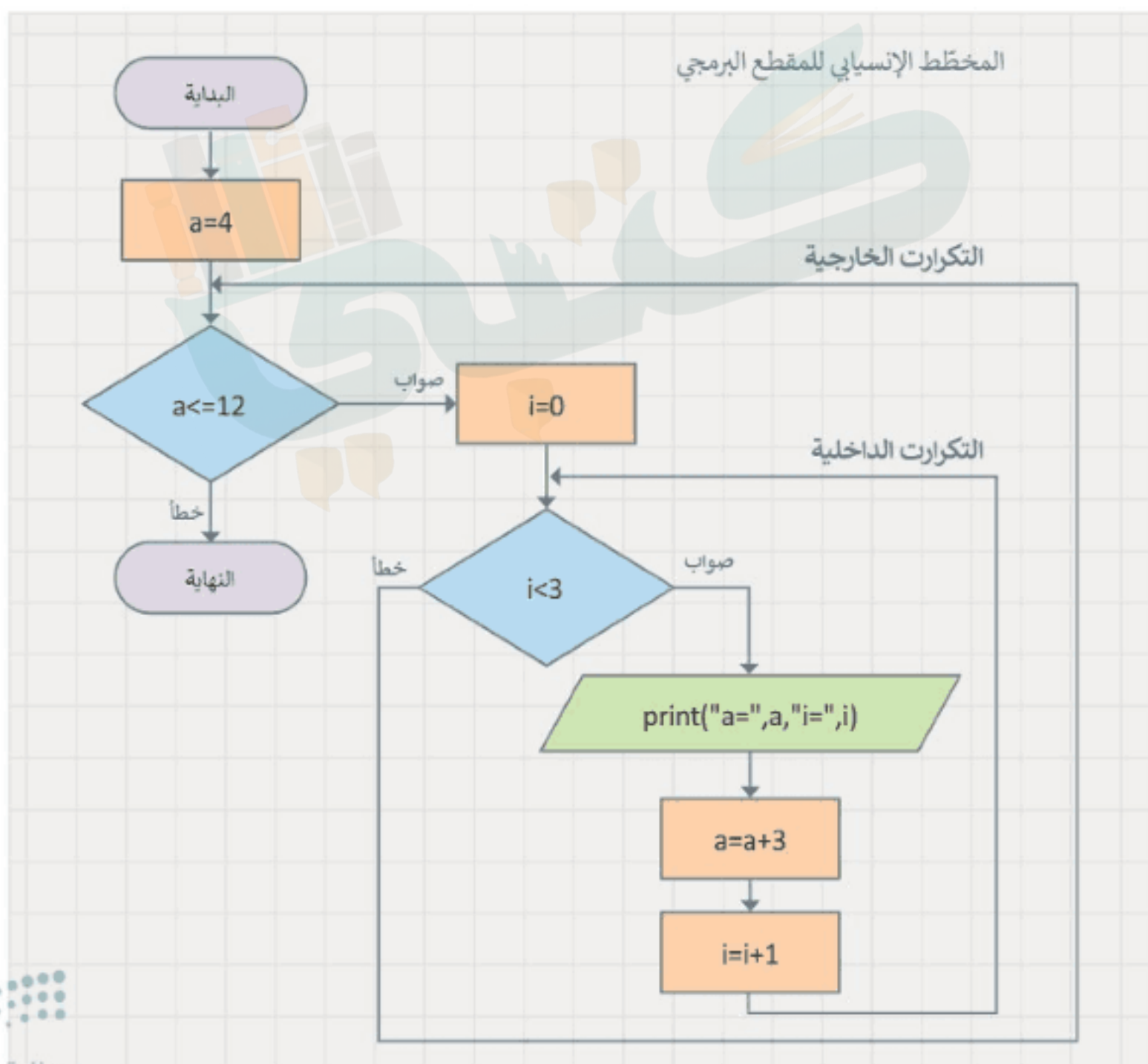
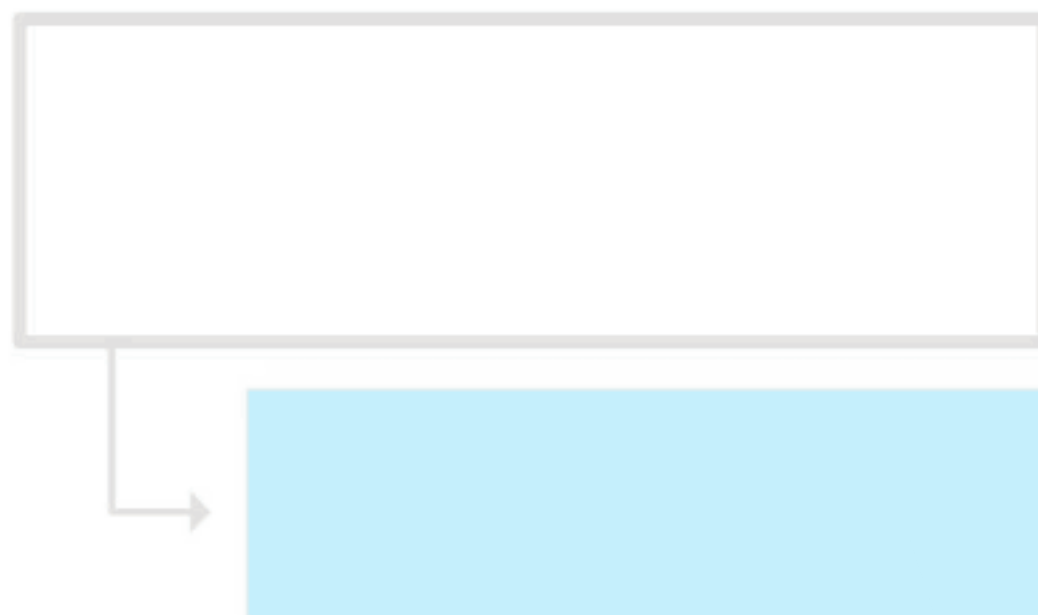




وضع المسافة البادئة في الحلقات المتداخلة مهم جداً، حيث تُحدد المسافة البادئة الأوامر التي يتم تضمينها في كل حلقة.



وضع المسافة البادئة مهم للغاية ويغير المقطع البرمجي بأكمله في بايثون. يمكنك أن ترى أنه إذا غيرت المسافة البادئة في المثال السابق، فإن النتيجة ستكون مختلفة.



القواعد التي تنطبق على الحلقات المتداخلة:

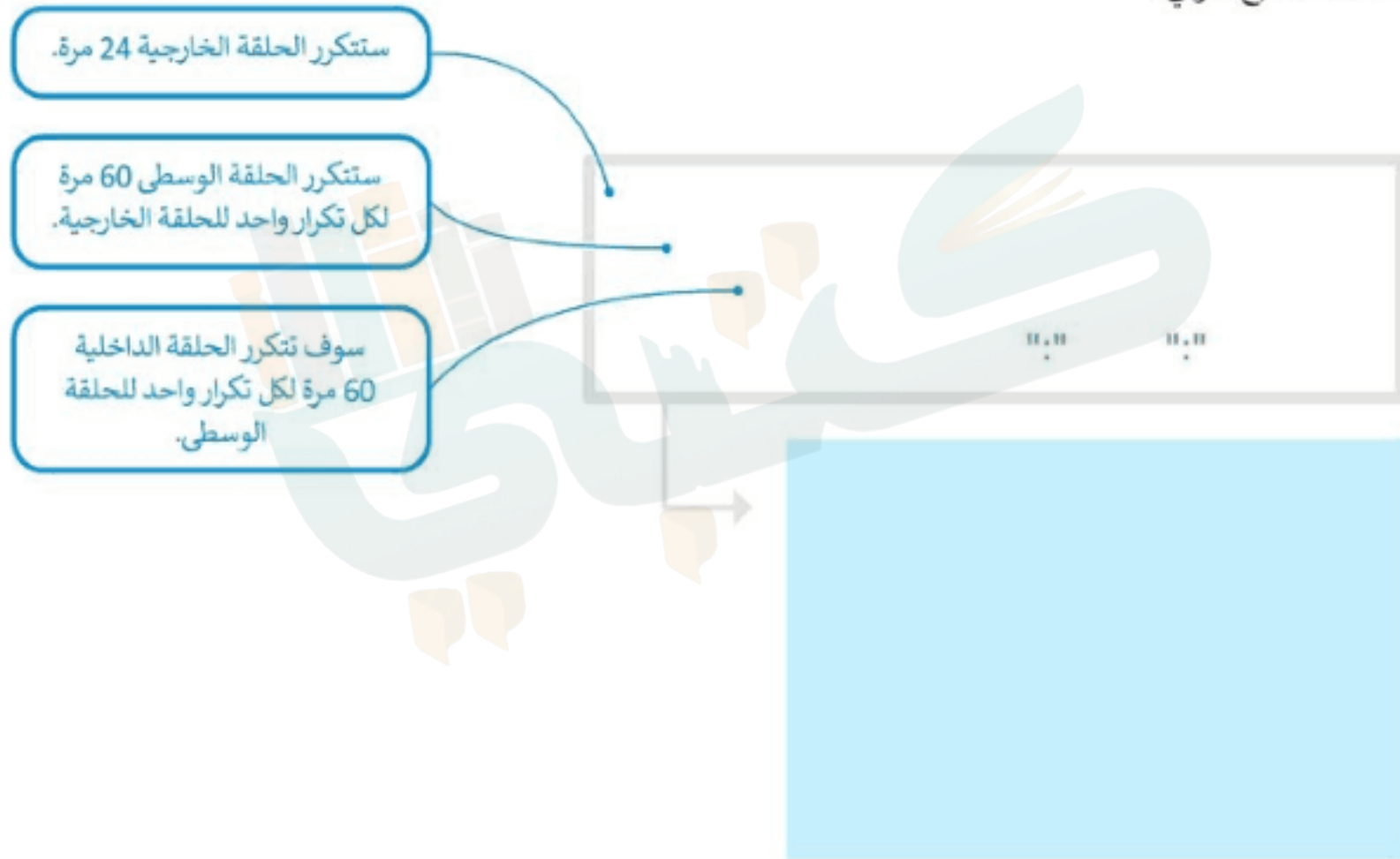
- < يجب أن تكون كل حلقة داخلية مدمجة بالكامل داخل الحلقة الخارجية، ولا يمكن أن تتشابك الحلقات.
- < لا يمكن استخدام نفس المتغير كعداد لحلقتين متداخلتين أو أكثر.
- < الحلقة الداخلية يجب أن تكتمل أولاً.
- < الحلقة الداخلية تنفذ جميع تكراراتها لكل تكرار من تكرارات الحلقة الخارجية.

مثال توضيحي

الساعة الرقمية هي إحدى أمثلة الحلقات المتداخلة في العالم الحقيقي.

في الساعة الرقمية، تحتاج إلى 3 حلقات:

- < الحلقة الأولى ستتبع الساعات.
- < الحلقة الثانية ستتبع الدقائق.
- < الحلقة الثالثة ستتبع الثواني.



مثال: حساب درجات تقييم الطلبة

المعلم يريد حساب درجات تقييم طلبته.

يضم فصله 30 طالبًا، وخضع كل طالب لاختبارين وامتحان نهائي واحد.
الدرجة النهائية لكل طالب هي متوسط تلك الدرجات الثلاث.

الحلقة الخارجية للطلبة ال 30

"اكتب اسم الطالب:"

ابدأ بكتابة الدرجات الإجمالية لكل طالب

حلقة داخلية للدرجات الثلاث

"للتالي:" "اكتب الدرجة"

مجموع الدرجات الثلاث للتالي

احسب الدرجة النهائية

"الدرجة النهائية لـ" "هي"

اكتب اسم الطالب: محمد

اكتب الدرجة 1 للتالي:

75

اكتب الدرجة 2 للتالي:

80

اكتب الدرجة 3 للتالي:

90

الدرجة النهائية ل محمد هي 81.66666666666667

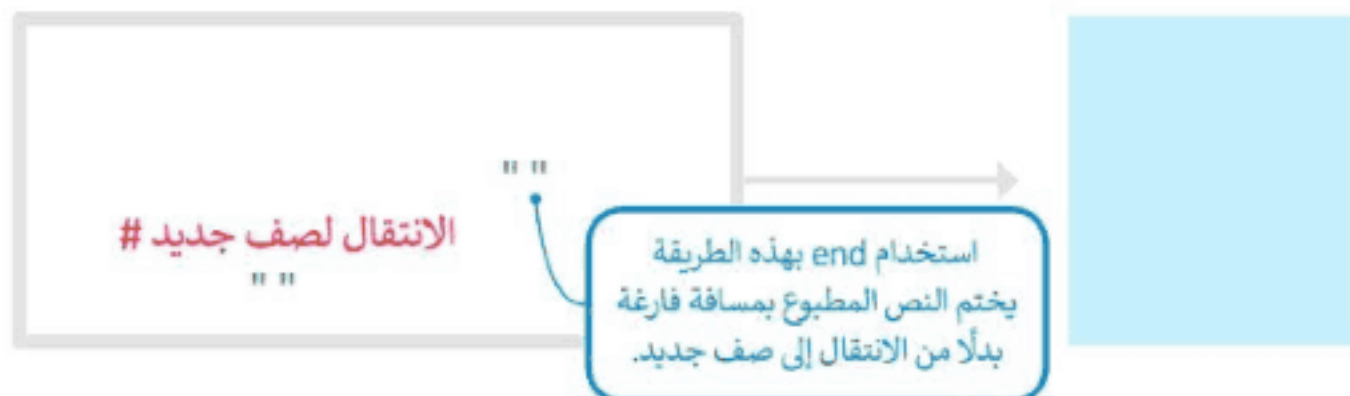
اكتب اسم الطالب:



أنماط الطباعة

يمكنك استخدام بايثون لعرض الأنماط على الشاشة. لطباعة أي نمط، هناك هيكل عام عليك اتباعه، حيث تحتاج إلى تحديد عدد الصفوف والأعمدة في النمط. تُخبرك الحلقة الخارجية بعدد الصفوف المستخدمة، تخبرك الحلقة الداخلية بعدد الأعمدة المستخدمة لطباعة النمط.

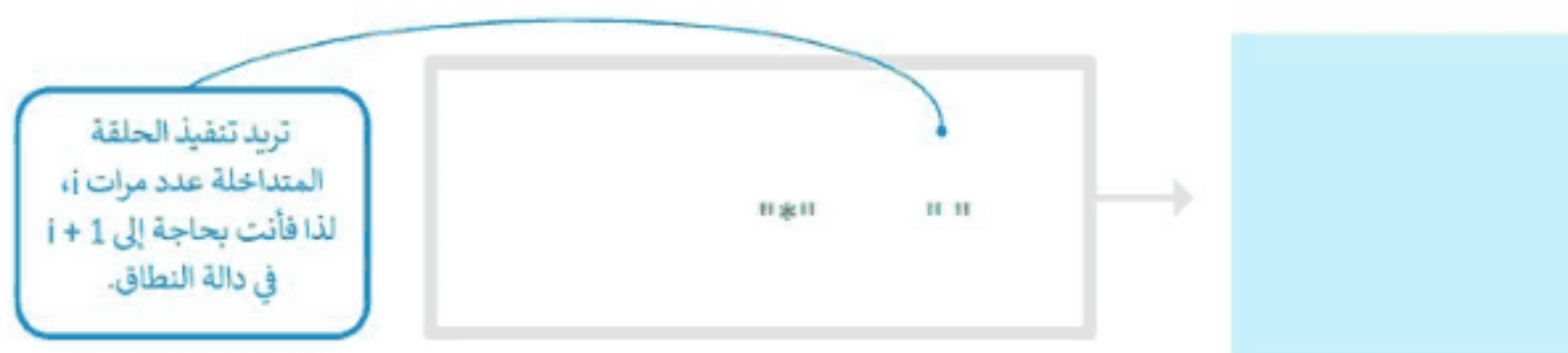
بعض الأمثلة على الأنماط الرقمية:



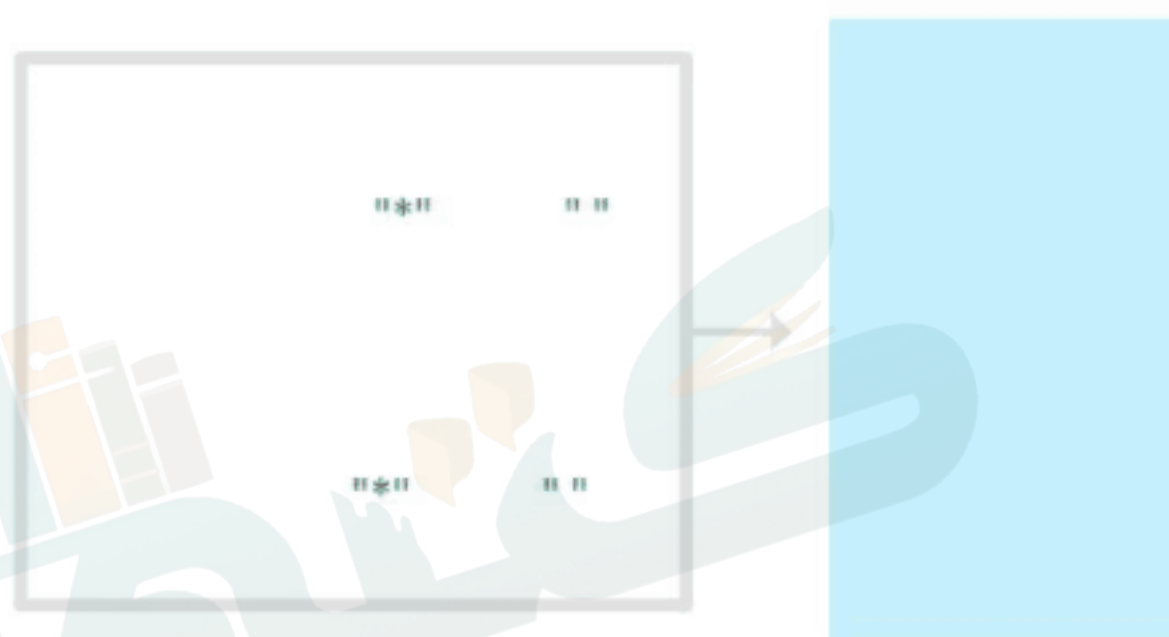
في المثال أعلاه، الحلقات الخارجية والداخلية من نوع حلقة **for**، حيث تعمل الحلقة الخارجية في النطاق (1,6) وتعمل الحلقة الداخلية داخل النطاق المحدد للحلقة الخارجية. سيتم عرض الرقم الأول مرة واحدة في الصف الأول، والرقم الثاني مرتين في الصف الثاني، والرقم الثالث ثلاث مرات في الصف الثالث وهكذا.



في المثال التالي، ستري أنه يمكنك طباعة أنماط هرمية من النجوم. ستطبع نجمة واحدة في الصف الأول، ونجمتين في الصف الثاني وتستمر بالطباعة حتى الصف الخامس.



يمكنك أيضًا رسم مثلث باستخدام النجوم، حيث ستستخدم 4 حلقات **for**، حلقتين داخليتين مخصصتين لتكرار العمود، وحلقتين خارجيتين لتكرار الصف.



جرب المقطع البرمجي التالي، واكتب النتيجة التي تظهر على الشاشة.



لنطبق معًا

تدريب 1

⬅ شغل المقطع البرمجي التالي وأكمل الجدول.

```
x=2
c=5
while c>0:
    for i in range (7,12,2):
        x=x+3
        print(i,x)
    c=c-3
    print(x,c)
```

جدول القيم			
الشاشة	i	c	x
		5	2
7 5	7	5	5
8 9	9	5	8
11 11	11	5	11
2 11	15	2	
7 14	7	2	14
17 9	9	2	17
20 11	11	2	20
-1 20	13	-1	

تدريب 2

⬅ شغل المقطع البرمجي التالي وتحقق من عدد الصف المطبوعة.

```
for i in range(4):
    for j in range(8):
        print(i, j)
```

A. 4



B. 5



C. 12



D. 24



E. 32



تدريب 3

اكتب مقطعًا برمجيًا لعرض جدول الضرب لجميع الأرقام من 1 إلى 10.

```
for i in range(11):  
    for j in range (11):  
        print(i,"*",j,"=",i*j)
```

تدريب 4

ما وظيفة المقطع البرمجي التالي؟ وما مخرجاته؟

```
for i in range(3):  
    for j in range(3):  
        if j == i:  
            break  
        print(i, j)
```

النتائج:

```
1 0  
2 0  
2 1
```

وظيفة المقطع البرمجي: طباعة الأعداد:

1 0

2 0

2 1

تلميح: تقع عبارة الإيقاف داخل حلقة متداخلة. لذا؛ فإذا كانت $j=i$ فإن عبارة الإيقاف ستوقف الحلقة الداخلية.



تدريب 5

اكتب المقطع البرمجي الذي يطبع النمط التالي:

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

المقطع البرمجي:

```
rows = 5
for i in range(0, rows + 1):
    for j in range(rows - i, 0, -1):
        print(j, end=' ')
    print()
```

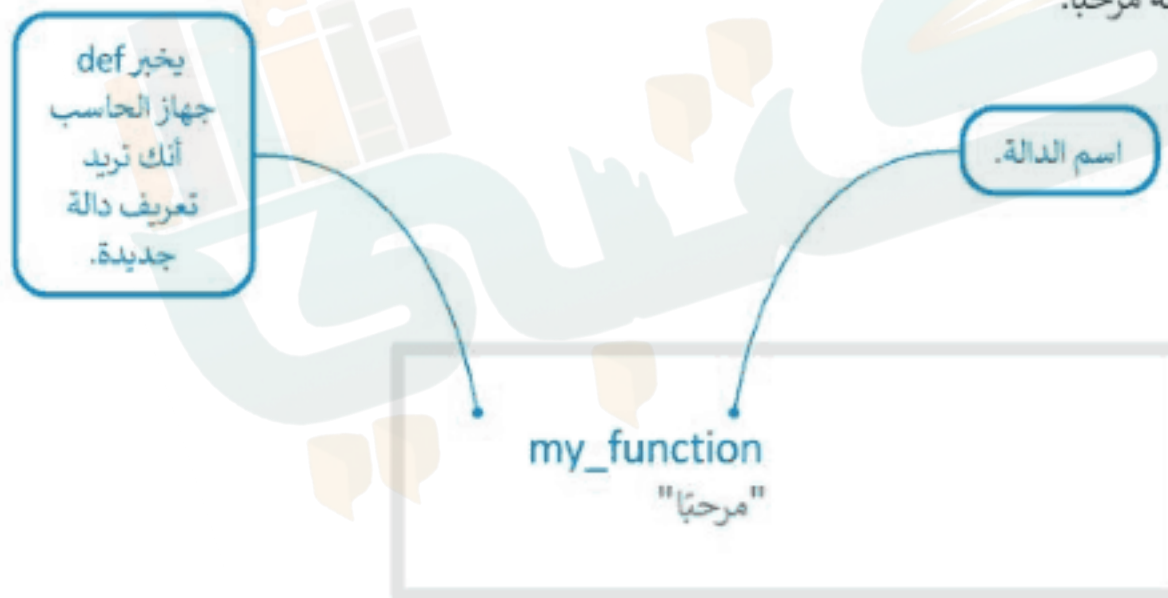




الدالة (Function) مجموعة من الأوامر التي يتم تجميعها في مكان واحد مع إعطائها اسمًا (تعريفًا)، ويتم تنفيذها من خلال استدعائها عند الحاجة لها. ويحتوي البايثون على مجموعة من الدوال الجاهزة والتي سبق لك استخدامها، مثل: دالة الطباعة (print())، ودالة الطول (len())، ودالة النطاق (range())، ودالة الإدخال (input()). في هذا الدرس ستتعلم كيفية إنشاء دوال جديدة لتساعدك في بعض المهام والمشكلات المتكررة.

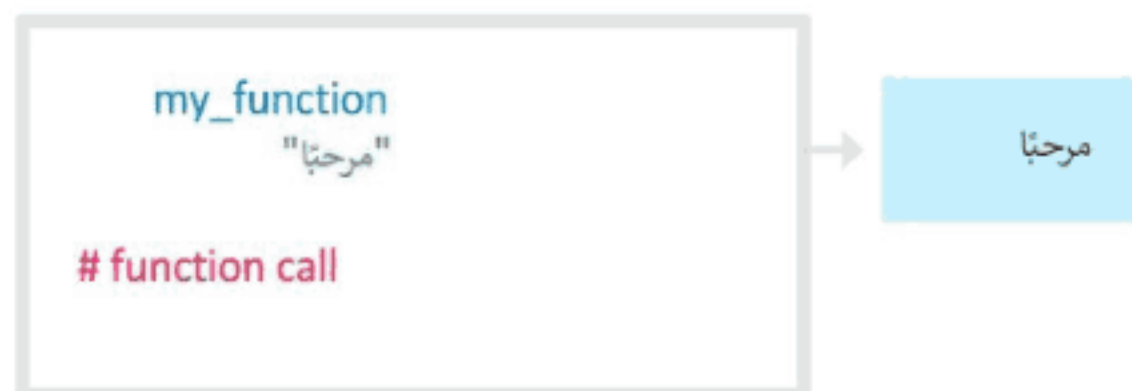
إنشاء الدوال الخاصة بك

في بايثون، يُعرّف الجزء الأول من الدالة باستخدام الكلمة الأساسية **def** ويحتوي هذا الجزء على تعريف الدالة (function definition)، ثم يأتي الجزء الثاني وهو اسم الدالة (name). في نهاية رأس الدالة (function header)، هناك نقطتان (colon) للإشارة إلى أن ما يلي عبارة عن مجموعة من المقاطع البرمجية، ويجب وضع مسافة بادئة لها. بعد ذلك، يتبع محتوى الدالة (function body)، والذي في هذا المثال، يطبع الرسالة فقط. الآن ستحدد دالة تطبع رسالة مرحبًا.



استدعاء دالة

لاستدعاء دالة، استخدم اسم الدالة متبوعًا بأقواس. ومن الأمثلة على الدالة ما يلي:



الميزة الرئيسية للدوال هي أنه يمكنك استدعاؤها مرارًا وتكرارًا من المقطع البرمجي الرئيس.



المعاملات والوسائط

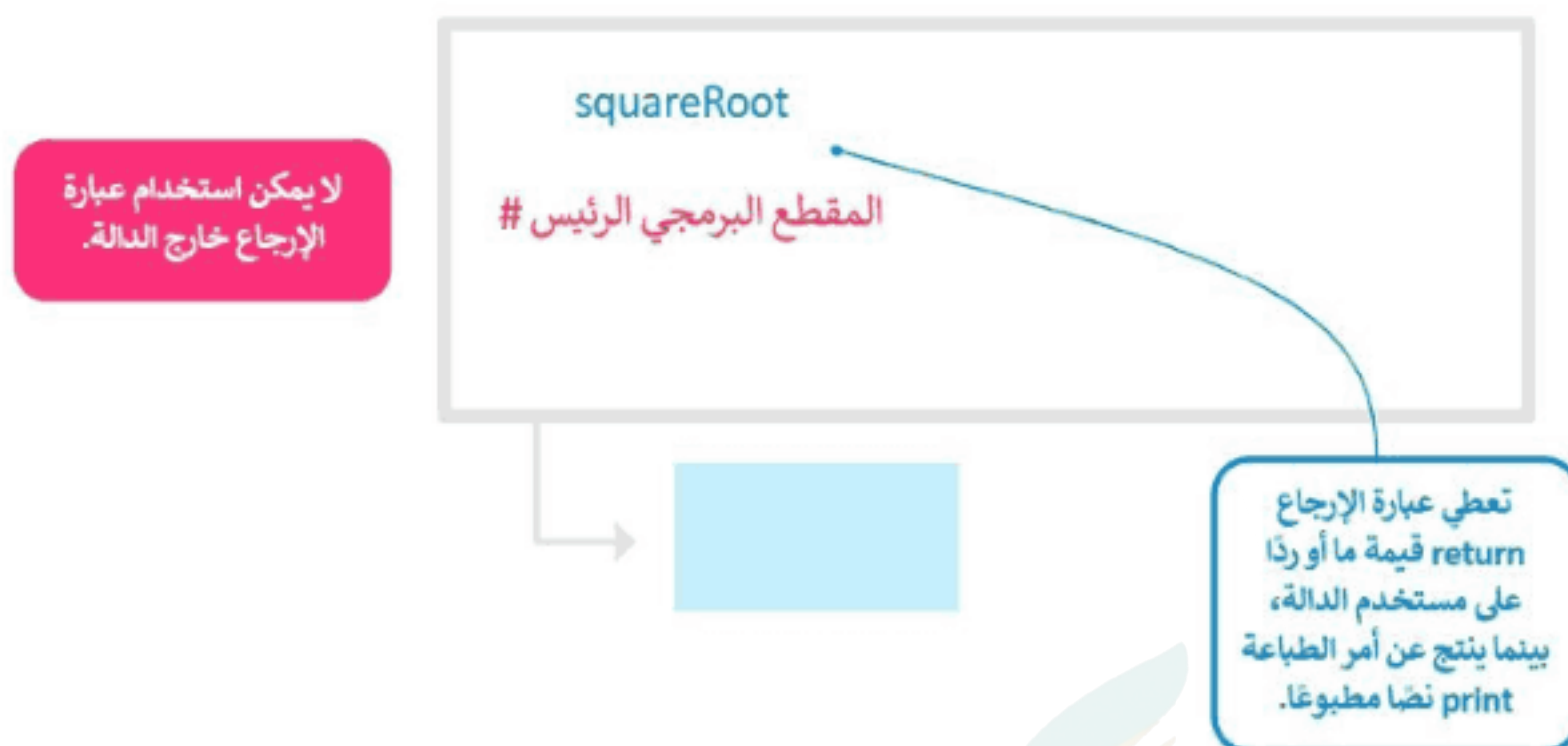
المتغيرات التي يمكنك الإعلان عنها في الدالة تُسمى **المعاملات** (Parameters)، وتُستخدم هذه المتغيرات داخل الدالة. ونظرًا لأنه لا يمكن الوصول للمعاملات إلا من خلال الدالة، فإنه يُشار إليها باسم **المتغيرات المحلية** (Local Variables)، كما يُشار إلى المتغيرات التي تم تمريرها إلى الدالة لتنفيذها على أنها **وسائط** (Arguments). وتتلقي المتغيرات المحلية للدالة قيم الوسائط كمدخلات ثم تتابع تنفيذ الدالة.

ألقي نظرة على المثال الآتي، حيث تحدد دالة تسمى **printMax** مع معاملي **a** و **b** وتجد أكبر عدد باستخدام جملة **if else** الشرطية، ثم تطبع هذا العدد باستخدام دالة **الطباعة** (print()).



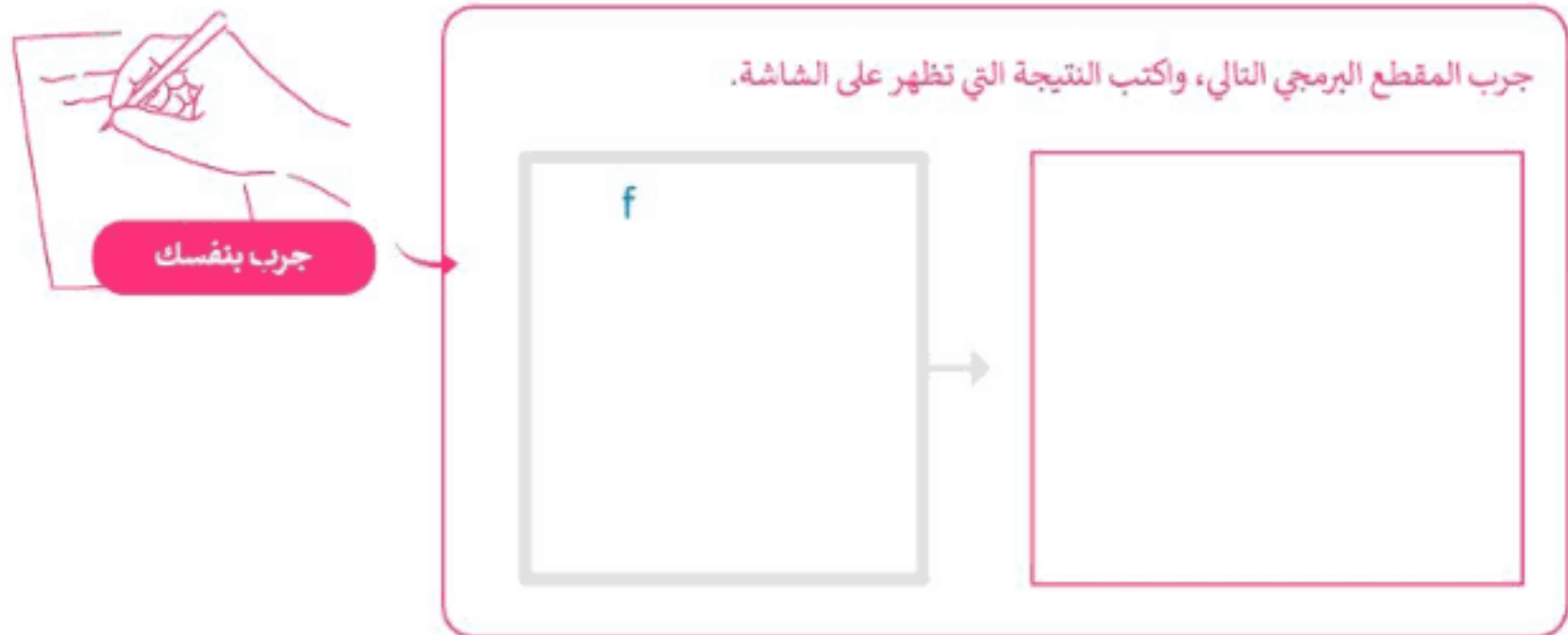
عبارة الإرجاع

في بايثون، تُستخدم عبارة الإرجاع (Return statement) لإنهاء تنفيذ استدعاء الدالة وإرجاع قيمة التعبير. يؤدي ظهور عبارة الإرجاع في أي نقطة من محتوى الدالة إلى إنهاء تنفيذ أوامر الدالة وإعادة المقطع البرمجي إلى النقطة التي تلي أمر استدعائه مباشرة.



في المثال أدناه، تُرجع دالة الحد الأقصى (maximum) أكبر القيم التي تم إدخالها، وللقيام بذلك تُستخدم جملة `if else` الشرطية للعثور على القيمة الأعلى ثم إرجاع تلك القيمة.

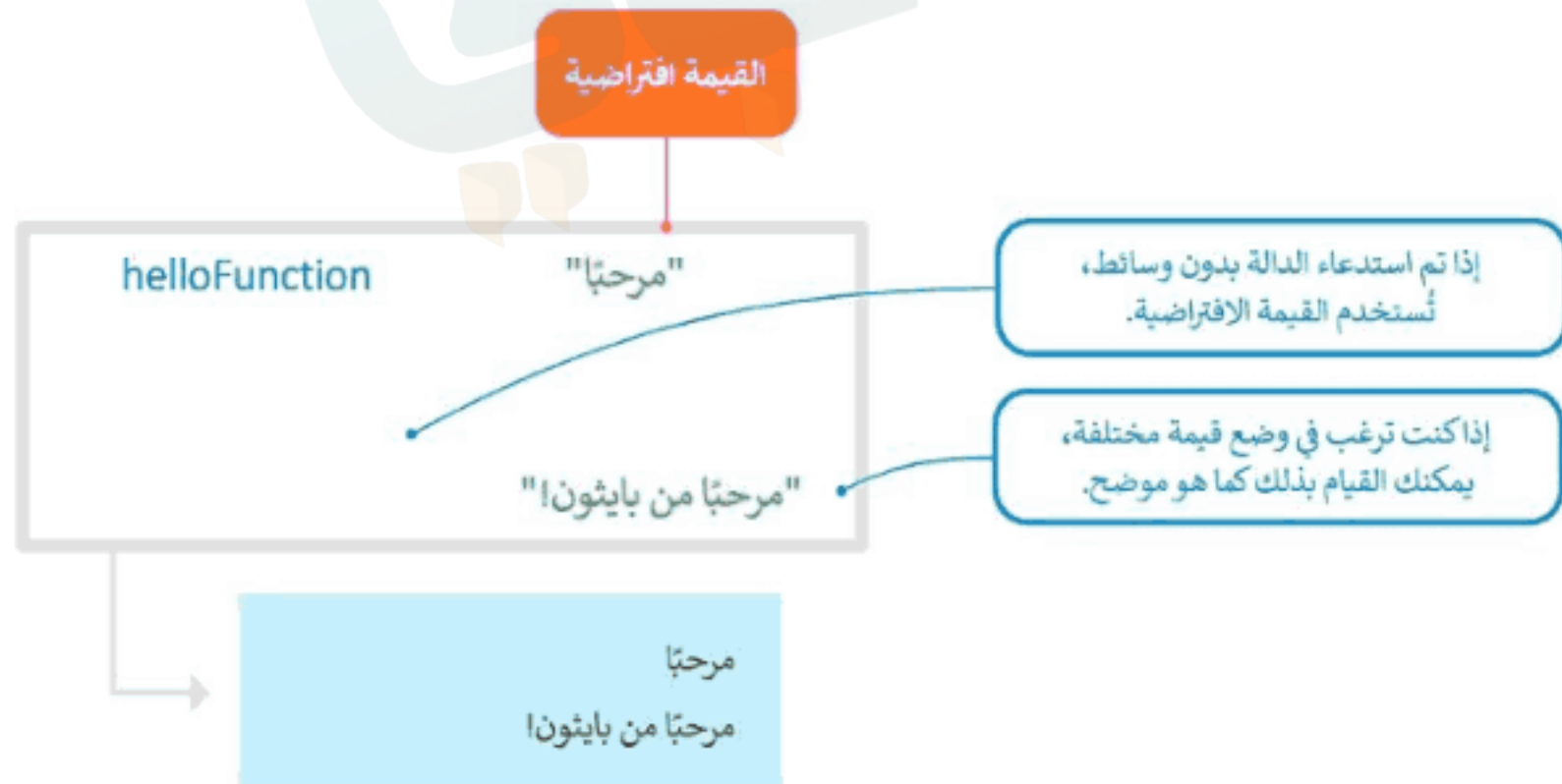




الوسائط الافتراضية

في بعض الدوال، قد ترغب في جعل بعض المعاملات اختيارية واستخدام القيم الافتراضية إذا كان المستخدم لا يريد إعطاء قيم لهذه المعاملات. يمكن تحقيق ذلك بمساعدة قيم الإعداد الافتراضي. إذا لم يتم تمرير قيمة أخرى أثناء تسمية الدالة، فستأخذ وسائط الدالة قيم الإعداد الافتراضية.

يمكنك تعيين وسائط افتراضية (Default arguments) للمعاملات عن طريق وضع عامل الإحالة (=) متبوعًا بالقيمة الافتراضية بعد تسمية المعاملات في تعريف الدالة.



المتغيرات المحلية والعامة

عند تحديد المتغيرات داخل تعريف الدالة، فإنها لا تؤثر ولا تتأثر بالمتغيرات الأخرى التي تحمل نفس الاسم والتي يتم استخدامها خارج تلك الدالة. يسمى جزء المقطع البرمجي الذي يُمكن الوصول إلى المتغير واستخدامه بنطاق المتغير، **المتغيرات المحلية** (Local Variables) نطاقها داخل الدالة حيث تم التصريح عنها، وتبدأ من تعريف اسم الدالة، وتنتهي عند توقف الدالة عن التنفيذ.

في المثال التالي دالة استخدمت لأول مرة معامل يسمى x ، يستخدم بايثون قيمة المعامل المعلن داخل تلك الدالة. ثم ستسند القيمة 2 إلى x ، والاسم x هو متغير محلي لدالتك. لذلك، عندما تغيرت قيمة x في الدالة، فإن x المحددة في الجزء الرئيس لن تتأثر.

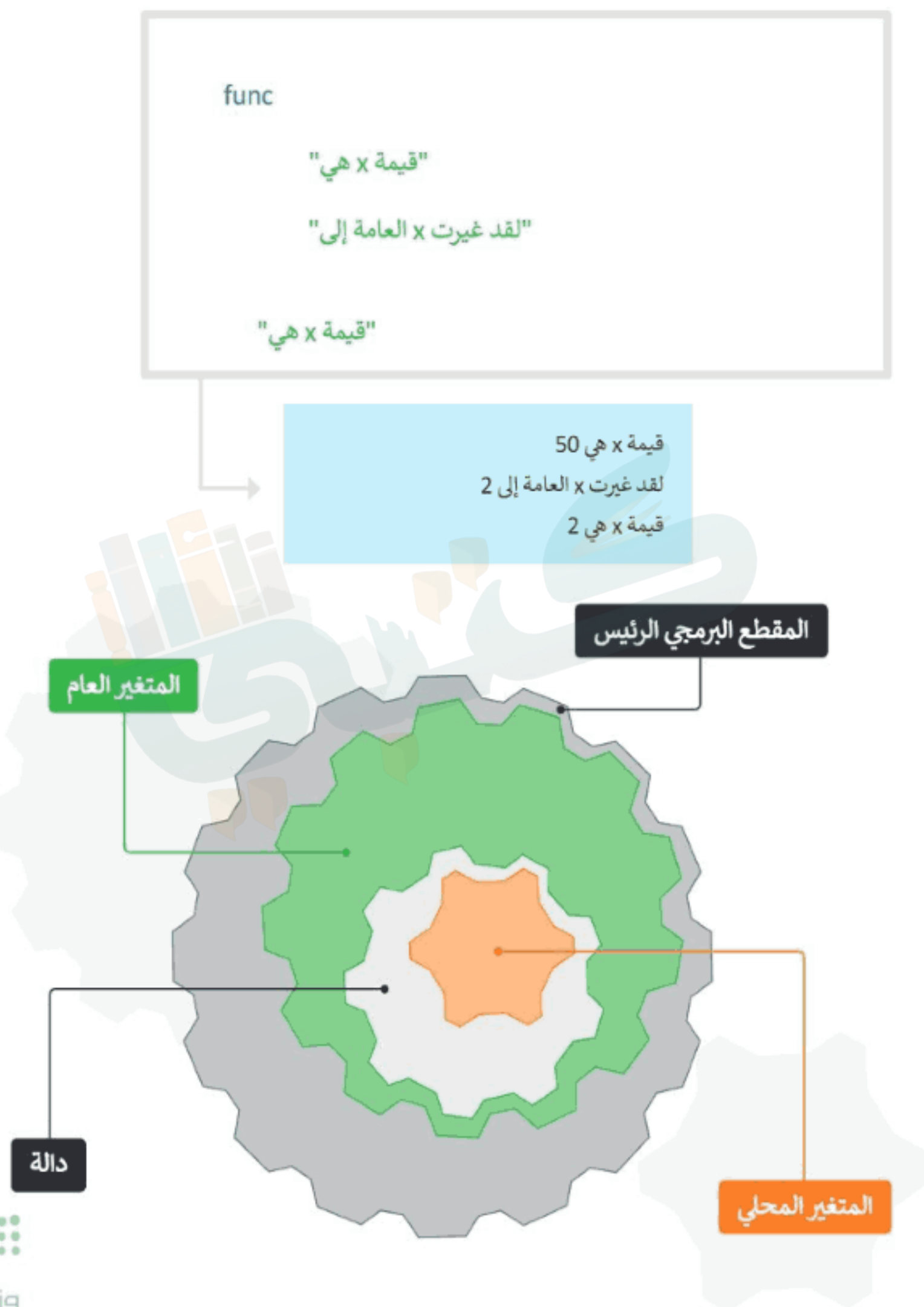
في الاستدعاء الأخير لدالة الطباعة، سُنطبع قيمة x في القسم الرئيس وتؤكد على أنها لم تتأثر.



إذا كنت ترغب في تعيين قيمة لاسم محدد في المستوى الأعلى من جزء المقطع البرمجي (أي ليس ضمن نطاق معين مثل الدوال أو الفئات)، فأنت بحاجة إلى إخبار بايثون أن الاسم ليس محلياً (local) ولكنه عام (global). يتم ذلك باستخدام **المتغيرات العامة** (Global Variables).



في المثال التالي يستخدم فيه الأمر العام (global) للإشارة إلى أن x هو متغير عام، حيث يؤدي إسناد قيمة إلى x داخل الدالة لتغيير قيمة x في الجزء الرئيس.



لنطبق معًا

تدريب 1

🔗 اكتب ناتج المقطع البرمجي التالي:

```
total = 0;
# تعريف الدالة هنا
def sum( arg1, arg2 ):
    total = arg1 + arg2;
    print("المجموع داخل الدالة:", total)
    return total;

# المقطع البرمجي الرئيس
sum( 10, 20 );
print("المجموع خارج الدالة:", total)
```

الناتج

المجموع داخل الدالة : 30
المجموع خارج الدالة : 0



تدريب 2

◀ المقطع البرمجي التالي غير مكتمل. عليك أن تكتب دالة لإيجاد متوسط الأعداد الثلاثة التي أدخلتها في المقطع البرمجي الرئيس.

```
def find_average(num1,num2,num):  
    result=(num1+num2+num)/3;  
    return result;
```

المقطع البرمجي الرئيس

```
num1 = float(input("أدخل الرقم الأول:"))  
num2 = float(input("أدخل الرقم الثاني:"))  
num3 = float(input("أدخل الرقم الثالث:"))  
result = find_average(num1, num2, num3)  
print("متوسط الأرقام هو:", result)
```

تدريب 3

◀ اشرح الفرق بين المتغيرات المحلية والعامة.

تلميح: اطلب من الطلبة حل التدريب، والاستعانة بصفحة 100 في كتاب الطالب عند الحاجة.



تدريب 4

◀ اكتب دالة في بايثون تقبل ثلاثة أرقام وتعيد الرقم الأكبر من بين هذه الأرقام.

تلميح: يمكنك العثور على إجابة التمرين المقترحة في ملف باسم G8.S2.U3.L3_EX4.py على منصة عين الإثرائية.

تدريب 5

◀ اكتب مقطعًا برمجيًا ودالة من أجل طباعة رسالة، على سبيل المثال، "مرحبًا بك في المدرسة يا خالد". اسم المستخدم هو معامل البرنامج.

◀ أنشئ المقطع البرمجي الرئيس الذي يحدد قيمة الوسائط والدالة التي تطبع الرسالة.

◀ أنشئ مقطعًا **تلميح:** يمكنك العثور على إجابة التمرين المقترحة في ملف باسم G8.S2.U3.L3_EX5.py على منصة عين الإثرائية.

تدريب 6

◀ اكتب مقطعًا برمجيًا يسأل الطالب عن درجاته في أربع مواد، ثم أنشئ دالة تأخذ هذه الدرجات كوسائط ويحسب ما إذا كان الطالب قد اجتاز الفصل أم لا.

تلميح: يمكنك العثور على إجابة التمرين المقترحة في ملف باسم G8.S2.U3.L3_EX6.py على منصة عين الإثرائية.



جداول بيانات إكسل في بايثون

لقد تعلمت سابقًا كيفية استخدام إكسل لإجراء معالجة البيانات البسيطة، مثل إنشاء جداول البيانات، وتطبيق الدوال لتحديد البيانات وفرزها وتنسيقها وإنشاء بعض المخططات بناءً على بياناتك. ستتعلم في هذا الدرس كيفية استخدام بايثون إلى جانب إكسل.

العمل مع إكسل و بايثون

نظرًا لأنك تعلم بالفعل كيفية استخدام إكسل، فقد تتساءل عن سبب استخدام بايثون للعمل معه. السبب هو أنه باستخدام بايثون يمكنك أتمتة بعض المهام المتكررة، وإجراء بعض العمليات الحسابية المعقدة بشكل أسرع من العمل مع إكسل فقط.

لاستخدام بايثون مع إكسل، تحتاج إلى مكتبة تسمى أوبيين بيكسل (openpyxl). المكتبة في لغات البرمجة هي عبارة عن حزمة مقاطع برمجية مكتوبة سابقًا، وتؤدي بعض المهام بحيث لا يحتاج المبرمجون إلى كتابة المقطع البرمجي من البداية في كل مرة.

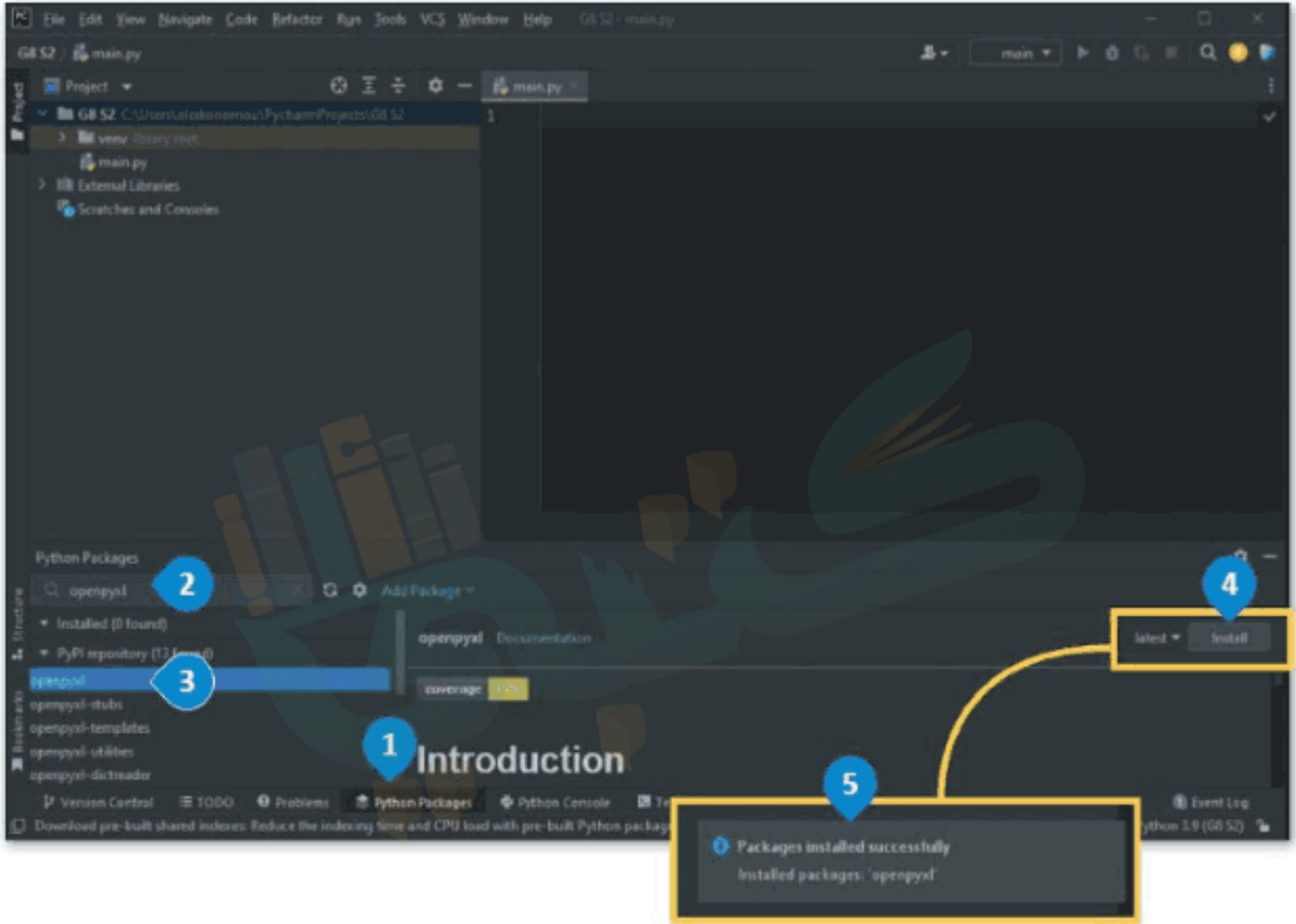
مكتبة أوبيين بيكسل

يوجد في بايثون العديد من المكتبات المثبتة سابقًا، ولكن هناك العديد من المكتبات المتاحة التي يمكنك تنزيلها وتثبيتها، مكتبة أوبيين بيكسل (openpyxl) هي إحدى تلك المكتبات. يمكنك باستخدامها إجراء عمليات مثل القراءة والكتابة والحسابات الرياضية، وإنشاء الرسوم، والمخططات البيانية.



لتثبيت مكتبة أوبين بيكسل (openpyxl):

- 1 < في نافذة PyCharm (بأي تشارم)، اضغط على Packages (الحزم).
- 2 < اكتب openpyxl (أوبين بيكسل) في شريط البحث.
- 3 < اختر openpyxl (أوبين بيكسل)، واضغط على Install (تثبيت).
- 4 < ستظهر رسالة تُخبرك بأن التثبيت قد اكتمل.



استيراد مكتبة أوبين بيكسل

لاستخدام مكتبة أوبين بيكسل بعد تثبيتها، تحتاج إلى استيرادها. يمكنك القيام بذلك باستخدام أمر الاستيراد (import).

استيراد مكتبة أوبين بيكسل

العمل مع دفاتر العمل

كما تعلم، يمكن أن تحتوي ملفات إكسل على أكثر من ورقة عمل واحدة، ويمكنك إدخال البيانات ومعالجتها في كل ورقة. في مكتبة أوبين بيكسل، يُطلق على ملف إكسل اسم دفتر العمل (Workbook)، ويمكنك إما إنشاء ملف إكسل جديد أو تحميل ملف موجود سابقًا.

دالة تحميل دفتر العمل (load_workbook())

للعمل على ملف إكسل موجود ستستخدم دالة تحميل دفتر العمل (load_workbook()). وعليك وضع الملف في نفس المجلد مع جزء المقطع البرمجي، أو استخدام المسار المطلق للملف.

سمات أسماء أوراق العمل (sheetnames)

عادةً ما تعمل على ورقة محددة من دفتر العمل، ومن المفيد جدًا معرفة أسماء هذه الأوراق. لرؤية أسماء الأوراق التي يحتوي عليها دفتر العمل، يمكنك استخدام سمة أسماء أوراق العمل (sheetnames).



سمة الصف الأقصى (max_row attribute) و سمة العمود الأقصى (max_column attribute)

عند العمل باستخدام ملف إكسل، فمن الجيد معرفة عدد الصفوف والأعمدة الموجودة في الورقة. للقيام بذلك، يمكنك استخدام سمة الصف الأقصى (max_row) و سمة العمود الأقصى (max_column) في ورقة العمل.



الوصول إلى الخلايا

تحتوي كل خلية في ورقة عمل إكسل على خاصيتين رئيسيتين، هما موقعها في الملف وقيمته. غالبًا ما ستحتاج إلى الوصول إلى قيمة الخلية من أجل طباعتها على الشاشة، أو تغييرها، أو استخدامها لإجراء عمليات حسابية أخرى مثل العثور على مبلغ وما إلى ذلك.

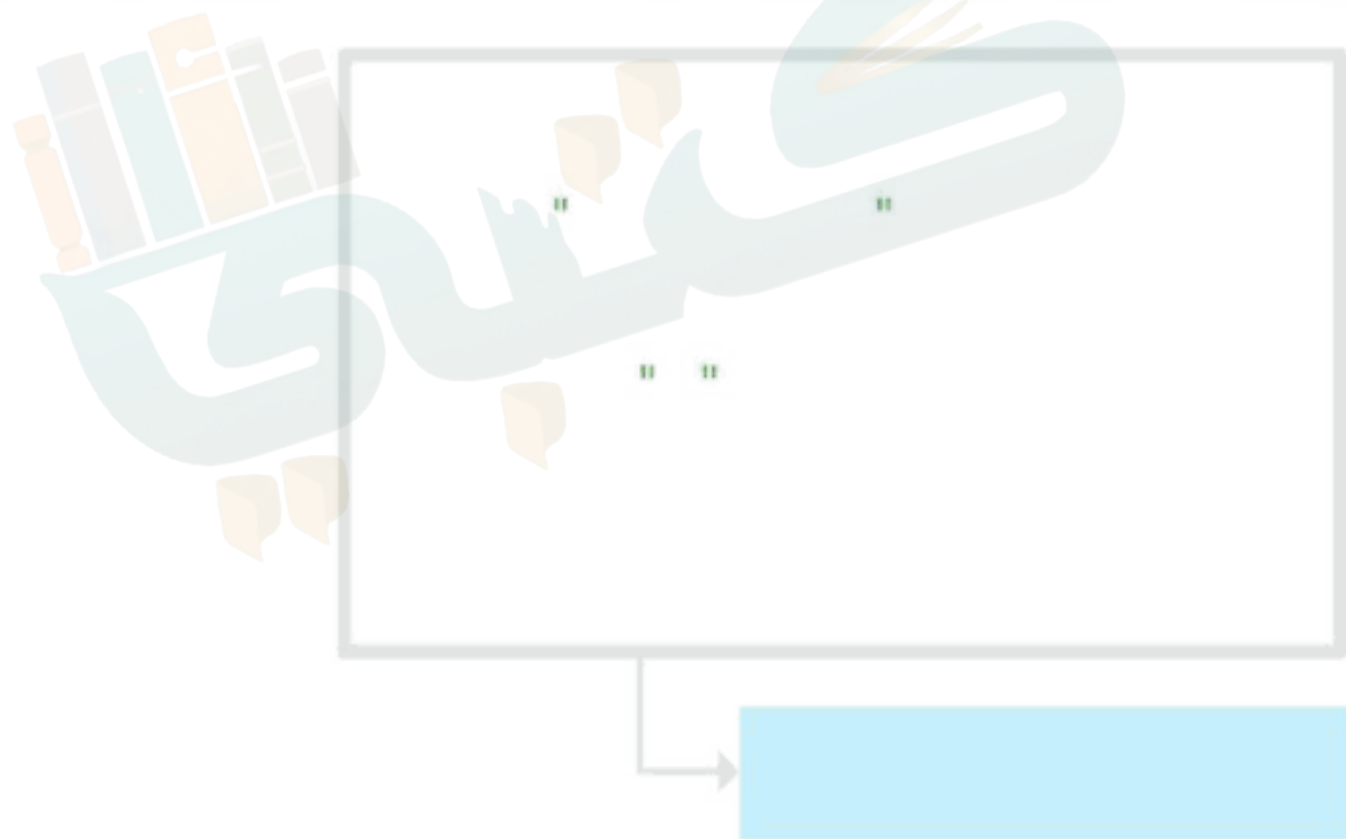


الوصول إلى قيم الخلايا

لقد تعلمت كيفية الحصول على موقع الخلية. لكن في معظم الأوقات، تحتاج إلى قيمتها. للحصول على قيمة الخلية، تحتاج إلى استدعاء سمة قيمة كائن خلية معين.

في هذا المثال ستستخدم ملف إكسل للتغذية لطباعة عدد السعرات الحرارية (kCal) من الحليب. كما رأيت من قبل، يحتوي هذا الملف على 12 صفًا و12 صفًا. للحصول على عدد السعرات الحرارية (kCal) من الحليب، عليك أولاً العثور على موقعها، وهو "C3" ثم يمكنك استدعاء سمة القيمة في هذه الخلية.

	K	J	I	H	G	F	E	D	C	B	A	
1	100 جرام											
2	مكونات	الماء	الطاقة	البروتين	الدهون	الكربوهيدرات	الكالسيوم	الفوسفور (P)	الحديد	الصوديوم	البوتاسيوم (K)	
3	الحليب	88.1	61	3.27	3.25	4.63	123	101	0	38	150	
4	البقي	75.8	143	12.4	8.65	0.96	48	184	1.67	129	132	
5	الدجاج	69.9	149	23.9	5.23	0	12	184	0.94	117	239	
6	برجر بالحن	48	261	13.9	11.8	23.9	239	201	2.46	508	211	
7	النفاخ	83.6	65	0.15	0.16	15.6	6	10	0.02	1	104	
8	الكعك	8.85	430	5.79	13.7	69.6	29	111	2.28	314	245	
9	منتجات بالشوكولاتة	55.7	216	3.8	11	28.2	109	107	0.93	76	249	
10	حليب بالشوكولاتة	1.5	535	7.65	29.7	59.4	189	208	2.35	79	372	
11	الحس	94.7	20	1.24	0.26	3.24	35	30	0.95	0	253	
12	الموز	74.9	89	1.09	0.33	22.8	5	22	0.26	1	358	

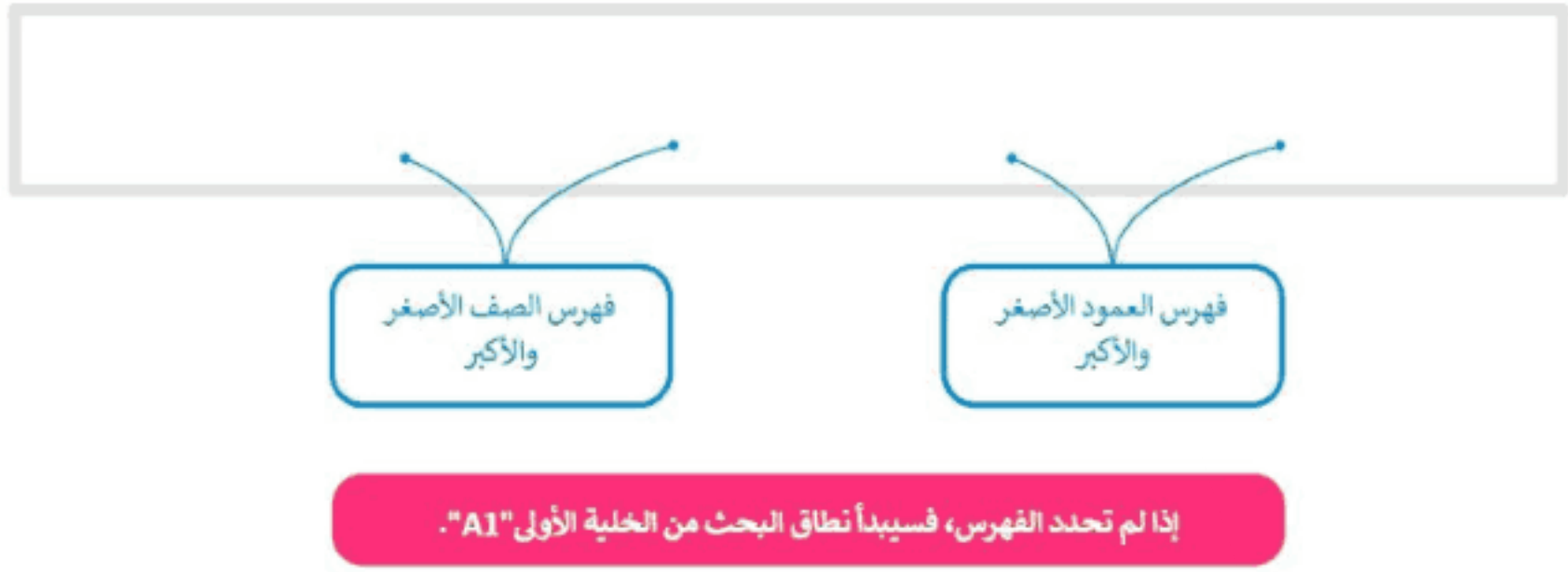


الوصول إلى قيمة أكثر من خلية واحدة

افترض الآن أنك تريد الحصول على جميع الحقائق الغذائية للحليب. يمكنك العثور يدويًا على موقع كل خلية ثم استدعاء سمة قيمتها، ولكن ذلك سيستغرق وقتًا طويلاً ومن الممكن أيضًا أن تكون قد ارتكبت بعض الأخطاء. لذلك، يوجد دالتان لهما وظيفتان مفيدتان في اختيار أكثر من خلية واحدة وهما `iter_rows` و `iter_cols`.

دالة iter_rows

تُستخدم دالة `iter_rows` عندما تريد الحصول على جميع القيم لصف أو أكثر في ورقة عمل إكسل.



ستستخدم دالة `iter_rows` الآن للحصول على جميع الحقائق الغذائية للحليب. عليك الحصول على جميع مواقع الخلايا لصف الحليب، وهو "3D" ثم استدعاء سمة القيمة.

أنشئ متغير صفوف يحتوي على مواقع حقائق التغذية الخاصة بالحليب #

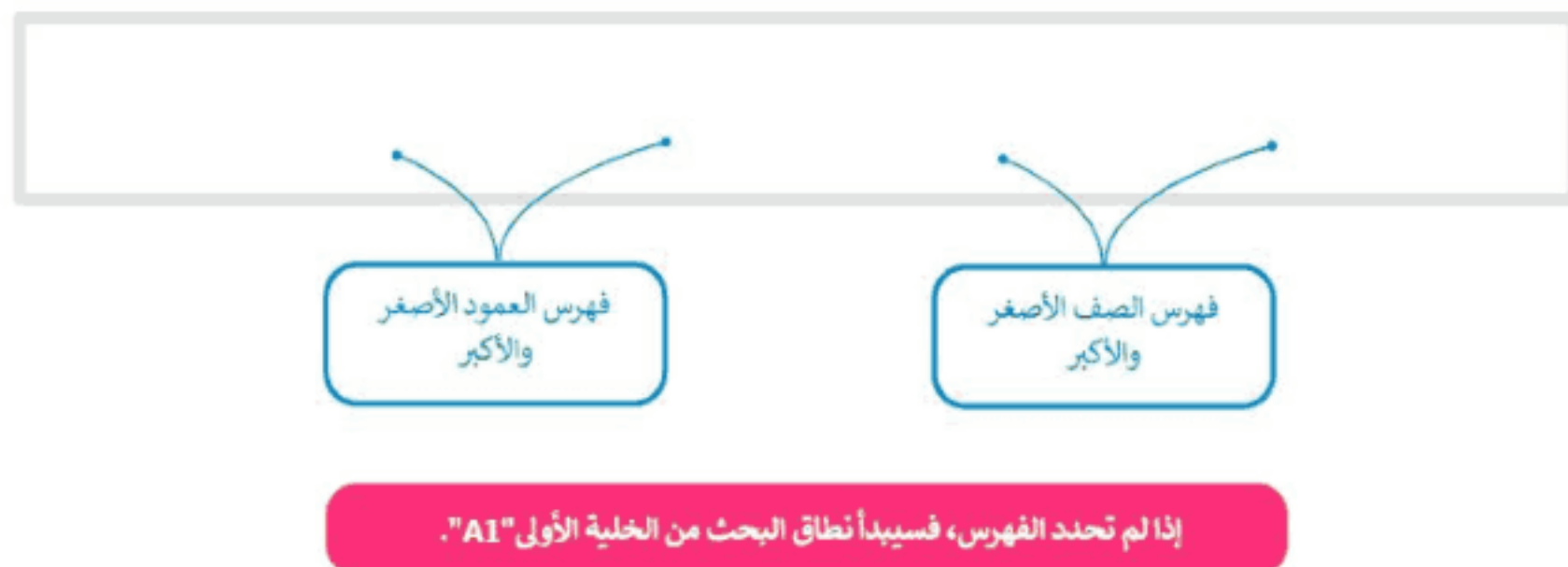
المتداخلة لطباعة قيمة كل خلية `rof` الآن استخدم حلقات #
"حقائق غذائية عن الحليب:"

" "

حقائق غذائية عن الحليب: 150 38 0 101 123 5.63 3.25 3.27 61 88.1



تُستخدم دالة Iter_cols عندما تريد الحصول على جميع القيم لعمود أو أكثر في ورقة عمل إكسل.



ستستخدم Iter_cols للحصول على عدد السعرات الحرارية (kCal) لجميع المكونات الموجودة في ورقة العمل. ستحتاج إلى الحصول على مواقع جميع الخلايا ثم استدعاء سمة القيمة.



كتابة القيم

لقد تعلمت كيفية الحصول على الخلايا والقيم من ورقة العمل. من الأمور المهمة عند العمل على ملفات إكسل هي كتابة القيم في الخلايا، حيث يمكنك إما إنشاء صفوف وأعمدة جديدة وإضافة قيم في خلايا فارغة، أو يمكنك تغيير قيم الخلايا الموجودة. يمكنك تغيير قيمة الخلية باستخدام دالة الخلية (cell).



ستعمل على ملف التغذية في إكسل وتغير القيم الموجودة. في هذا الملف، ستحسب قيم العناصر الغذائية لكل مكون في كل 100 جرام. إذا كنت ترغب في تحويل هذه القيم لإظهارها في كل 30 جرامًا من المكونات، فيجب عليك تغيير كل قيمة وفقًا للصيغة التالية:

$$\text{new_value} = \text{old_value} * 30 / 100$$

عند حفظ التغييرات في ملف إكسل من خلال أوامر بايثون، يجب فتح الملف فقط في بايثون وإلا فلن تحدث التغييرات، وستظهر رسالة خطأ في حال كان ملف الإكسل مفتوحًا في برنامج إكسل في نفس الوقت.

عليك أولاً تغيير قيمة "B1" من "100 جرام" إلى "30 جرام".



لحفظ دفتر العمل

الآن إذا فتحت ملف إكسل مرة أخرى، فستتغير قيمة الخلية "B1".

	K	J	I	H	G	F	E	D	C	B	A	
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												

لتغيير كل خلية تغذية عليك بتطبيق الصيغة ثم حفظ دفتر العمل. للقيام بذلك، عليك استخدام حلقتين **for** ، واحدة تمر عبر كل صف والأخرى تمر عبر كل عمود.



لأن قيم الخلايا هي من نوع كائن (object)، ستحتاج إلى تحويلها إلى عدد عشري (float) لإجراء العمليات الحسابية

	K	J	I	H	G	F	E	D	C	B	A	
1					30 جرام							
2					الكالسيوم (Ca) ملليجرام	الكربوهيدرات جرام	الدهون جرام	البروتين جرام	الطاقة سعر حراري	الماء جرام	المكونات	
3					36,9	1,389	0,975	0,981	18,3	26,43	الحليب	
4					14,4	0,288	2,595	3,72	42,9	22,74	أبيض	
5					3,6	0	1,569	7,17	44,7	20,97	الدجاج	
6					71,7	7,17	3,54	4,17	78,3	14,4	برجر بالجين	
7					1,8	4,68	0,048	0,045	19,5	25,08	الفلاح	
8					8,7	20,88	4,11	1,737	129	2,655	الكعك	
9					32,7	8,46	3,3	1,14	64,8	16,71	مكسبات بالشوكولاتة	
10					56,7	17,82	8,91	2,295	160,5	0,45	حليب بالشوكولاتة	
11					10,5	0,972	0,078	0,372	6	28,41	الخبز	
12					1,5	6,84	0,099	0,327	26,7	22,47	الموز	

كما ترى فقد تغيرت جميع القيم الغذائية، واستغرقت هذه المهمة وقتًا أقل بكثير من تغيير كل قيمة يدويًا. هذا النوع من الأتمتة مفيد للغاية خاصة إذا كنت تعمل مع ملفات كبيرة جدًا.



لنطبق معًا

تدريب 1

اكتب ناتج المقطع البرمجي التالي:

```
import openpyxl

wb = openpyxl.load_workbook(filename="nutrition.xlsx")
sheet_obj = wb["arabic"]

rows = sheet_obj.iter_rows(min_row=3, max_row=5, min_col=1, max_col=11)
for row in rows:
    for cell in row:
        print(cell.value, end=" ")
    print("")
```

الناتج

```
150 38 0 101 123 4.63 3.25 3.27 61 88.1 الحليب
132 129 1.67 184 48 0.96 8.65 12.4 143 75.8 البيض
239 117 0.94 184 12 0 5.23 23.9 149 69.9 الدجاج
```



تدريب 2

ما وظيفة المقطع البرمجي التالي؟ وما مخرجاته؟

```
import openpyxl

wb = openpyxl.load_workbook(filename="nutrition.xlsx")
sheet_obj = wb["arabic"]

for row in range(3,13):
    value = float(sheet_obj.cell(row=row, column=5).value)
    if (value < 2):
        print(sheet_obj.cell(row=row, column=1).value)
```

الناتج

التفاح
الخس
الموز

تلميح: عند تشغيل المقطع البرمجي فإنه يعرض المنتجات التي تحتوي على أقل من 2 جرام من الدهون.



تدريب 3

⬅ اكتب مقطعًا برمجيًا باستخدام مكتبة أوبين بيكسل (openpyxl)، يعمل على تحميل مجموعة بيانات ملف التغذية و طباعة قيم عناوين الأعمدة.

تلميح: يمكنك العثور على إجابة التمرين المقترحة في ملف باسم G8.S2.U3.L4_EX3.py على منصة عين الإثرائية.

تدريب 4

⬅ اكتب مقطعًا برمجيًا باستخدام مكتبة أوبين بيكسل (openpyxl)، يعمل على تحميل مجموعة بيانات ملف التغذية و طباعة جميع القيم الغذائية الأصغر من 25.

تلميح: يمكنك العثور على إجابة التمرين المقترحة في ملف باسم G8.S2.U3.L4_EX4.py على منصة عين الإثرائية.





مشروع الوحدة

في هذا المشروع، عليك إنشاء دالة بايثون تصمم نمط الساعة الرملية باستخدام النجوم. سيعطى حجم نمط الساعة الرملية كمداخلات في المقطع البرمجي الرئيس.
يتكون هذا النمط في الواقع من نموذجين. النموذج الأول هو نمط هرم يشير للأسفل، والنموذج الثاني هو نمط هرم يشير للأعلى. أنشئ نمط الساعة الرملية من خلال الجمع بين هذين النموذجين.

```
*****
*****
*****
***
*
***
*****
*****
*****
```

1 أنشئ دالة تأخذ رقمًا كإدخال ثم ترسم النمط.

أ استخدم حلقات **for** من أجل رسم نمط هرم يشير للأسفل.

ب استخدم حلقات **for** من أجل رسم نمط هرم يشير للأعلى.

2 أنشئ مقطعًا برمجيًا بلغة بايثون:
< اطلب من المستخدم إعطاء رقم صحيح (حجم الساعة الرملية).

3 استدع الدالة لإنشاء النمط.

4 شغل المقطع البرمجي وتحقق من صحته وعدم وجود أخطاء.



في الختام

جدول المهارات

درجة الإتقان		المهارة
لم يتقن	أتقن	
		1. إنشاء مقطع برمجي في بيئة التواصل باي تشارم.
		2. استخدام المعاملات الشرطية.
		3. استخدام المعاملات المنطقية.
		4. استخدام الجملة الشرطية البسيطة.
		5. استخدام الجملة الشرطية <code>if..else</code> .
		6. استخدام الجملة الشرطية المتداخلة.
		7. إنشاء مقطع برمجي باستخدام حلقة <code>for</code> ، وحلقة <code>while</code> الشرطية.
		8. استخدام عبارة الإيقاف لإنهاء حلقة.
		9. إنشاء مقطع برمجي باستخدام حلقات متداخلة.
		10. إنشاء أنماط باستخدام الحلقات.
		11. إنشاء دوال جديدة.
		12. استيراد مكتبة إلى جزء مقطع برمجي.
		13. عمل جداول البيانات باستخدام بايثون.



Library	مكتبة	Attributes	سمات
local Variables	المتغيرات المحلية	Arguments	الوسائط
Logical Operator	المعامل المنطقي	Boolean	القيمة المنطقية
Multiple Decisions	قرارات متعددة	Code	المقطع البرمجي
Nested Loops	الحلقات المتداخلة	Condition	الشرط
Nesting	التداخل	Conditional Operator	المعامل الشرطي
Nesting Conditions	الشروط المتداخلة	Decision	قرار
Operators	المعاملات	Function	الدالة
Outer Loop	الحلقة الخارجية	Global Variables	المتغيرات العامة
Parameter	مُعامل	Infinite Loop	الحلقة اللانهائية
Pattern	النمط	Inner Loop	الحلقة الداخلية
Statement	عبارة	Indentation	المسافة البادئة

الإجابة عن أسئلة قسم "اختبر نفسك"

السؤال الأول

خطأ	صحيحة	حدد الجملة الصحيحة والجملة الخطأ فيما يلي:
✓		1. .ai .eps .svg .dwg هي امتدادات لملفات الفيديو.
	✓	2. في الفيديو عالي الدقة (HD - High Definition)، تكون صورة الفيديو أو عدد الإطارات بالثانية التي يتم التقاطها أعلى من فيديو الدقة القياسية (SD - Standard Definition).
✓		3. يستخدم برنامج ترميز VC-1 لمقاطع الفيديو عالية الدقة وبث مقاطع الفيديو على الإنترنت.
	✓	4. يمكن أن تتضمن "الحاوية" ترجمات.
	✓	5. بالنسبة للأصوات البشرية، يمكن استخدام مُعدّل البت (Bit rate) يتراوح بين 64 و 128 كيلوبت / ثانية.
✓		6. في برنامج تحرير الفيديو المختصر، يمكنك استخدام مُرشح البني الداكن (Sepia Tone) للحصول على صورة بدون ألوان.
✓		7. درجات الألوان وتصحيح الألوان هي نفسها.
	✓	8. يستخدم تنسيق صور "GIF" بشكل أساسي لرسومات الإنترنت لدعمها للشفافية.
✓		9. ملفات صور "JPEG" مناسبة للمنشورات الاحترافية والمنشورات ذات التنسيق الكبير.
✓		10. تنسيقات الصوت غير المضغوطة الأكثر شيوعًا هي MP3 و WMA.
✓		11. يعرض اسم الملف نوع الملف الذي هو عليه.



الإجابة عن أسئلة قسم "اختبر نفسك"

السؤال الثاني

اختر الإجابة الصحيحة:		
☐	جودة أقل بالإضافة إلى مساحة تخزين أقل مطلوبة.	1. في الملفات الصوتية، مُعدّل العينة (Sample rate) الأعلى يعني:
☒	جودة أعلى بالإضافة إلى مساحة تخزين أكبر.	
☐	جودة أعلى بالإضافة إلى مساحة تخزين أقل.	
☐	مخطط القصة (Storyboard).	2. الجزء الأول من التخطيط السينمائي هو:
☐	جدول التصوير (Découpage).	
☒	النص (Script).	
☐	الأحداث (Events).	3. لا يعد مكونًا من مكونات البرنامج النصي:
☒	جدول التصوير (Découpage).	
☐	الحوار (Dialogue).	
☐	TIFF	4. لا يعد تنسيقًا للصورة:
☐	JPEG	
☒	MPEG-4	



الإجابة عن أسئلة قسم "اختبر نفسك"

السؤال الثالث

خطأ	صحيحة	حدد الجملة الصحيحة والجملة الخاطئة فيما يلي:
	✓	1. تستخدم مخططات المعلومات البيانية لنقل رسالة محددة بسرعة.
	✓	2. أحد الخصائص الرئيسة لمخططات المعلومات البيانية هي التوازن.
✓		3. الجدول الزمني ليس من أنواع مخططات المعلومات البيانية.
	✓	4. الخطوة الأولى في تصميم مخطط المعلومات البياني هي اختيار موضوع.
	✓	5. لتعديل مظهر أي عنصر حدده ثم استخدم الشريط الجانبي.
✓		6. لا يمكنك تغيير حجم كل عنصر من تصميمك في كانفا.
	✓	7. يمكنك تصدير مخطط المعلومات البياني كملف PDF.
	✓	8. تكون تصميماتك متاحة في صفحة كانفا الرئيسة.
✓		9. لا يمكنك إنشاء مخطط معلومات بحجم مخصص في كانفا.
	✓	10. تساعد الخلفية على التركيز على عناصر التصميم الأساسية لمخطط المعلومات البياني.
	✓	11. تساعد الصور في إنشاء اتصال مع النص ويمكن أن توضح المعلومات المقدمة في مخطط المعلومات البياني.
	✓	12. الطباعة من خلال تطبيق كانفا لها تكلفة إضافية.



الإجابة عن أسئلة قسم "اختبر نفسك"

السؤال الرابع

خطأ	صحيحة	حدد الجملة الصحيحة والجملة الخطأ فيما يلي:
	✓	1. يمكنك استخدام كانفا لإنشاء كتاب إلكتروني.
✓		2. لا يمكنك تحميل صورك الخاصة في كانفا.
✓		3. لا يمكنك إنشاء مخطط المعلومات البياني الخاص بك دون استخدام قالب في كانفا.
	✓	4. يمكنك نقل العناصر في كانفا باستخدام طريقة السحب والإفلات.
✓		5. يمكنك حذف عنصر من تصميمك بالضغط على مفتاح Enter.
	✓	6. لا يمكنك إنشاء حساب في كانفا باستخدام إكس X (تويتر سابقًا).
✓		7. كانفا مخصص للمصممين فقط.
✓		8. يمكنك تنزيل تصميم من كانفا بتنسيق exe.
✓		9. التسجيل لاستخدام كانفا اختياري.
	✓	10. يحفظ كانفا تصميماتك تلقائيًا.
✓		11. يمكنك وضع عنصر في كانفا من خلال الضغط عليه.
	✓	12. يمكنك استيراد وتحرير ملف PDF في كانفا.



الإجابة عن أسئلة قسم "اختبر نفسك"

السؤال الخامس

❖ في الجدول التالي، يمكنك رؤية قيمة كل متغير أثناء تنفيذ المقطع البرمجي. املأ الفراغ في كل صف من المقطع البرمجي حسب الجدول:

الشرط	z	y	x
<code>x=int(input("من فضلك أدخل قيمة المتغير x:"))</code>			6
<code>y=int(input("من فضلك أدخل قيمة المتغير y:"))</code>		3	
<code>__z__=0</code>	0		
<code>if x>__y:</code>			True
<code>__z__=x+__y</code>	9		
<code>print (__z__, __y__, __x__)</code>			
<code>if x!=__y__ and y!=__z__:</code>			True
<code>__x__=x-__y</code>			3
<code>y=x-__y</code>		0	
<code>z=__x__+y</code>	3		
<code>print (__x__, __y__, __z__)</code>			

تلميح: نرجوا الملاحظة أن العبارة الصحيحة هنا هي

("من فضلك أدخل قيمة المتغير y:")

سيتم تصحيح هذه العبارة في إصدار الكتاب العام المقبل. اطلب من الطلبة استخدام الكتاب الرقمي في هذا التدريب.

ك أدخل قيمة المتغير x:

6

ك أدخل قيمة المتغير y:

3

9 3 6

3 0 3



وزارة التعليم

Ministry of Education
2025 - 1447

الإجابة عن أسئلة قسم "اختبر نفسك"

السؤال السادس

اكتب ناتج المخطط الانسيابي التالي

مستخدماً القيم:

a **5 17** ناتج المخطط الانسيابي: a: x=5, y=12

b **17 5** ناتج المخطط الانسيابي: b: x=12, y=5



الإجابة عن أسئلة قسم "اختبر نفسك"

السؤال السابع

أدخل قيم المتغيرات أثناء تنفيذ المقطع البرمجي. افترض أن إدخال المتغير x هو دائما الرقم 15.

```
a = 0
for i in range(1,6):
    x = int(input())
    a = a + x
    if i%5 == 0:
        y = a / 5
        print(a,x,y)
```

النتاج	الشرط	y	x	a	i
					1
			15		
				15	
	False				
			15		2
				30	
	False				
			15		3
				45	
	False				
			15		4
				60	
	False				
			15		5
				75	
	True				
		15			
75,15,15					



الإجابة عن أسئلة قسم "اختبر نفسك"

السؤال الثامن

اكتب ناتج المخطط الانسيابي التالي مستخدماً القيم:

- a **13** ناتج المخطط الانسيابي: $n=7$
- b **55** ناتج المخطط الانسيابي: $n=10$

