ملخص تقن 102 الشامل

التواصل الاجتماعي بحسب الصف الأول					
صفحة مناهج مملكة البحرين على فيسبوك					
المواد على تلغرام	التربية الاسلامية	اللغة العربية	اللغة الانجليزية	الرياضيات	

المزيد من الملفات بحسب الصف الأول والمادة علوم وتقانة في الفصل الأول

تقن 107

مملكة البحرين
وزارة التربية والتعليم
مدرسة المحرق الثانوية للبنين

البرمجة بلغة البايثون

هذا الملخص لايغني عن الكتاب المدرسي

2025

2026

إعداد / أ. حازم طه الحمامصي

37780935

فهرس مقررتقن 107

python™ programming

إعداد / أ. حازم طه الحمامصي

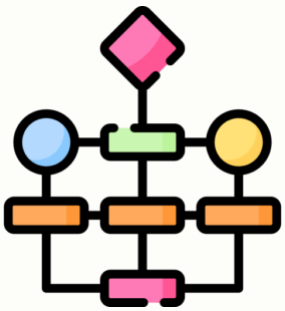
37780935

الخوارزميات وحل المشكلات

تعريف الخوارزمية Algorithm

هي مجموعة من الخطوات أو الأوامر التي تكون متتالية بتسلسل منطقي للأحداث قصد الوصول إلى الهدف او النتيجة المطلوبة.

ينبغي أن تكون خطوات وتعليمات الخوارزمية واضحة ومُرتّبة بحيث تنتهي بحل المشكلة. فكّر في الخوارزمية كما لو كانت وصفة طبخ، إذ تقدّم الوصفة طريقة تحضير وجبة ما خطوة بخطوة، ابتداءً بالمقادير الضرورية للوجبة، وحتى آخر خطوة من تحضير الوجبة وتقديمها.(أذكر خطوات تحضير وجبة الإفطار؟)



مثال خوارزمية ضرب عددين صحيحين:

- الخطوة 1: ابدأ
- الخطوة 2: قم بالإعلان عن ثلاثة أعداد صحيحة X ,Y, Z
- الخطوة 3: أدخل قيم المدخلات X,Y
- الخطوة 4: اضرب قيم Y بـ X
- الخطوة 5: خزّن ناتج الضرب في Z
- الخطوة 6: اعرض قيمة Z
- الخطوة 7: توقف

عندما تواجهك أي مشكلة وتريد كتابة خوارزمية لحلها عليك أن تفكر في العثور على إجابات للأسئلة التالية:

- ما هي مدخلات أو معطيات الخوارزمية أي ما هي المعلومات التي أحتاج إلى الحصول عليها من المستخدم؟
- ما هي مخرجات الخوارزمية أي ما هي المعلومات التي أحتاج لعرضها على المستخدم؟
- ما هي الخطوات الرئيسية المطلوبة لحل هذه المشكلة؟
- ما هو ترتيب تنفيذ هذه الخطوات؟
- ما هي القرارات أو الشروط التي أحتاج مراعاتها عند معالجة المعلومات؟
- هل هناك تعليمات بحاجة لأن أكررها عدة مرات؟

مثال خوارزمية قيادة السيارة:

رتب الخطوات منطقياً للانطلاق بالسيارة للذهاب لمكان ما

2- ربط حزام الأمان

3- تعديل المرآة الوسطى

4- الجلوس على مقعد القيادة

5- افتح السيارة

6- أشغل السيارة

7- انطلق بالسيارة

1- تغيير السرعات

الترتيب

غير بالأرقام ترتيب الخطوات دون التأثير على انطلاق السيارة

غير بالأرقام ترتيب الخطوات بحيث لا يمكن انطلاق السيارة

مثال : أكتب الخطوات اللازمة لتنفيذ برنامج بيتقبل عددان ويقوم بتخزينهما في متغيرين يتم ضرب العددين وطباعة النتيجة .

1- تعريف متغيرين مثلاً X , Y وإدخالهما من المستخدم
2- تعريف متغير result لحساب حاصل ضرب- العددين
3= حساب X*Y ووضع الناتج في متغير result
4- طباعة قيمة المتغير result

تدريب 1 : أكتب الخطوات اللازمة لتنفيذ برنامج يسمح بإدخال درجة الطالب ثم يحدد ما إذا كانت درجة الطالب ناجح أو راسب ويطبع رسالة بنتيجة الطالب (إذا كانت الدرجة أمبر من أو تساوي 50 فالطالب ناجح)

.....

.....

.....

تدريب 2 : أكتب الخطوات اللازمة لتنفيذ برنامج يقرأ عدد إذا كان العدد أكبر من صفر يتم طباعة الرسالة " العدد موجب " وإذا كان أصغر من الصفر يطبع الرسالة " العدد سالب "

.....

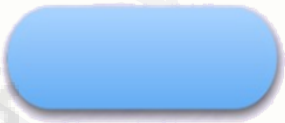
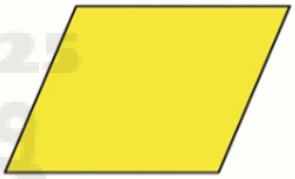
.....

.....

الخرائط التدفقية Flowchart

هي حل رسومي للمشكلة البرمجية . حيث ترتبط مجموعة من الأشكال الهندسية بعضها ببعض في ترتيب منطقي للأحداث والإجراءات البرمجية للحل الخوارزمي .

الأشكال المكونة للخريطة التدفقية

الأشكال	الأسم عربي	الأسم انجليزي	الوصف
	البداية / النهاية	Start /End	يستخدم في بداية ونهاية الخريطة التدفقية
	مدخلات / مخرجات	Input / Output	يستخدم عند إدخال المعطيات و/أو عرض المخرجات
	معالجـــة	Process	عمليات حسابية / منطقية ، تعليمة برمجية
	اتخاذ القرار	Decition	عندما يكون هناك اجراء سيتخذ بناء على شرط نتيجته (نعم/لا)
	الاتجاه	Flow arrow	يبين اتجاه الارتباط بين مختلف أشكال الخريطة التدفقية

الخرائط التدفقية Flowchart

❖ ملاحظات عامة:

- يمكن استخدام اللغة العربية أو الانجليزية عند رسم الخريطة التدفقية .
- التأكد من اتجاه الأسهم عند وضعها.

الخريطة يجب أن تبدأ وتنتهي بالشكل : 

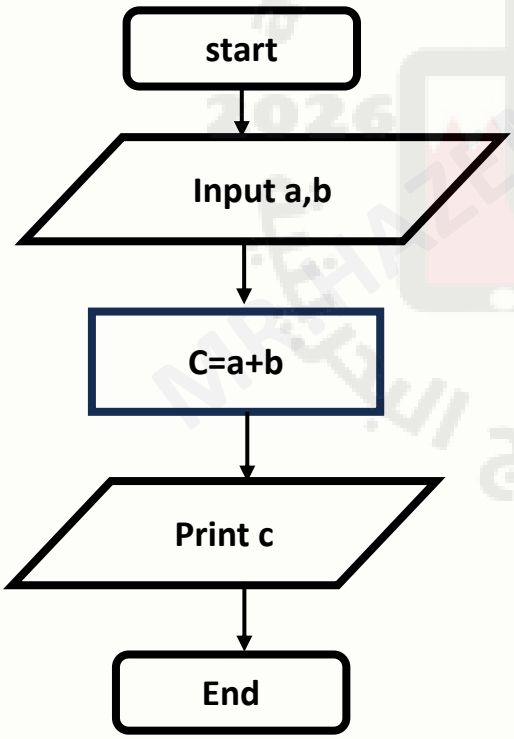
خريطة تدفقية لبرنامج بسيط:

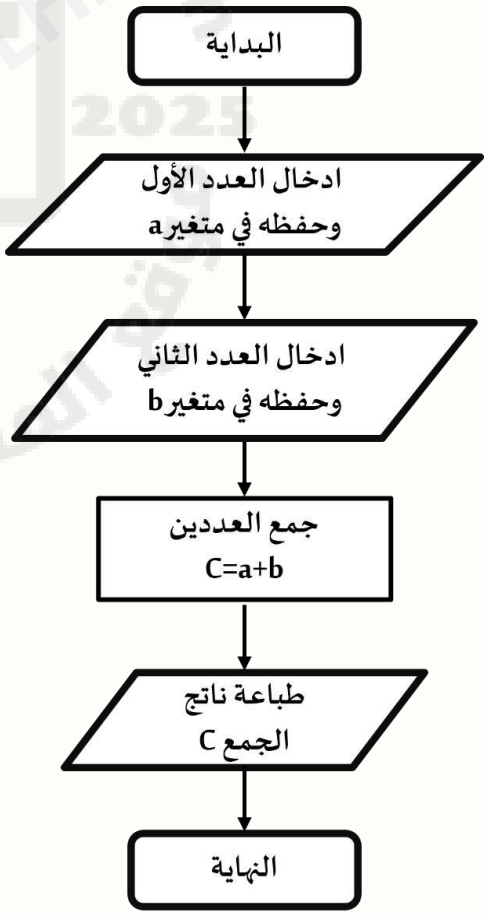
مثال1 : برنامج يستقبل عددين ، يقوم بجمعهما، وعرض ناتج الجمع.

قبل البدء برسم الخريطة التدفقية يجب تحديد:

- **المدخلات : ستكون عددين**
- **المعالجة: جمع العددين وتخزين الناتج في متغير.**
- **المخرجات: طباعة ناتج الجمع.**

بعد تحديد المطلوب من البرنامج، نقوم برسم الأشكال الخاصة بكل خطوة.





مثال2: برنامج لحساب محيط الدائرة، حيث يتم ادخال نصف القطر، استخدم المعادلة التالي:

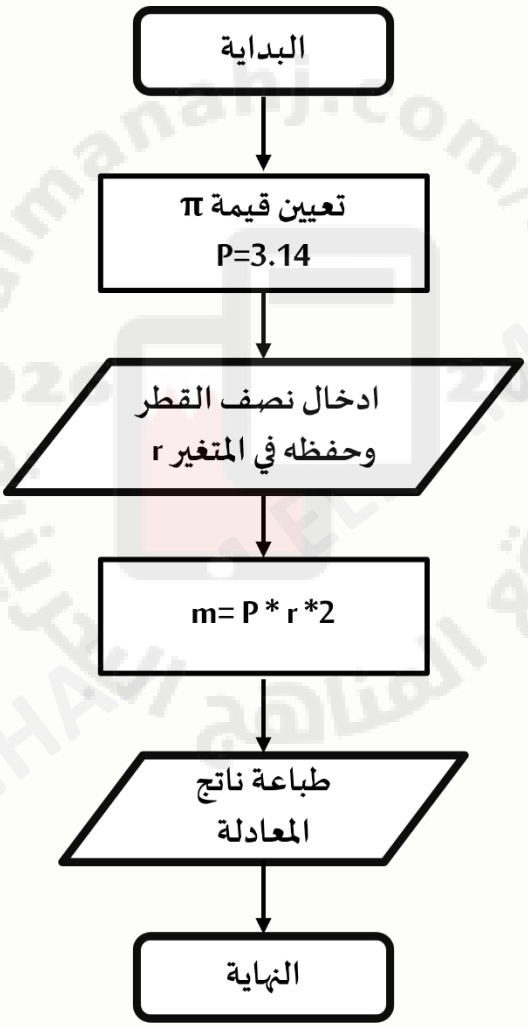
محيط الدائرة= π * نصف القطر * 2

علماً بأن قيمة π = 3.14 ويتم تعيينها كقيمة ثابتة في البرنامج

قبل البدء برسم الخريطة التدفقية يجب تحديد:

- المدخلات : نصف القطر
- المعالجة: π * نصف القطر* 2
- المخرجات: محيط الدائرة.

ملاحظة: تم ذكر بأن هناك قيمة ثابتة سيتم تعيينها في البرنامج، ولتعيين أي قيمة يتم استخدام الشكل الخاص بالمعالجة.

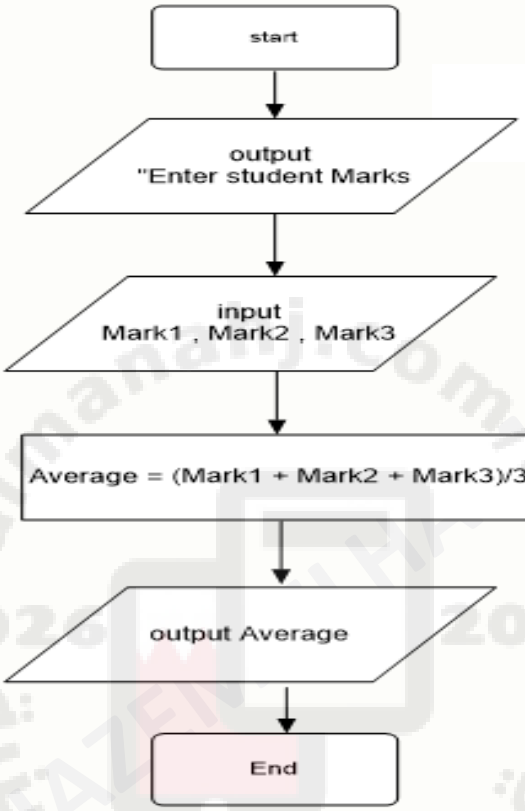


الملاحظات:

- لا يشترط بدء البرنامج بادخال او استقبال البيانات، يمككننا أن نبدأه بالمعالجة تعيين قيمة – معادلة ....

أمثلة :الخرائط التدفقية Flowchart

- طلب منك تصميم برنامج يسمح بإدخال درجات الطالب في ثلاث مواد دراسية مختلفة ثم يقوم بحساب معدل الطالب وطباعته ارسم خريطة تدفقية للبرنامج .



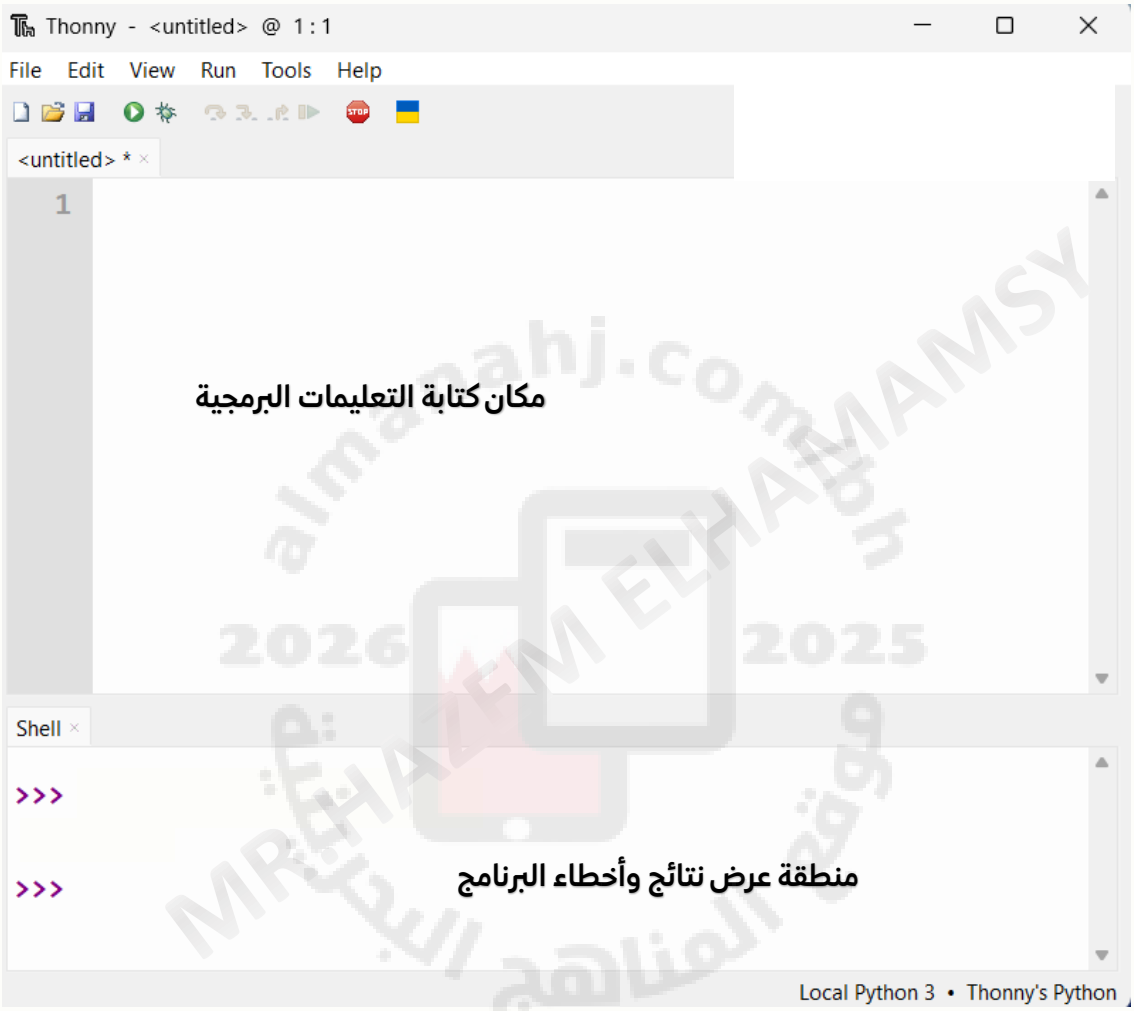
- ارسم خريطة تدفقية لتوصيف برنامج يسمح بإدخال رقمين صحيحين ثم يطبع حاصل ضربهما

- ارسم خريطة تدفقية لقراءة عمري جاسم وحسين ثم كتابة أكبرهم(جاسم = j & حسين = H.)

- ارسم خريطة تدفقية لحساب مساحة المستطيل

❑ التعامل مع واجهة برنامج "Thonny

طريقة تحميل برنامج Thonny



انشاء ملف

↓

File

↓

New

فتح ملف

↓

File

↓

Open

حفظ ملف

↓

File

↓

Save As- File

↓

Save

تشغيل البرنامج

↓

Run

↓

Run Current Script

قواعد كتابة الأسطر البرمجية في لغة بايثون

❑ إضافة تعليق من سطر واحد

ليتم ادراج تعليق في البرنامج ليتمكن المبرمج من كتابة ملاحظاته ليتذكر شيئًا خاصًا بالكود الذي أدخله ويكون الكود واضح بالنسبة له . علمًا بأن البرنامج لاينفذ التعليق .

نستخدم العلامة # لادراج تعليق من سطر واحد

```
1  #تعليق من سطر واحد
```

❑ عند إدخال تعليق على عدة أسطر استخدم الرمز " " " في بداية التعليق ونهايته

```
1  """السطر الأول من التعليق
2  السطر الثاني من التعليق
3  السطر الثالث من التعليق """
```

❑ إدراج سطر جديد

عزيزي الطالب عند استخدام print لطباعة النصوص سنلاحظ انه لايمكننا إدراج سطر جديد بين النصوص باستخدام لوحة المفاتيح والضغط على مفتاح Enter لذا

سوف نستخدم الرمز \n لإدراج السطر في المكان الذي نريده.

مثال : عند كتابة الجملة البرمجيةالتالية وتشغيل البرنامج تكون النتيجة:

```
1  print("welcome to Bahrain")

Shell
>>> %Run -c $EDITOR_CONTENT
 welcome to Bahrain
>>>
```

عندما نريد طباعة Bahrain في سطر منفصل نستخدم الرمز \n كالتالي:

```
1  print("welcome to \n Bahrain")

Shell
>>> %Run -c $EDITOR_CONTENT
 welcome to
  Bahrain
>>>
```

قواعد كتابة الكود في بايثون

❑ طريقة إدراج علامة تنصيص مزدوجة أو مفردة

نستخدم علامة التنصيص المزدوجة أو المفردة في جملة print لطباعة النصوص ، لكن نحتاج في بعض الأحيان بطباعة علامة تنصيص لتظهر مع النص على الشاشة مثل:

```
print("\"مزدوجة تنصيص علامة\"")
print("\' مفردة تنصيص علامة\'")
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
  "علامة تنصيص مزدوجة"
  ' علامة تنصيص مفردة'
>>>
```

❑ طريقة ادراج مسافة tab

لا تقم بإضافة أي مسافة فارغة باستخدام الزر TAB لأن المسافة التي يعطيها هذا الزر غير مسموح إستخدامها في لغة بايثون ، ويتم ادراجها عن طريق استخدام الرمز \t مثال:

```
print("welcom to \t Bahrain")
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
  welcom to       Bahrain
>>>
```

❑ كتابة الإجراء على أكثر من سطر

إذا أردت كتابة أمر واحد على أكثر من سطر قم بوضع الرمز \ (backslash) في نهاية كل سطر وهكذا سيفهم مترجم لغة بايثون أن الأمر يتألف من أكثر من سطر

```
# متغيرات ثلاث بتعريف قمنا هنا
item_1 = 10
item_2 = 20
item_3 = 30
# واحد أمر عن عبارة التالية أسطر الثلاث
# total المتغير في الناتج وضع و item_3 و item_2 و item_1 إذا المتغيرات قيم جمع سيتم هنا
total = item_1 + \
        item_2 + \
        item_3# total المتغير قيمة بعرض قمنا هنا
print("total contains:", total)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 total contains: 60
>>>
```

قواعد كتابة الكود في بايثون

ملاحظة هامة : الأوامر التي تحتوي على الأقواس ( ) أو {} أو [] لا تحتاج لإضافة \

```
seasons = ['Autumn',
           'Winter',
           'Spring',
           'Summer']
print("Seasons contains:", seasons)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
  Seasons contains: ['Autumn', 'Winter', 'Spring', 'Summer']
>>>
```

- ✓ إستخدم 4 مسافات فارغة Space عند وضع الكود بشكل متداخل.
- ✓ ضع سطر فارغ على الأقل بين السطر الذي تم فيه تعريف الكلاس والدوال المعرفة بداخله.
- ✓ ضع سطر فارغ على الأقل بين كل دالتين.
- ✓ ضع سطر فارغ بين كل إثنين بلوك تضيفهما بداخل الدوال.
- ✓ ضع مسافة فارغة حول جمل التحكم وجمل الشرط.
- ✓ عدد الأحرف القصوى التي يمكن وضعها في كل سطر هو 79 حرف.
- ✓ حساسية حالة الأحرف Case Sensitivity أي أن لغة البايثون تميز حالة الأحرف فمثلاً تعيين المتغير A والمتغير a لايمثل نفس المتغير فهما مختلفان.
- ✓ أسماء المتغيرات إذا كان المتغير مكوناً من أكثر من كلمة استخدم الرمز (_) للفصل بين الكلمات first_name.
- ✓ كتابة أكثر من تعيين في نفس السطر نفصل بينهما بالنقطة الفاصلة (;) a=1 ; b=2 ; c=3
- ✓ طريقة أخرى لإسناد قيم لعدة متغيرات في نفس السطر a , b , c = 10 , 20, 30

أنواع البيانات

البيانات:

- لغات البرمجة تخزن البيانات في ذاكرة الكمبيوتر في شكل مخازن (متغيرات) لتستطيع التعامل معها نوع المتغير يكون بحسب نوع البيانات المسندة اليه.

النوع بلغة البايثون	الوصف باللغة الانجليزية	الوصف باللغة العربية	مثال
Str	String	حرف / حروف	Name= “Hazem” , “25” , “H”
Int	Integer	عدد صحيح	Num = 9 , 0 , -75
Float	Float	عدد عشري	Salary = 345.650 , - 254.25
bool	Boolean	قيمة منطقية (True || False)	Male = True 20 > 6
Date	date	تاريخ	1/9/2024
List	list	قائمة	[.....,.....,.....]

أمثلة

```
1 print("my name is hazem")
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
  my name is hazem
>>>
```

```
1 print(8*5)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
  40
>>>
```

```
1 print(8*5.5)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
  44.0
>>>
```

```
1 x=8
2 y=5
3 print(x>y)
```

```
Shell
>>> %Run welcome.py
  True
>>>
```

أنواع البيانات

أنشطة:

❑ حدد نوع كل من البيانات الآتية:

	Int	Str	Bool	float
X = 15				
Y = 36.5				
Z = 8 > 4				
City = "Bahrain"				

❑ ماهو نوع البيانات المناسب لإدخال:-

1- عدد طلاب الصف : .....

2- عنوان البريد الإلكتروني: .....

3- تخزين سعر منتج:.....

❑ ضع علامة ✓ في المكان المناسب للعبارات التالية :

م	العبـــــــارة	صح	خطأ
1	يمكن للعدد الصحيح أن يكون سالباً		
2	القيمة المنطقية للمعادلة 4>8 هي True		
3	تعتبر المسافة space بين الكلمات من النوع نصي		
4	القيمة المنطقية للمعادلة 4=>5 هي True		

37780935

المتغيرات والثوابت

#المتغيرات : عبارة عن مواقع في الذاكرة تخزن البيانات بشكل مؤقت ويمكن تغيير قيمتها أثناء تشغيل البرنامج، في بايثون, المبرمج غير مسؤول عن تحديد أنواع المتغيرات التي يعرّفها في برنامجه فعلياً, عندما تقوم بتعريف متغير وتضع فيه أي قيمة, سيقوم مفسّر لغة بايثون بتحديد نوع هذا المتغير بناءاً على القيمة التي أسندتها إليه بشكل تلقائي وقت التشغيل.

قواعد تسمية المتغيرات والثوابت

يمكننا تسمية المتغير بأسماء مختلفة لكن مع مراعاة بعض الأمور مثل :

- يتكون من حروف وأرقام، ويمكن استخدام (_) اذا كان اسم المتغير من كلمتين .
- الا يجب أن يبدأ برقم .
- الا يحتوي على الرمز @ ، # ، $ ، ! ، % ، &
- يمكن أن يحتوي الإسم على حروف كبيرة وصغيرة، مع الانتباه الى أن أ سماء المتغير حساسة لنوع الأحرف، مثال :المتغيران mycountry و MyCountry
- لانستخدم الكلمات المفتاحية(المحجوزة) للغة البرمجة مثلاً : لايمكن استخدام كلمة print كمتغير لأنها تعتبر كلمة خاصة بلغة البايثون وهذا جدول يبين الكلمات المحجوزة للغة البايثون

if	print	for	break	in	finally	none	not	return
elif	import	nonlocal	raise	class	or	true	try	while
else	continue	except	from	is	with	yield	input	and
assert	del	false	pass	lambda	global	exec	def	

اسناد قيم للمتغيرات :

عملية اسناد أي قيمة للمتغير تكفي باستخدام علامة (=) فلو كان لدي متغير اسمه stName وأردنا اسناد قيمة له سيكون كالتالي:

```
stName = "Hazem"
```

لماذا كتبنا القيمة بين علامتي التنصيص " " ؟

نحن نستخدم علامات التنصيص المزدوجة " " لتحديد قيمة السلسلة النصية في لغة البرمجة بايثون.

السلسلة النصية هي مجموعة من الحروف أو الأرقام أو الرموز التي تمثل نوعًا من البيانات. عندما نكتب stName = "Hazem"، نقول للمترجم أن stName هو متغير يحمل قيمة السلسلة "Hazem" ، وهي اسم شخص. علامات التنصيص تميز السلسلة عن باقي الكود وتمنع حدوث أخطاء في التفسير.

بعض الأمثلة على تسمية المتغيرات الصحيحة في بايثون هي:

المتغير	تسميته الصحيحة لماذا؟
name = “Ali”	متغير نصي يحمل اسم شخص
age = 25	متغير رقمي يحمل عمر شخص
is_married = False	متغير منطقي يحمل قيمة صح أو خطأ
score = 95	متغير رقمي يحمل درجة اختبار
num_of_students = 30	متغير رقمي يحمل عدد الطلاب في فصل
colors = [“red”, “green”, “blue”]	متغير قائمة يحمل ألوان مختلفة

بعض الأمثلة على تسمية المتغيرات غير الصحيحة في بايثون هي:

المتغير	تسميته خطأ لماذا؟
2St_name = “Ali”	يبدأ برقم
$price = 100	يحتوي على رمز
First name = “Ali”	يحتوي على مسافة
print = “Hello”	يستخدم كلمة محجوزة

الثوابت constant

الثابت في لغة البايثون هو عبارة عن متغير لا يمكن تغيير قيمتة اثناء البرنامج.

ملاحظة : على عكس لغات البرمجة الأخرى، لا تحتوي لغة بايثون على أي ثوابت.

أمّا بالنسبة لتعاملنا مع الثوابت سنعتمد على المتغيّرات بحروف انجليزية كبيرة كثوابت بحيث لا تقوم بتغيير قيمها أثناء تنفيذ البرنامج. هذه الطريقة هي طريقة متّفق عليها وليست قاعدة من قواعد لغة البايثون

مثال على الثوابت في بايثون:

```
PI = 3.14 # ثابت رقمي
NAME = “Ahmed” # ثابت نصي
COLORS = [“red”, “green”, “blue”] # ثابت قائمة
```

العوامل الحسابية والمنطقية وعوامل المقارنة

❑ العوامل الحسابية

رمز العامل الحسابي	الوصف	مثال
+	للجمع في حالة البيانات الرقمية	X=5 , y=2 (x+y)
+	للربط في حالة البيانات النصية	X="Ali" , y="omar" (x+y)
-	الطرح	x-y 5-2
*	الضرب	x*y 5*2
*	في حال ضرب نص مع رقم سيتم تكرار النص بعدد مرات الرقم	"Bahrain"*3 Bahrain Bahrain Bahrain
/	القسمة	x/y 5/2
//	للحصول على الناتج الصحيح من ناتج القسمة	x//y 5//2 =2
//	في حال كان أحد الرقمين عدد سالب تكون النتيجة العدد الصحيح الأقل	-13//3=-4.333 Result=-5
%	للحصول على باقي القسمة	X%y 5%2=1
**	للحصول على قوة العدد	X**y 5**2=25

لاتنسى قواعد ترتيب العمليات الحسابية :

الأقواس( ) – قوة العدد **(الأس) – الضرب والقسمة (*و/) – الجمع والطرح (+و-)

❑ عوامل المقارنة

رمز عامل المقارنة	الوصف	مثال
==	يساوي	X==y
!=	لايساوي	X!=y
<	أصغر من	X<y
<=	أصغر من أو يساوي	X<=y
>	أكبر من	X>y
>=	أكبر من أو يساوي	X>=y

لاحظ عزيزي الطالب :

- ✓ يمكننا المقارنة بين (متغير وقيمة & متغير ومتغير)
- ✓ يمكننا استخدام معاملات المقارنة للقيم النصية أيضاً
- ✓ يمكننا المقارنة بين عدد وعدد أو نص ونص ولايمكننا أن نقارن بين عدد ونص
- ✓ قراءة المعاملات من اليسار الى اليمين .. مثال (x>y) تقرأ x أكبر من y

العوامل الحسابية والمنطقية وعوامل المقارنة

❑ العوامل المنطقية

الرمز	المعنى	الوصف	مثال	النتيجة
and	و	عند تحقق الشرطين معاً في نفس الوقت ستكون النتيجة true، عدم تحقق شرط واحد يعطي نتيجة false.	5 > 2 and 5 < 10	True
			6 > 10 and 5 < 10	False
			5 > 8 and 10 < 1	False
or	أو	عند تحقق شرط واحد ستكون النتيجة true، عدم تحقق أي شرط ستكون النتيجة false.	5 > 2 or 5 < 10	True
			6 > 10 or 5 < 10	True
			5 > 8 or 10 < 1	False
not	لا	تستخدم لنفي شرطٍ ما	not(3 > 1)	False
			not(3 < 1)	True

أنشطة على العوامل الحسابية والمنطقية والمقارنة

اوجد نتيجة العمليات الحسابية التالية:

```
X=52
Y=5
R=x / /y
Print(R)
```
الناتج =.....

```
X=52
Y=5
R=x / y
Print(R)
```
الناتج =.....

```
X=4
Y=2
R=x**y
Print(R)
```
الناتج =.....

```
X=52
Y=5
R=x ٪ y
Print(R)
```
الناتج =.....

اوجد نتيجة عمليات المقارنة التالية:

```
X=52
Y=5
R=x != y
Print(R)
```
الناتج =.....

```
X=52
Y=5
R=x == y
Print(R)
```
الناتج =.....

37780935

أنشطة على العوامل الحسابية والمنطقية والمقارنة

اوجد نتيجة عمليات المقارنة التالية:

```
x=5
y=2
print(x==5)
```
الناتج=.....

```
x=int(input("age="))
print(x>15)
```
الناتج =.....

```
X=4
Y=2
R=y<=x
Print(R)
```
الناتج =.....

```
x = 10
y = 5
z = x > y
print(z or False)
print(z and False)
print(not z)
```
الناتج :.....

اوجد نتيجة العمليات المنطقية التالية:

```
print(6==3 or 7<=2)
```
الناتج =.....

```
print(6>3 and 7>2)
```
الناتج =.....

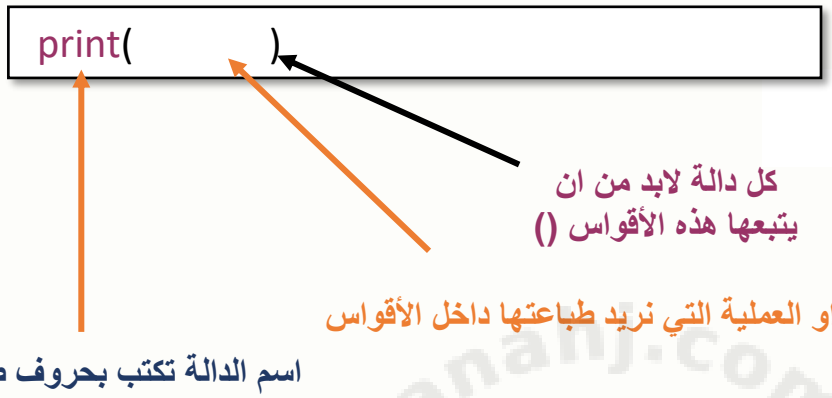
x=7 B= 150 c="welcome" N=-5 Y=20

جملة المقارنة	النتيجة
Print(c=="welcome")	
Print(y>20)	
Print(y!=N)	
Print(B>150)	
Print(N<=0)	
Print(x>N)	

جملة الإدخال والإخراج

دالة عرض / طباعة المعلومات ()print

نستخدم هذه الجملة لعرض البيانات قبل / بعد معالجتها أو عرض نتيجة أي معادلة أو متغيرات وتكون صيغتها كالتالي :



مثال طباعة رسالة ترحيبية: "Welcome to Bahrain"

```
1  print("welcome to Bahrain")
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
  welcome to Bahrain
>>>
```

مثال عرض نتيجة معادلة حسابية :

```
1  print(6*7)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
  42
>>>
```

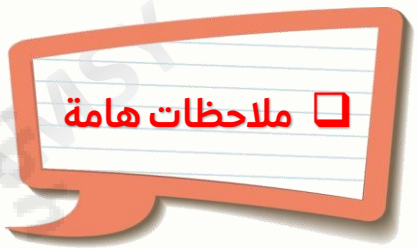
جملة الإدخال والإخراج

دالة عرض / طباعة المعلومات ()print

مثال طباعة نص متبوعاً بمعادلة أو دالة أو متغير :

```
1  print('the result =',6*7)
```

```
Shell ×
>>> %Run -c $EDITOR_CONTENT
  the result = 42
>>>
```



- عند استخدام الجملة لعرض قيمة نصية يجب أن تكون القيمة النصية بين علامتي تنصيص مزدوجة " " أو علامتي تنصيص مفردة ' '
- عند استخدام الجملة لعرض معادلة / رقم / متغير لانستخدم علامتي التنصيص
- عند طباعة أكثر من عنصر باستخدام جملة الطباعة يجب أن نفصل بين كل عنصر والعنصر الأخر بالفاصلة (,)
- عند طباعة علامة التنصيص ضمن دالة الطباعة ()print لحصر النص المراد طباعته نستخدم الرمز \" في المكان المطلوب
- عند إضافة مسافة Tab مباشرة ضمن دالة الطباعة نستخدم الرمز \t في المكان المطلوب

```
print( "welcome ali"  )
```

```
print( 5+2 )
```

```
print( "5"+"2" )
```

```
print( "5"+ 2 )
```

```
>>>      المخرجات(output)
welcome ali
7
52
  print("5"+2)
TypeError: can only concatenate str (not "int") to str
>>>
```

جملة الإدخال والإخراج

دالة إدخال البيانات input()

لا يوجد برنامج يعمل بشكل كامل دون أن يستقبل مدخلات من المستخدم حتى تتم معالجتها وعرضها. وتنفيذ هذه الجملة يجعل الحاسوب ينتظرادخال قيمة معينة من المستخدم،وتكون صيغتها:

```
input("message:"        )
```

اسم الدالة تكتب بحروف صغيرة

كل دالة لابد من ان يتبعها هذه الأقواس ()

رسالة عبارة عن سؤال يدل على المطلوب إدخاله من المستخدم

مثلاً لو طلبنا من المستخدم إدخال إسمه سوف نستخدم الأمر

```
input("Enter your name:" )
```

سنرى عند تنفيذ الأمر في البرنامج ظهور الرسالة وبجوارها مؤشر الكتابة | نكتب الأسم وعند الضغط على مفتاح Enter ينتهي البرنامج.

```
1 input("Enter your name:")

Shell
>>> %Run welcome.py
 Enter your name:Hazem Elhamamsy
>>>
```

الأمر السابق يتم عند طلب ادخال اسم المستخدم في نفس الوقت أما لو أردنا استخدام الإسم فيما بعد في خطوات أخرى من البرنامج فعلينا حفظ الإسم في متغير ليتم إستخدامه لاحقاً

```
st_name=input("Enter your name:" )
```

اسم المتغير يليه علامة يساوي

ولعرض القيمةالتي يتم استخدامها سيكون عن طريق استخدام الجملة . print

```
welcome.py
1 st_name=input("Enter your name:")
2 print("your name is:",st_name)

Shell
>>> %Run welcome.py
 Enter your name:Hazem Elhamamsy
 your name is: Hazem Elhamamsy
>>>
```

رسالة لطباعة الإسم ثم فاصلة واسم المتغير

الاسم الذي تم إدخاله بواسطة المستخدم

طباعة الاسم مسبوقاً برسالة

دالة إدخال البيانات input()

مثال: صمم برنامج لإدخال بيانات وحفظها في متغير :.

```
1 stname=input("Enter the student name:")
```

بعد الضغط على زر تشغيل البرنامج يتم تنفيذ الجملة البرمجية وينتظر مني إدخال البيانات

```
1 stname=input("Enter the student name:")

Shell

>>> %Run -c $EDITOR_CONTENT
 Enter the student name:
```

ندخل القيمة Naser Ali كمثال وأضغط على زر Enter فيتم حفظ القيمة في المتغير

```
1 stname=input("Enter the student name:")

Shell

>>> %Run -c $EDITOR_CONTENT
 Enter the student name:Naser Ali
```

يمكن استخدام دالة الطباعة print() لعرض النص الذي أدخل عن طريق دالة الإدخال input() مثال :

```
1 stname=input("Enter the student name:")
2 print(stname)

Shell

    %Run -c $EDITOR_CONTENT
 Enter the student name:Naser Ali
 Naser Ali

>>>
```

مثال: صمم برنامج لإدخال اسم طالب وحفظه في متغير Sname وإدخال الصف وحفظه في متغير Sclass :.
بعد الضغط على زر تشغيل البرنامج يتم تنفيذ الجملة البرمجية وينتظر مني إدخال البيانات

.....

.....

.....

جملة الإدخال والإخراج

- ❑ أكتب برنامجًا بلغة البايثون لحساب مساحة مربع بحيث يدخل المستخدم طول الضلع ويتم حفظه في المتغير side
- ❑ استخدم المعادلة التالية : مساحة المربع = طول الضلع * طول الضلع
- ❑ عرض الرسالة "square area=" متبوعة بالمساحة متبوعة بالرسالة "m2"

.....

.....

.....

- ❑ أكتب برنامجًا بلغة البايثون لحساب قيمة ضريبة محل ملابس بحيث يدخل المستخدم السعر ويتم حفظه في متغير وحفظ نسبة الضريبة في ثابت Z وقيمته 0.2
- ❑ علمًا بأن : السعر النهائي = السعر +(السعر * الضريبة)
- ❑ اعرض "السعر النهائي=" متبوعًا بناتج المعادلة ويتبعه النص "دب"

.....

.....

.....

- ❑ أكتب برنامج بلغة البايثون يقوم بحساب متوسط درجة الاختبار الأول والثاني متبعاً الخطوات التالية :
- ❑ أدخل قيمة عددين عشريين واحفظهما في متغيرين test1 , test2
- ❑ اوجد متوسط درجة الاختبار الأول test1 والاختبار الثاني test2 واحفظ الناتج في متغير avg
- ❑ استخدم المعادلة (avg = (test1 +test2)/2)
- ❑ اطبع العبارة "average=" متبوعاً بالمتغير avg

.....

.....

.....

37780935

الدوال المضمنة

الدوال هي كتلة الجمل البرمجية الجاهزة مسبقًا يتم استدعاؤها عند اللزوم عادة ماترافقها بيانات خاصة تسمى "parameters" لاستخدامها للوصول الى النتيجة المطلوبة . عادة ماتقوم الدالة بارجاع النتيجة الى البرنامج الرئيسي الذي تم استدعاؤها منه.

الدوال المضمنة الخاصة بالأرقام

دوال تحويل القيم النصية الى قيم رقمية

دالة int( ) –float( ) :

يتم استخدام هذه الدوال للتحويل الى عدد صحيح int أو عدد عشري float ويمكن استخدامها بطريقتين:

الطريقة الأولى: تحويل القيمة المدخلة مباشرة الى عدد كالتالي: n1=int(input("Enter number = "))

الطريقة الثانية: تحويل القيمة الى عدد أثناء العملية الحسابية المطلوبة كالتالي:result=int(n1)+int(n2)

مثال: صمم برنامج لحساب ناتج جمع عددين وعرض الناتج على الشاشة. علما بان الأعداد سيتم إدخالها من قبل المستخدم؟ نستخدم المتغير num1 لتخزين العدد الأول ونوعه عدد صحيح int والمتغير num2 لتخزين العدد الثاني ونوعه عدد عشري float والمتغير result لناتج جمع العددين .

```
1 num1 = int(input("أدخل العدد الأول (صحيح عدد): "))
2 num2 = float(input("أدخل العدد الثاني (عشري عدد): "))
3 result = num1 + num2
4 print(result)
```

```
ell ×
>> %Run -c $EDITOR_CONTENT
أدخل العدد الأول (عدد صحيح): 9
أدخل العدد الثاني (عدد عشري): 7.3
16.3
>>
```

تحويل القيم الرقمية الى قيم نصية

دالة str() :

يتم استخدام هذه الدالة للتحويل الى قيمة نصية وهي str() ويمكن استخدامها كالتالي:
تحويل القيمة المدخلة مباشرة الى عدد كالتالي:

```
n1=int(input("Enter number = ")
print (str(n1) )
```

الدالة ()round:

تستخدم دالة ()round للتقريب لأقرب عدد صحيح

```
1 num=5.45
2 print(round(num))
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 5
```

الدالة ()abs:

تستخدم دالة ()abs للحصول على القيمة المطلقة لأي عدد والمقصود بها القيمة الموجبة

```
1 num=-9
2 print(abs(num))
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 9
```

ضع علامة (✓) في الخانة المناسبة حسب نوع المتغير ضمن البرنامج المرفق :

```
nam=input("المستخدم اسم ادخل:\t")
n=float(input("الجسم كتلة ادخل:\t"))
d=int(input("المولية الكتلة ادخل:\t"))
```

float	str	int	
			num
			n
			d

نفذ الأكواد التالية ذهنياً بلغة بايثون ثم دون النتيجة

code	output
x=5.25 Y=int(x) Print(y)	
x="15" Y=float(x) Print (y)	

الدوال المضمنة الخاصة بالأرقام

أكمل البرنامج التالي، علماً بأن الهدف من البرنامج هو :

ادخال عددان أحدهما عشري و الآخر صحيح، جمع القيم المطلقة للعددين ،حفظ الناتج في متغير، و يتم عرض ناتج الجمع بعد التقريب.

```
num1= _________input ("Enter a Number: ")

_______=_______input ("Enter a Number: ")

res=_____________num1 + _____________num2

print(_____________res)
```

نفذ الأكواد التالية ذهنياً بلغة بايثون ثم دون النتيجة

#	Python code	output
1	price="12.23" quant=5 total=float(price) * quant print(round(total))	
3	a=-45 b=14 res=abs(a) + b print(res)	
4	a=-45 b=14 res=abs(a + b) print(res)	
5	w=str(7) print(w*3)	

الدوال المضمنة

دوال التعامل مع القوائم:

في البداية نود أن نعرف ماهي القائمة ؟ تعتبر القائمة أحد أنواع البيانات التي تسمح بتعيين أكثر من قيمة لها ولايشترط أن تكون جميع هذه القيم من نفس النوع ، فقد تحتوي القائمة الواحدة على : قيمة نصية ، عدد صحيح ، عدد عشري

درسنا فيما سبق القائمة list وهي أحد أنواع البيانات والتي تسمح بتخزين أكثر من قيمة في متغير واحد ومن الممكن لهذه القيم أن تكون رقمية أو نصية أو تكون متنوعة (رقمية ، نصية ، .....)

مثال لقائمة تحتوي على قيم رقمية : num=[71,95,15,23,-9]

مثال لقائمة تحتوي على قيم نصية : city=["Manama","Cairo","Dubai"]

مثال لقائمة تحتوي على قيم متنوعة (نصية – رقمية .... الخ)Z=["Manama",24.5,"Cairo",26]

ملاحظة مهمة جداً :القيم النصية يجب أن توضع بين علامتي تنصيص مزدوجة " " أو تنصيص فردية ' '

وهنا سنلاحظ عندما نريد طباعة قائمة بأكملها يتم طباعتها مع الأقواس الخاصة بالقائمة

```
1 city=["Manama","Cairo","Dubai"]
2 print(city)
```

```
ell

>> %Run -c $EDITOR_CONTENT
['Manama', 'Cairo', 'Dubai']
>>
```

وعندما نريد طباعة قيمة من قيم القائمة نجد مثلاً أن القائمة تحتوي على 5 عناصر وكل عنصر من هذه العناصر له موقع (index) ودائمًا يبدأ الموقع بـ 0

num=[71,95,15,23,-9]

index = موقع القيمة 0 1 2 3 4

فمثلاً لطباعة العناصر حسب موقعها من القائمة num هناك طريقتان:

الطريقة الأولى

```
1 num=["manama","cairo","dubai"]
2 print(num[0])
3 print(num[1])
4 print(num[2])
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 manama
 cairo
 dubai
>>>
```

الطريقة الثانية

```
1 num=["manama","cairo","dubai"]
2 print(num[0],num[1],num[2])
3
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 manama cairo dubai
>>>
```

أما إذا أردنا طباعة عناصر القائمة من الاتجاه العكسي (من اليمين إلى اليسار) والتي لانعرف عدد عناصرها يمكننا استخدام الأرقام السالبة ونبدأ من الرقم -1

num=[71,95,15,23,-9]

index = موقع القيمة 0 4- -3 -2 -1

```
1 num=["manama","cairo","dubai"]
2 print(num[-1])
3
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 dubai
>>>
```

الدوال المضمنة

في بعض الأوقات نحتاج لتغيير قيمة عنصر ما وذلك عن طريق اسناد القيمة الجديدة الى موقع العنصر

❑ تعديل قيمة عنصر من القائمة

```
num=[71,95,15,23,-9]
num[2]=25
print(num)
```

```
>> %Run -c $EDITOR_CONTENT
[71, 95, 25, 23, -9]
>>
```

وهنا سنلاحظ أن بعد طباعة القائمة قد تغير العنصر الثالث من 15 إلى 25

❑ دمج قائمتان أو أكثر :

يمكن أن ندمج قائمتان أو أكثر معًا وطباعة الناتج :

```
x=[1,2,3]
y=[4,5,6]
print(x+y)
```

```
>> %Run -c $EDITOR_CONTENT
[1, 2, 3, 4, 5, 6]
>>
```

❑ أو دمج القائمتين في قائمة واحدة

```
x=[1,2,3]
y=[4,5,6]
z=x+y
print(z)
```

```
>>> %Run -c $EDITOR_CONTENT
[1, 2, 3, 4, 5, 6]
>>>
```

❑ استخدام عناصر القائمة في عملية حسابية

يمكن أن أوظف عناصر القائمة في عمليات حسابية أو منطقية

```
x=[1,2,3]
z=x[1]+3
print(z)
```

```
>>> %Run -c $EDITOR_CONTENT
5
>>>
```

❑ عرض جزء معين من النص

اذا اردنا طباعة الحرف الأول من أي نص نستطيع استخدام طريقة الموقع كما في عناصر القائمة فمثلاً

```
city="Muharraq"
print(city[0])
```

```
>>> %Run -c $EDITOR_CONTENT
M
>>>
```

دوال التعامل مع القوائم:

الدالة append():

تستخدم الدالة لإدراج عنصر جديد في نهاية القائمة

```
city=["Manama","Muharraq","isa town"]
city.append("hidd")
print(city)
```

```
>>> %Run -c $EDITOR_CONTENT
['Manama', 'Muharraq', 'isa town', 'hidd']
>>>
```

لإدراج عنصر مخزن في متغير

```
city=["Manama","Muharraq","isa town"]
name="snabis"
city.append(name)
print(city)
```

```
>>> %Run -c $EDITOR_CONTENT
['Manama', 'Muharraq', 'isa town', 'snabis']
>>>
```

الدالة index():

نتستخدم هذه الدالة fruits=["banana","orange","lemon","apple","mango"]

للحصول على رقم موقع عنصر معين ، سيتم البحث عن موقع العنصر apple من القائمة fruits

```
fruits=["banana","orange","lemon","apple","mango"]
indx=fruits.index("apple")
print(indx)
```

```
>>> %Run -c $EDITOR_CONTENT
3
>>>
```

لقد استخدمنا في المثال السابق متغير indx ومن الممكن الطباعة مباشرة بدون استخدام المتغير كالتالي:

print(fruits.index("apple"))

الدالة pop():

تستخدم دالة pop() لحذف عنصر معين بحسب موقعه ، مثال لحذف العنصر الثالث من القائمة fruits (العنصر الثالث في القائمة رقمه 2 سيتم حذف العنصر apple ويتم طباعة القائمة بعد الحذف.

```
fruits=["banana","orange","lemon","apple","mango"]
fruits.pop(3)
print(fruits)
```

```
>>> %Run -c $EDITOR_CONTENT
['banana', 'orange', 'lemon', 'mango']
>>>
```

دوال التعامل مع القوائم:

الدالة ()remove:

تستخدم دالة ()remove لحذف عنصر معين بحسب قيمته ، فعندما نريد حذف العنصر lemon من القائمة fruits

```
fruits=["banana","orange","lemon","apple","mango"]
fruits.remove("lemon")
print(fruits)
```

```
>>> %Run -c $EDITOR_CONTENT
['banana', 'orange', 'apple', 'mango']
>>>
```

دوال الدالة ()split:

تستخدم لتقسيم النص الى كلمات وحفظها في متغير من النوع قائمة (المسافة هي الفاصل بين كل كلمة وأخرى

```
text="My contry is Bahrain"
new=text.split()
print(new)
```

```
>>> %Run -c $EDITOR_CONTENT
['My', 'contry', 'is', 'Bahrain']
>>>
```

الدالة ()sorted:

تستخدم هذه الدالة لترتيب عناصر القائمة تصاعديًا

```
num=[71,95,15,23,-9]
x=sorted(num)
print(x)
```

```
>>> %Run -c $EDITOR_CONTENT
[-9, 15, 23, 71, 95]
```

نلاحظ في المثال السابق انه تم ترتيب العناصر من الأصغر الى الأكبر..فهل يمكن ترتيب العناصر تنازليًا؟

نعم يمكننا ذلك ولكن سيتم إضافة تغيير على الدالة sorted وهي:

```
num=[71,95,15,23,-9]
x=sorted(num,reverse=True)
print(x)
```

```
>>> %Run -c $EDITOR_CONTENT
[95, 71, 23, 15, -9]
```

ملاحظة مهمة جداً :في الجملة reverse = True يجب أن تبدأ كلمة True بحرف كبير ، لأن كتابة true تعتبر خطأ برمجي

```
num=[71,95,15,23,-9]
x=sorted(num,reverse=true)
print(x)
```

```
>>> %Run -c $EDITOR_CONTENT
Traceback (most recent call last):
  File "<string>", line 2, in <module>
NameError: name 'true' is not defined
>>>
```

ملاحظة مهمة جداً :بعد التشغيل تظهر رسالة خطأ بأن المتغير num غير موجود

الدوال الإحصائية

الدالة sum():

تستخدم هذه لجمع الاعداد الموجودة في القائمة (فقط القائمة التي تحتوي أرقام)

```
1 x=[2,24,37,7,19]
2 s=sum(x)
3 print(s)
```

```
Shell
>>> %Run -c $EDITOR_CON
 89
>>>
```

الدالة max():

تستخدم هذه الدالة لايجاد أكبر قيمة في القائمة

```
1 x=["d","h","m"]
2 s=max(x)
3 print(s)
```

```
Shell
>>> %Run -c $EDITOR_
 m
```

```
1 x=[2,24,37,7,19]
2 s=max(x)
3 print(s)
```

```
Shell
>>> %Run -c $EDITOR_CC
 37
>>>
```

الدالة Min():

تستخدم هذه الدالة لايجاد أصغر قيمة في القائمة

```
1 x=["d","h","m"]
2 s=min(x)
3 print(s)
```

```
Shell
>>> %Run -c $EDITOR_C
 d
```

```
1 x=[2,24,37,7,19]
2 s=min(x)
3 print(s)
```

```
Shell
>>> %Run -c $EDITOR_CC
 2
```

تستخدم هذه الدوال مع القوائم والنصوص والأرقام ويجب أن تحتوي على عناصر من نوع واحد (أرقام فقط أو نصوص فقط)

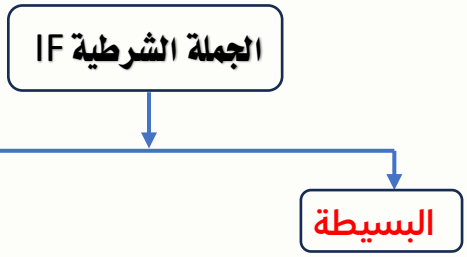
الدالة Len():

تستخدم هذه الدالة لايجاد عدد العناصر الموجودة في القائمة أو حجمها

```
1 x=[2,24,37,7,19]
2 s=len(x)
3 print(s)
```

```
Shell
>>> %Run -c $EDITOR_C
 5
```

#	Python code	output
1	price="12.23" quant=5 total=float(price) * quant print(round(total))	
2	t="*" c="3" print(int(c) * t)	
3	a=-45 b=14 res=abs(a) + b print(res)	
4	a= - 45 b=14 res=abs(a + b) print(res)	
5	t=[48,75,10,-5] t.append(11) t.pop(3) print(t)	
6	b=[20,89,7,0,6] ind=b.index(0) print(ind)	
7	side=[78,96,10,178,0,8] side.remove(8) side.pop(0) print(sorted(side))	
8	w=str(7) print(w*3)	



الجملة الشرطية هي جملة برمجية تمكن من تنفيذ أوامر برمجية بناء على تحقق شرط محدد من عدمه

الجملة الشرطية البسيطة: if

نستخدم الجملة الشرطية البسيطة لتنفيذ إجراءات معينة بناءً على تحقق شرط محدد وتكون صيغة الجملة بالشكل التالي:

if condition : الشرط المطلوب

statement 1 الجملة البرمجية

statement 2 الجملة البرمجية

.....

الرجوع إلى بداية السطر يعني نهاية الجملة الشرطية

يجب أن ننهي سطر الجملة الشرطية if دائماً بنقطتين

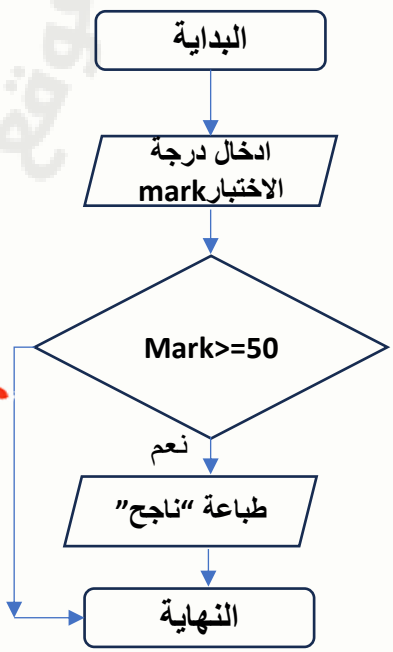
يجب وضع مسافة بادئة لأسطر التعليمات البرمجية بعد if وهي مهمة جداً في بايثون.

مثال:

أكتب برنامج بايثون : يطلب من المستخدم إدخال درجة الاختبار وعرض رسالة " ناجح" إذا كانت درجته أكبرمن أو يساوي 50 ثم ارسم الخريطة التدفقية له

```
1  mark=int(input("Enter your mark:"))
2  if mark>=50:
3      print("ناجح")
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 Enter your mark:55
 ناجح
>>>
```



عند عدم تحديد أي إجراء عند عدم تحقق الشرط يتم رسم خط الى نهاية البرنامج

تذكير:

- في هذا المثال استخدمنا في الشرط عوامل المقارنة أكبر من أو يساوي
- كما استخدمنا الدالة ()int لتحويل البيانات المدخلة الى أرقام لنتمكن من المقارنة بشكل صحيح

الجملة الشرطية IF

مثال2:

أكتب برنامج بايثون : عرض رسالة " ناجح" إذا كانت درجته أكبرمن أو يساوي 50

```
1  mark=45
2  if mark>=50:
3      print("ناجح")
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
```

في هذا المثال لم يظهر ناتج وذلك لأن الشرط لم يتحقق وهو أن الدرجة تكون أكبر من أو تساوي 50 لذا سنحتاج إضافة للشرط وهو ما إذا لم يتحقق الشرط الناتج وللإجابة على هذا السؤال ننتقل الى النوع الثاني من الجمل الشرطية وهو:

الجملة الشرطية الكاملة: if – else

"else " هي جزء من الجمل الشرطية في لغة البايثون، وتستخدم لتحديد تعليمات يتم تنفيذها عندما لا يتحقق الشرط المحدد في جملة IF

الشكل العام لجملة "if-else"، يكون كالتالي:

if condition الشرط المطلوب :
⟷ statement 1 الجملة البرمجية
⟷ statement2 الجملة البرمجية
else :
⟷ statement 1 الجملة البرمجية
⟷ statement2 الجملة البرمجية

- ✓يجب وضع مسافة بادئة لأسطر التعليمات البرمجية بعد if
- ✓الرجوع إلى بداية السطر يعني نهاية الجملة الشرطية
- ✓ولذلك يتم كتابة else في بداية السطر

ويمكن فهم الجملة كالتالي: "إذا كان الشرط صحيحًا، فقم بتنفيذ الجملة التي تلي if، وإلا، فقم بتنفيذ الجملة التي تلي "else"

مثال2:

أكتب برنامج بايثون : يطلب من المستخدم إدخال درجته في الاختبار
عرض رسالة " ناجح" إذا كانت درجته أكبرمن أو يساوي 50
عرض رسالة " راسب" إذا كانت درجته أقل من 50

```
2  if mark>=50:
3      print("ناجح")
4  else:
5      print("راسب")
```

```
Shell
    %Run -c $EDITOR_CONTENT
enter your mark:40
راسب
>>>
```

إرسم الخريطة التدفقية لهذا البرنامج ؟

الجملة التكرارية For

تعتبر الحلقات التكرارية في لغة البايثون أحد أساسيات البرمجة، حيث تسمح للمبرمجين بتكرار تنفيذ مجموعة من الأوامر بشكل متكرر حتى يتحقق شرط معين. وتتيح لنا الحلقات التكرارية في لغة البايثون إنشاء برامج تنفيذية وقابلة للإعادة باستخدام كود قليل جدًا.

حلقة for هي حلقة تكرارية ، وتستخدم لتنفيذ مجموعة من الأوامر على مجموعة من العناصر.لعدد محدد من التكرار

ولدينا (3) ثلاث طرق لتوظيف الجملة التكرارية for

(1)استخدام حلقات التكرار مع الدالة ()range

اسم المتغير وهو يعتبر العداد أو counter ومن الممكن تسميته بأي اسم حسب البرنامج

المجال

مهمة جدا في نهاية الجملة

```
For i in range() :
```

الإجراءات

الإجراءات

- ✓يجب وضع مسافة بادئة لأسطر التعليمات البرمجية بعد For
- ✓الرجوع إلى بداية السطر يعني نهاية الجملة التكرارية

الجملة التكرارية مع range()

- range(stop)
- range(start,stop)
- range(start,stop,step)

Range(5)	Range(5,10)	Range(5,10,2)
يبدأ هنا المجال بعرض خمس أرقام من صفر إلى 4 ❖ نميل إلى بدء العد من الرقم صفر (0) في البرمجة ❖ لانكتب الرقم الأخير (5)	يبدأ هنا المجال بعرض خمس أرقام من 5 إلى 9 ❖ يبدأ العد من الرقم (5) ❖ رقم (10) ليس ضمن المجال	يبدأ العد من الرقم(5)وينتهي ب(9) معامل step سيقوم بتخطي خطوتين في كل دورة ❖ رقم (10) ليس ضمن المجال

الجملة التكرارية For

أمثلة:

```
1 for i in range(6):
2     print(i)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 0
 1
 2
 3
 4
 5
>>>
```

هنا قيمة المعامل stop مساويةً للرقم 6، لذا ستمر حلقة التكرار من بداية المجال 0 إلى نهايته 6 (باستثناء الرقم 6 "نهاية المجال")

```
1 for i in range(5,10):
2     print(i)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 5
 6
 7
 8
 9
>>>
```

هنا البداية الرقم (5) وقيمة المعامل stop مساويةً للرقم 10، لذا ستمر حلقة التكرار من بداية المجال 5 إلى نهايته 10 (باستثناء الرقم 10"نهاية المجال")

```
1 for i in range(5,10,2):
2     print(i)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 5
 7
 9
>>>
```

هنا البداية الرقم (5) وقيمة المعامل stop مساويةً للرقم 10، لذا ستمر حلقة التكرار من بداية المجال 5 إلى نهايته 10 (باستثناء الرقم 10"نهاية المجال") ولكن لدينا هنا step موجبة أي عدد الخطوات وهنا سينتقل خطوتين ليصبح الناتج (5,7,9)

ماذا يحدث لو أن step سالبة ؟؟
قم بكتابة الكود بعد التعديل

الجملة التكرارية مع ( ) range يتم تحديد المجال من قبل المستخدم:

يمكننا أن نقوم بتحديد بداية المجال أونهايته أو حتى الخطوات عن طريق المستخدم،

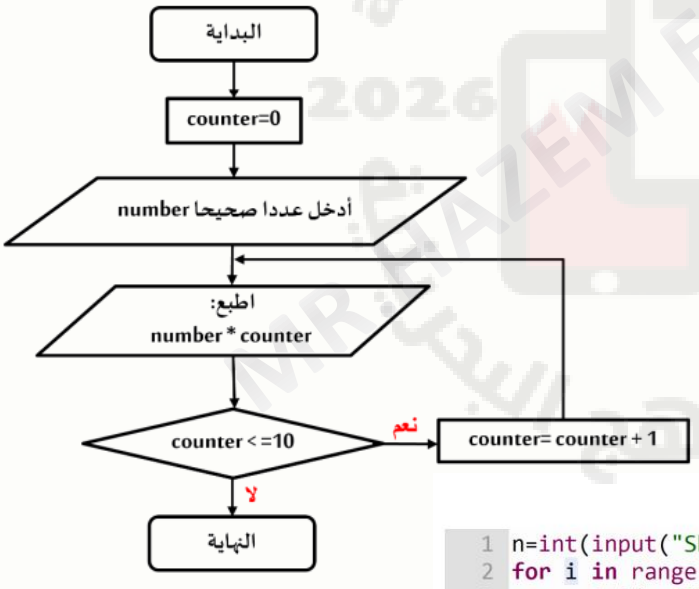
في المثال التالي سنقوم بطباعة الأرقام من صفر الى الرقم الذي يحدده المستخدم

فمثال لو أدخل المستخدم الرقم 5 ، سيتم طباعة الأرقام من ٥ الى 5 ، بالطبع يمكننا أن نقوم باجراء عمليات حسابية أو غيره بنفس الطريقة.

```
1 x=int(input("enter number:"))
2 for i in range(x):
3         print(i)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
enter number:5
0
1
2
3
4
>>>
```

- ✓ **أكتب برنامج بلغة البايثون يعرض لي جدول الضرب لأي عددأدخله للحاسوب .**
- ✓ **أعلم أن جدول الضرب يكون من ٥ إلى 10 أي 11 تكرارا لعملية الضرب.**
- ✓ **اعرض جدول الضرب بحيث يظهر قيمتي الضارب والمضروب.مثال: 35 = 5 * 7**



يكون حلّ المثال أعلاه بلغة Python كالآتي:

```
1 n=int(input("Show the muliple table of the number: "))
2 for i in range(1,11):
3     print(n,"*",i,"=",n * i)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
Show the muliple table of the number: 6
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
6 * 10 = 60
```

(2)استخدام حلقات التكرار مع القائمة list

يمكن أن يكون متغير من نوع قائمة هو المجال في الجملة التكرارية، ونستخدم هذه الطريقة لقراءة محتوى القائمة أو لطباعتها أو لإجراء معالجة بحسب المطلوب

```
1  x=["ahmed","ali","hussen","hazem"]
2  for i in (x):
3          print(i)
```

في هذه المثال نريد طباعة محتوى القائمة (x) وتكون (i) هي نفس العناصر الموجودة بالقائمة

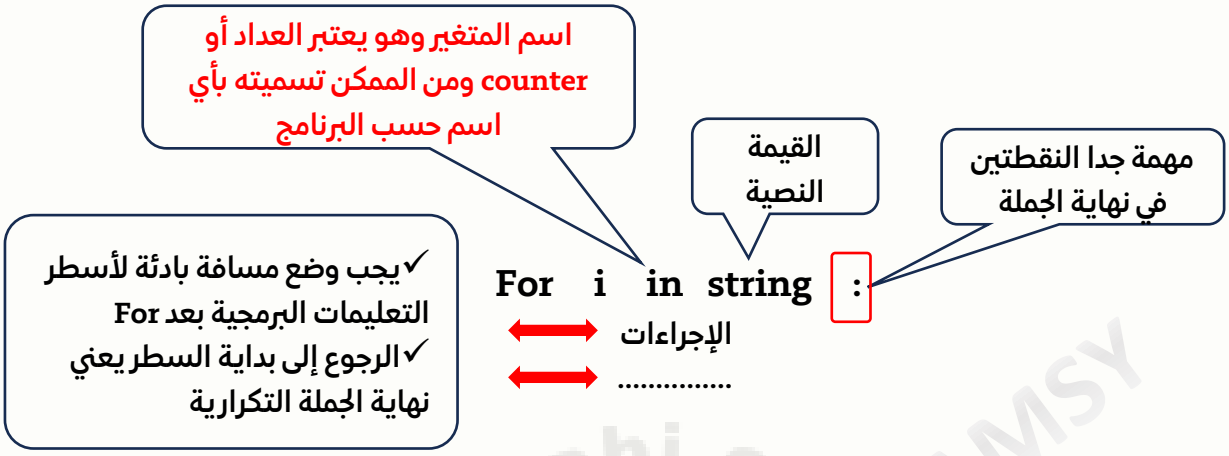
```
Shell
>>> %Run -c $EDITOR_CONTENT
 ahmed
 ali
 hussen
 hazem
>>>
```

ادخال قيم للقائمة:

يمكن ايضا استخدام الجملةالتكرارية لإدخال عناصر في القائمة، فمثلاً في المثال التالي سأطلب من المستخدم ادخال اربعةاسماء،وادخال الإسم مباشرة في القائمة، نلاحظ انه تم استخدام المتغير thelist و اسناد قيمة له قبل البدء بالجملة التكرارية، وذلك حتى يتعرف البرنامج على المتغير ويتمكن من استخدامه.

```
1  thelist=[]
2  for i in range(3):
3      x=input("enter your name:")
4      thelist.append(x)
```

```
Shell
>>> %Run -c $EDITOR_CONTENT
 enter your name:hazem
 enter your name:taha
 enter your name:mahmoud
>>>
```



يمكن استخدام نص ليكون هو المجال الخاص بالجملة التكرارية، فمثال لو كانت الكلمة Hazem هي المجال، واستخدمتها مع الجملة التكرارية فسأتمكن من طباعة كل حرف من الكلمة، أو حتى التعامل معه ضمن إجراءات معينة

```
for i in "hazem":
        print(i)
```

هنا يتم طباعة كل حرف من حروف كلمة hazem

```
Shell
>>> %Run -c $EDITOR_CONTENT
h
a
z
e
m
>>>
```

```
Mark=[95,65,75,45,35]
for i in Mark:
    if i>50:
        print(i)
```

هنا يتم طباعة كل الدرجات الموجودة بالقائمة بشرط أن تكون أكبر من 50

```
Shell
>>> %Run -c $EDITOR_CONTENT
95
65
75
>>>
```


أكتب برنامج بلغة البايثون يمكنني من ضرب العناصر الأصغر من عشرة لقائمة تعينها . وتعرض نتيجته

1- ارسم الخريطة التدفقية 2- حول الحل الى لغة البايثون

أكتب برنامج بلغة البايثون يمكن من عرض الأرقام الزوجية من 0 الى 20

مع العلم أن : إذا كان العدد يقبل القسمة على 2 أي باقي قسمته يساوي صفر فهو عدد زوجي

أسئلة على الحلقة التكرارية for

السؤال الأول:

أكتب برنامجاًيستقبل 5 اعداد صحيحةيتم ادخالها الى قائمة، بعد الإنتهاء اعرض القائمة مرتبةترتيباً تصاعدياً

السؤال الثاني:

أكتب برنامجاًيستقبل رقماً صحيحاً واعرض ناتج جمع الأعداد من 1 إلى الرقم الذي تم إدخاله

السؤال الثالث:

أكتب برنامجاً يطبع الأرقام الفردية من 1 الى 20

السؤال الرابع:

أكتب برنامجاً يقوم باستقبال كلمات لايتجاوز عددها 15 واضافتها إلى قائمة وعند إدخال كلمة stop يتوقف البرنامج ويتم عرض القائمة

أسئلة على الحلقة التكرارية for

السؤال الخامس:

ضع علامة (✓) في الخانة المناسبة حسب نوع المتغير ضمن البرنامج المرفق

```
Num=[12,5,12.5,8,9]
maxN=max(Num)
minN=min(Num)
For I in range(1,5):
        print(I * Num[i])
Res = maxN >=10
Msg="max >=10?"
Print(msg,res)
```

float	bool	list	str	int	
					Num
					maxN
					minN
					I
					Res
					msg

السؤال السادس:

إقرأ البرنامج ونفذ ذهنياً ثم اكتب الناتج في خانة output:

	Python code	Output
1	For i in range(5): print(i)	
2	For i in range(2,10,2): print(i*2)	
3	a=[] For i in a: if i<5: print(i)	
4	txt1=["chem","Eng","arb"] txt2=["102","202","302"] for i in range(3): print(txt1[i] + txt2[i])	

	Python code	Output

5	words=["manama","hid","arad","riffa"]
	for i in words:
	  if i[0]=="m":
	  print(i)	

6	for i in "bahrain": print(i)	

7	words=["moaz","abass","sayed","hussin"]
	for i in words:
	  if len(i) > 3:
	  print(i)	

8	total=1 for i in range(1,5): total=total*i print(total)	
9	newW="" for letter in "Bahrain": newW=letter + newW print(newW)	

10	for b in range(1,5):
	  txt=b * "*"
	  print(txt)	

وإلى لقاء مع الجزء الثاني
التدريبات والإمتحانات

مع أطيب الأمنيات
بالنجاح والتفوق

إعداد / أ. حازم طه الحمامصي
37780935