

تعلم البايثون (بالعربي)

للمبتدئين



By

Hadeel M.Taher

Learn

Python

For

Beginners



مقدمة

بسم الله الرحمن الرحيم

والحمد لله رب العالمين والصلوة والسلام على سيد المرسلين محمد (صلى الله عليه وسلم) اما بعد:

تمت كتابة تعلم البايثون(بالعربي) لمساعدة اي مبتدأ على تعلم برمجة Python بشكل جيد ، حتى وان كنت مبتدئاً بشكل مطلق في البرمجة ، اتمنى ان تجد ما يساعدك على فهم الامور المعقدة بطريقة سلسة وبسيطة فضلا عن فهم البايثون بطريقة مختلفة .

سيكون في هذا الكتاب قاعدة جيدة من المفاهيم التي يمكنك من خلالها استكشاف البايثون. كما سنقدم الكثير من التجارب والشروط والاخفاء التي قد تمر بك في بعض الاحيان ... سيتم الاستفادة من تجارب كثيرة وكورسات عالمية تمت ترجمتها للعربية لسهولة فهم اللغة ...

وفي النهاية انا ايضا سأتعلم معك وسأنقل تجربتي الصغيرة اليك مع عالم البايثون وكلما اتطور في هذه اللغة سأزيد من تطورك ايضا بإذن الله...

إن أحسنت فمن الله، وإن أساءت أو أخطأت

فمن نفسي والشيطان

(صدقه جارية)

Hadeel M.Taher

hadeilmt@gmail.com



Python, what Python?

البايثون ؟ وما هي لغة بايثون ؟

البايثون هي لغة برمجية تستخدعم على نطاق واسع جدا خلال الفترة الحالية. تم إنشاؤها من قبل (Guido van Rossum) في أواخر الثمانينيات من القرن الماضي اما معنى كلمة بايثون فهي تمثل احدى انواع الافاعي تعد لغة البايثون من اللغات القوية التي يمكن قراءة تعليماتها البرمجية بسهولة وبساطة مما يسهل الامر على المبرمجين من تطوير التطبيقات بسرعة . اضافة الى ذلك فان لغة البايثون قابلة للتنفيذ لبعض أشهر أنظمة التشغيل مثل Windows و Mac هذا يسمح لنا بتوزيع برامج Python الخاصة بنا بطريقة مريحة بدون مطالبة المستخدمين بتنزيل مترجم Python.

تعد البايثون لغة مثالية للمبتدئين وذلك لبساطتها حيث انها تتطلب عدد أقل من الأسطر البرمجية لتنفيذ نفس المهمة مقارنة باللغات البرمجية الأخرى.

اذا اردت التعلم هيا بنا نبدأ..



Getting ready for Python: Installing

انت مستعد للبايثون: تثبيت البايثون

ان الطريقة الأمثل لتنزيل البايثون على جهازك من خلال الرابط التالي:

<https://www.python.org/>

يعد الرابط اعلاه الموقع الرسمي لتنزيل احدث نسخة من البايثون والآن 3.8.3 هي النسخة الاحدث حاليا.

The screenshot shows the official Python website at <https://www.python.org/>. The top navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. Below the header, there's a search bar and a "Socialize" button. The main content area features a large Python logo and a "Quick & Easy to Learn" section with text about Python's syntax and a link to an "Overview". On the left, there's a code editor window displaying a Python 3 script with interactive session output. At the bottom, a banner states: "Python is a programming language that lets you work quickly and integrate systems more effectively. [Learn More](#)".



حيث يمكننا اختيار الباليون المناسب على حسب نظامك التشغيلي و المستخدم سواء كان 32-bit او 64-bit فضلا عن ذلك يوجد في الموقع الرسمي تثبيت الباليون لأغلب Platforms كما في الصورة أدناه :

The screenshot shows the Python.org download page. In the top navigation bar, the 'Downloads' tab is selected. On the left sidebar, under the 'Windows' category, the Python 3.5.3 version is highlighted. A callout box provides information about Python 3.5+ not being compatible with Windows XP or earlier, noting that Python can be used on many other operating systems. It also mentions that AIXTOOLS provides binary packages for Python 3 and Python 2.

ولتثبيت باليون لنظام ويندوز ٦٤ فقط نضغط على تنزيل(Download) نختار الـ 64 windows

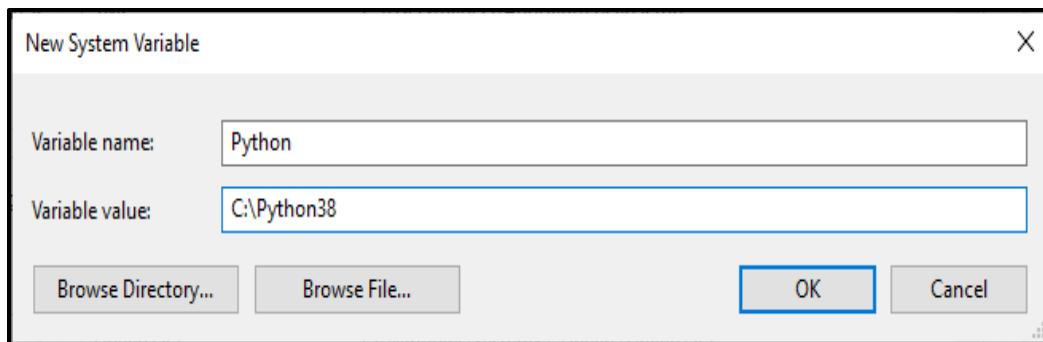
The screenshot shows the Python.org download page displaying a list of files for Python 3.5.3. The table includes columns for Version, Operating System, Description, MD5 Sum, File Size, and GPG. The files listed include various installers and source tarballs for Windows, Mac OS X, and Linux platforms.

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		87c10a2aa59062de75a0ca5204e0e7bd7	24067487	SIG
XZ compressed source tarball	Source release		3000f90aa4413092ae923d2122ea78	17912964	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	dd9e7f84e205d21fb0d40773a7a416a9	30129781	SIG
Windows .msi file	Windows		4aemrbdf7zRldf0d011e7cf0d0a0d9422	8568303	SIG
Windows .whl 64 embeddable zip file	Windows	for AMD64/EM64T/x64	c12fe7f4c12b447241d502ed9fb5d1	8175001	SIG
Windows .exe executable installer	Windows	for AMD64/EM64T/x64	f0258f0d0e7ead10cf9fc2370a42b53b	27005800	SIG
Windows .exe web-based installer	Windows	for AMD64/EM64T/x64	17e0ff0d2fec770f1a79f825095c4	1364136	SIG
Windows .xml embeddable zip file	Windows		8ee09403ec0cc2e89b43b4a4fb3521e	7530315	SIG
Windows .exe executable installer	Windows		e12373e21467c142250eb10f4c231f	26744744	SIG
Windows .exe web-based installer	Windows		1e725a2bbfca3dbed7a814d09ec01df2	1333800	SIG



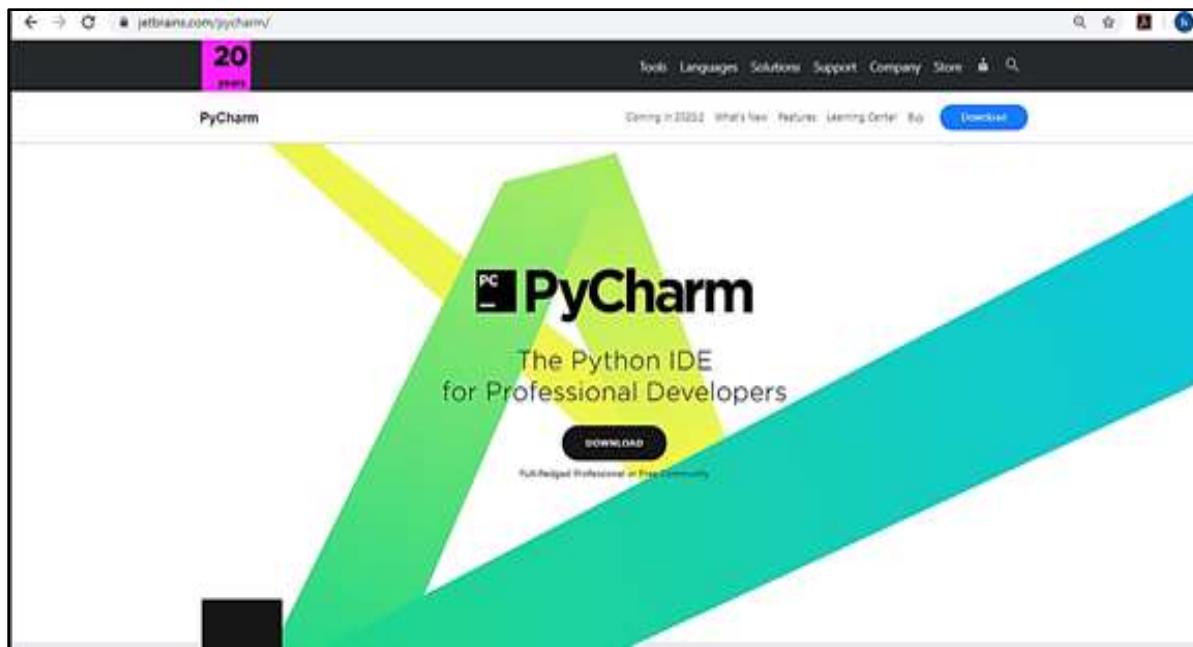
لكن قبل البدء بكتابه البرنامج هناك بعض الملاحظات التي جمعتها من هنا وهناك للاستفادة منها :

- بعد تثبيت البايثون على الكمبيوتر نذهب الى Start ثم نبحث عن IDLE(python3.8) ثم نضغط open والآن الواجهة الموجودة هي الواجهة التي تنفذ البرامج من خلالها
- لحفظ اي ملف Python على جهازك يجب ان يكون بصيغة (py). وهذه الصيغة غير موجودة في نظامك لذلك ستحتاج الى اضافتها بنفسك من خلال بالضغط click يمين على ايقونة جهاز الكمبيوتر ثم اختيار(خصائص) ثم (إعدادات النظام المتقدمة) ثم (متغيرات البيئة) بعدها نذهب الى (متغيرات النظام) ونقر على (جديد) لظهور الواجهة التي يجب ان نملأها كالتالي:



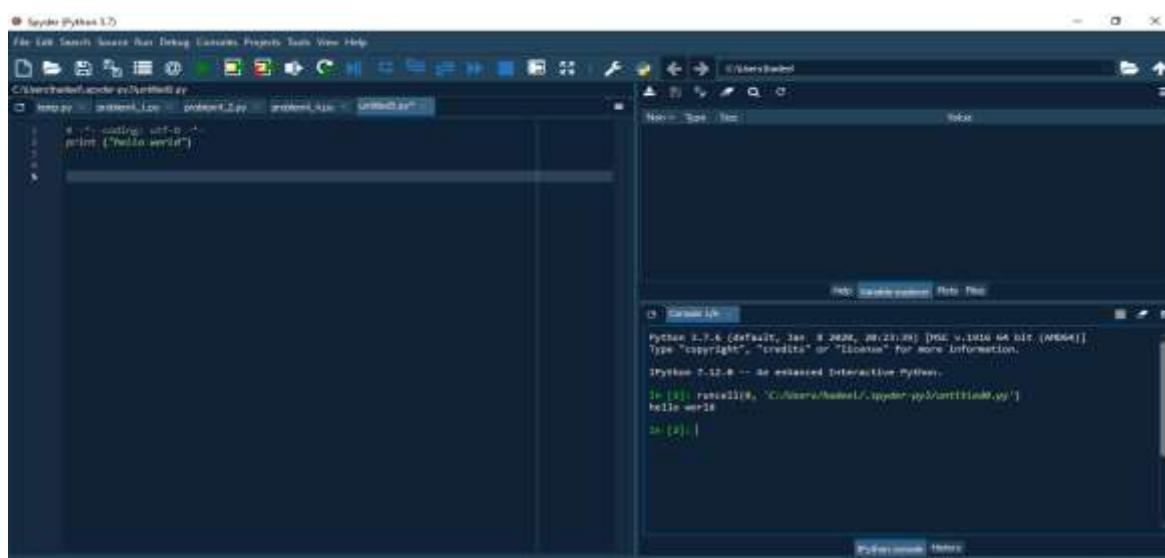
حيث تم كتابة الكلمة Python في اسم المتغير الجديد ومن ثم اضفنا الامتداد (C:\Python38) سنكتب الامتداد (Python 3.8) فاذا كان (Python 3.8) سنتباهي (امتداد) واحيرنا نضغط على OK ثم نكون بذلك اتممنا عملية تهيئة الويندوز لاستخدام البايثون.

- في حال عدم حفظ الملف بالصيغة (py). يمكننا حفظ الملف باستخدام الصيغة (.pyw).
- يمكن ايضا تحميل برنامج pycharm من خلال الرابط التالي:
<https://www.jetbrains.com/pycharm/>



حيث يعد pycharm بيئة تطويرية متكاملة تستخدم في لغات برمجة الحاسوب، خاصة البرمجة بلغة بايثون. حيث يتميز بدعم وتطوير المكتبات الأساسية للبايثون مثل مكتبة Django لذلك انصح ان تجرب ال pycharm لأنها ممتعة بالعمل.

- Spyder هو برنامج شهير تم اعتماده واستخدامه من قبل الجامعات العالمية حيث ان بعض الكورسات التي اطلقتها الجامعات العربية تستخدم هذه البيئية المتكاملة مفتوحة المصدر لتعلم البايثون وذلك لأنها تتميز بضمها مجموعة عديدة من المكتبات اهمها Matplotlib و SciPy و NumPy .





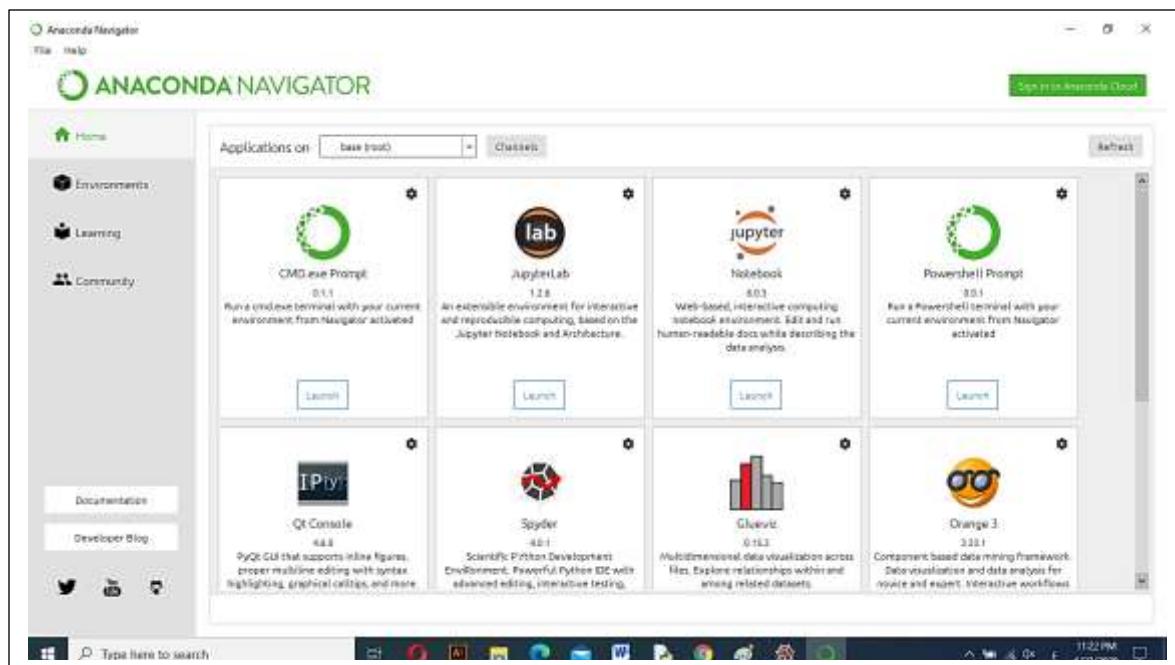
يمكننا ملاحظة وجود ثلاث اقسام للشاشة في الصورة اعلاه :

الجزء الاول : في حال كتابة برنامج (hello world) نلاحظ ان برنامج سبايدر قام بانشاء ملف بصيغة (.py) مباشرة.

الجزء الثاني: يمكنك الحصول على المساعدة وايضا تستطيع التعرف على جميع انواع المتغيرات التي تم كتابتها في البرنامج فضلا عن امكانية استعراض كافة الملفات python الموجودة في جهازك.

الجزء الثالث : هو Console والذي يشابه تماما عمل python shell والصورة اعلاه توضح اليه العمل ببرنامج سبايدر.

- اخيرا وليس اخرا برنامج Anaconda الشهير المكتوب بلغة البايثون يعد الافضل في مجال البيانات اضافته الى ذلك يعد أسهل الطرق لتنصيب المكتبات العلمية في البايثون.
- ادناه واجهة البرنامج:



والآن يمكننا كتابة اول برنامج في البايثون😊

FIRST program Using the Python Shell, IDLE and Writing our



Writing the first program

لنكتب برنامجنا الأول

من قائمة Start نختار Python IDLE ستفتح لنا الواجهة أدناه بعدها تستطيع كتابة برنامج "hello world".

```
*Python 3.8.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hello world")|
```

وكلما شاهد في الصورة أعلاه لقد قمنا بطباعة عبارة "hello world" من خلال الدالة print بعدها بمجرد الضغط على enter ستكون النتيجة كالتالي :

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hello world")
hello world
>>> |
```



مبارك لك لقد تم تنفيذ أول برنامج بنجاح



ملاحظة عامة عن البايثون

- البايثون لا تحتاج الى فارزة منقوطة في نهاية الجملة
- يجب الكتابة بعد >> مباشرة
- البايثون يحسب عدد ال space خلال التنفيذ فيجب ان تكون دقيق باستخدامها خصوصا في if و for
- لأنها قد تسبب لك الأخطاء وسيتم توضيحها لاحقا.
- لغة البايثون تعد لغة مرنة جدا

ولكتابته البرنامج وتنفيذه من خلال فتح File كما في الصورة أدناه:

```

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
New File Ctrl+N :7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (In
Open... Ctrl+O "credits" or "license()" for more information.
Open Module... Alt+M
Recent Files ▶
Module Browser Alt+C
Path Browser
Save Ctrl+S
Save As... Ctrl+Shift+S
Save Copy As... Alt+Shift+S
Print Window Ctrl+P
Close Alt+F4
Exit Ctrl+Q

```

عند فتح الفايل سنكتب بداخله البرنامج بدون اي مقدمات كأقواس او main لأن البايثون لا تحتاج لذلك
نستطيع الطباعة مباشرة :

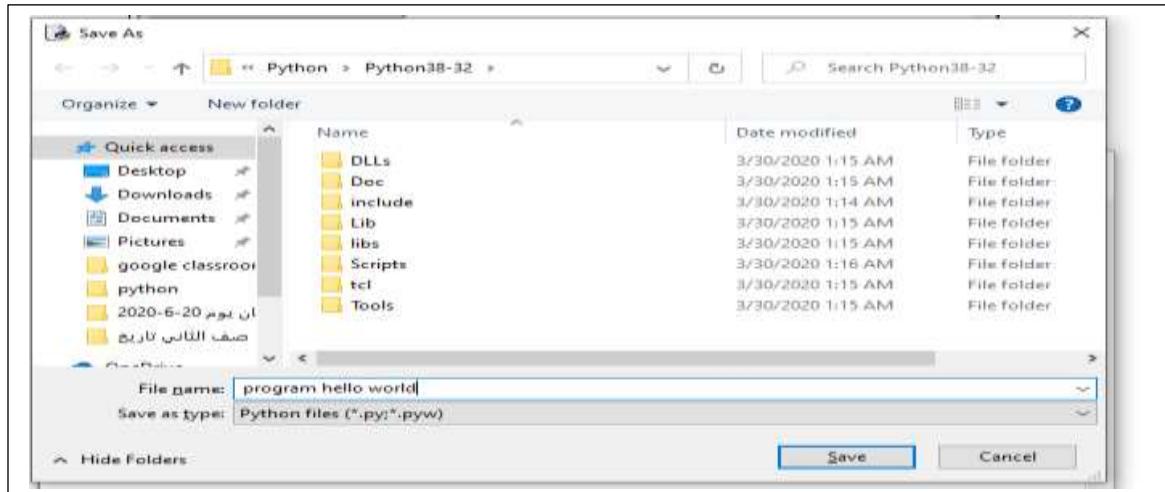
```

*untitled*
File Edit Format Run Options Window Help
print("hello world")

```



اما تنفيذ البرنامج سيكون من خلال الكلمة `run` الموجودة في الاعلى او بالضغط على F5 لكن يجب حفظ الملف `:save as file` ثم `run` له . والصورة أدناه تبين كيفية الحفظ من خلال الضغط على الكلمة `run` قبل ان عمل `run` له .



سيتم حفظ البرنامج بالاسم الذي تريده وبصيغة البايثون (.py)(.pyw).

بعد ان تم حفظ البرنامج سيكون من السهولة فتحه وعمل `run` له والنتيجة ستكون كالتالي:

```

hello world.py - C:/Users/hadeel
Python 3.8.2 Shell
File Edit Format Run Options Window Help
print("hello world")
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world
.py
hello world
>>> |

```

نلاحظ عند التنفيذ ظهرت `python shell` وتم طباعة جملة "hello world" ببساطة



هناك طريقة أخرى لتنفيذ برنامج البايثون من خلال cmd من خلال تتبع خطوات بسيطة:

- حفظ الملف باسم print مثلًا وسيكون الملف بصيغة py. أو pyw. على سطح المكتب



- نضغط على زر Shift و زر الفارة اليمين في أي مكان فارغ على سطح المكتب حتى يظهر لنا عبارة Open power shell windows here



- بعد فتح power shell سنكتب الامر التالي: python print.pyw (البايثون بعدها اسم الملف ثم الامتداد) ستلاحظ انه سيتم تنفيذ البرنامج بيسرا.

```
Windows PowerShell
PS C:\Users\hadeel\Desktop> python print.pyw
hello world
PS C:\Users\hadeel\Desktop>
```



تعد الطريقة اعلاه طريقة فرعية يمكن تجربتها في حال حدوث خلل ما ...

لقد احترنا اول برنامج الان لنبدأ بأساسيات اللغة





Comments in Python

التعليقات في البايثون

تعد التعليقات مهمة جداً في أي لغة برمجية ولإضافة تعليقات إلى برنامج مكتوب بلغة البايثون نكتب علامة # أمام كل سطر من التعليق كالتالي:

The screenshot shows a code editor window with the following content:

```
*hello world.py - C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py (3.8.2)*
File Edit Format Run Options Window Help
print("hello world") # لكتابه التعليق المناسب
```

نلاحظ ان التعليق اصبح باللون الاحمر فهذا يعني انه سيتم تجاهله من قبل البايثون ولكنه سيوضح الخطوات البرمجية .

اما اذا اردنا اضافة # لإخفاء عدة خطوات برمجية يمكن ذلك من خلال الضغط على Alt+3 او بالذهاب الى Format ستجد عبارة Comment Out Region يتم الضغط عليها وسيظهر لنا # في جميع الخطوات المطلوب اخفائها. كما موضح ادناه:

The screenshot shows a code editor window with the Format menu open. The 'Comment Out Region' option is highlighted with a blue selection bar. The menu items are:

- Format Paragraph
- Indent Region
- Detent Region
- Comment Out Region** (highlighted)
- Uncomment Region
- Tabify Region
- Untabify Region
- Toggle Tabs
- New Indent Width
- Strip Trailing Whitespace



```
*hello world.py - C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py (3.8.2)*
File Edit Format Run Options Window Help
print("hello world")
##print("hello world")
##print("hello world")
##print("hello world")
```

طريقة الاخرى لإضافة تعليقات(هذه الطريقة مفيدة جدا) :

في حال لدينا تعليقات متعددة او شروحات لأكثر من سطر يمكننا ايضا استخدام ثلاث علامات اقتباس حيث نطبع جملة hello world وبين علامات الاقتباس كتابة لن تظهر في التنفيذ ولن توقف البرنامج عن التنفيذ كما أدناه :

<pre>File Edit Format Run Options W print("hello world") ''' This is a comment This is also a comment This is yet another comment '''</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (In tel)] on win32 Type "help", "copyright", "credits" or "license()" for more information. >>> = RESTART: C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world .py hello world >>> </pre>
---	--



Python I/O

الادخال والاخراج في البايثون

في لغة البايثون الادخال والاخراج يختلف نوعاً ما عن اللغات الأخرى فمثلاً في C++ نلاحظ أن الادخال عن طريق (cout <>) والاخراج عن طريق (cin <>) لكن هنا في البايثون الأمر يحتاج إلى دالتين فقط هما واحدة للإدخال للطباعة تسمى (print) والإدخال عن طريق (input) يعملاً على أداء مهمة الإدخال / الإخراج في Python. والدالتين تكتبهان بطريقة يسيرة بدون تعقيدات كالتالي:

 tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw	 Python 3.8.2 Shell
<pre> File Edit Format Run Options Window x= input("enter num: ") #لإدخال print(x) #لإخراج </pre>	<pre> File Edit Shell Debug Options Window I Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" >>> ===== RESTART: C:\Users\ enter num: 6 6 >>> </pre>

في المثال أعلاه نلاحظ أن (input) المستخدمة للإدخال من قبل المستخدم هي مرنة للغاية حيث يمكن إدخال رقم . حرف . جملة ... الخ. أما الطباعة (print) تعمل لإخراج البيانات إلى الشاشة.

من جهة أخرى يمكننا أن نحدد الإدخال برقم صحيح أو رقم عشري او جملة بمجرد كتابة int او str او float قبل عبارة input ونمثل ذلك كالتالي:



<pre>tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw</pre> <pre> File Edit Format Run Options Window Help x= int(input("enter num: ")) #لإدخال # print(x) #لإخراج </pre>	<pre>Python 3.8.2 Shell</pre> <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 tel] on win32 Type "help", "copyright", "credits" or "license()" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw ===== enter num: 5.5 Traceback (most recent call last): File "C:\Users\hadeel\Desktop\tryexception.pyw", line 1, in <module> x= int(input("enter num: ")) #لإدخال # ValueError: invalid literal for int() with base 10: '5.5' >>> </pre>
--	--

الخطأ الظاهر في التنفيذ انه تم ادخال float فظهر الخطأ لأن البرنامج لا يقبل سوى رقم int (اي تم تحديد القيمة المدخلة برقم صحيح فقط) فالتنفيذ الصحيح كالتالي :

<pre>tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw</pre> <pre> File Edit Format Run Options Window Help x= int(input("enter num: ")) #لإدخال # print(x) #لإخراج </pre>	<pre>Python 3.8.2 Shell</pre> <pre> File Edit Shell Debug Options Window + Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" >>> ===== RESTART: C:\Users\ enter num: 6 6 >>> </pre>
--	--

<pre>tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw</pre> <pre> File Edit Format Run Options Window Help x= str(input("enter string : ")) #لإدخال # print(x) #لإخراج </pre>	<pre>Python 3.8.2 Shell</pre> <pre> File Edit Shell Debug Options Window + Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" >>> ===== RESTART: C:\Users\ enter string : python python >>> </pre>
--	--



Python Variables

متغيرات البايثون

المتغيرات هي أسماء تُعطى للبيانات التي نحتاج إلى تخزينها ومعالجتها في برمجنا. على عكس لغات البرمجة الأخرى ، ليس لدى Python Shell تعريف المتغير declaring (تم استخدام Python Shell للسهولة والسرعة

(كما سنلاحظ في الأمثلة أدناه:



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> x=0
>>> y="hadeel"
>>> print(x)
0
>>> print(y)
hadeel
>>> |
```

نلاحظ في المثال أعلاه انه تم تعريف متغيرين مختلفين لكن البايثون تعامل معهما بكل احترافية. ولنثبت ذلك سنقوم بكتابة الامر type والذي سيبيّن ان البايثون ميز بين int و str كما نلاحظ أدناه:

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> x=0
>>> y="hadeel"
>>> print(x)
0
>>> print(y)
hadeel
>>> type(x)
<class 'int'>
>>> type(y)
<class 'str'>
>>>
```



يمكن تغيير نوع المتغير في البايثون حتى بعد تعبيئها كالاتي:

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (In
sel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> x=0
>>> y="hadeel"
>>> print(x)
0
>>> print(y)
hadeel
>>> type(x)
<class 'int'>
>>> type(y)
<class 'str'>
>>> x="x is now str"
>>> print(x)
x is now str
>>> type(x)
<class 'str'>
>>> print(y)
hadeel
>>> type(y)
<class 'str'>
>>> |
```

لقد تم تغيير نوع المتغير X من int الى str مع الابقاء على المتغير y كما هو، فضلا عن انه يمكن التعريف عن متغير str إما باستخدام علامات اقتباس مفردة أو مزدوجة بالحالتين ستكون النتيجة صحيحة:

```
>>> x= "hadeel"
>>> print(x)
hadeel
>>> y='hadeel'
>>> print(y)
hadeel
>>> |
```

هناك ميزة في البايثون وهي يمكن تعين قيم لمتغيرات متعددة في سطر واحد:



```
*hello world.py - C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py
dog
cat
bird
>>>
```

نلاحظ ان كلًا من `a,b,c` اخذت القيم `dog ,cat ,bird` بالتتابع من جهة اخرى يمكنك تعين نفس القيمة
لتغييرات متعددة في سطر واحد:

```
*hello world.py - C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py
hadeel
hadeel
hadeel
>>>
```

واخيرا يمكن جمع جملتين(str) معا وذلك بإضافة (+) فقط.

```
*hello world.py - C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py
i love summer i hate winter
>>>
```



إلا أنه يجب الانتباه الى ان علامة (+) تأتي ايضاً operator عند جمع عددين من نوع int او float اضافته الى ذلك لا يمكن جمع int و str سيظهر لنا error حيث انه من الخطأ الجمع بينهما (+) . وكما ادناه مثالين الاول جمعنا عددين int والثاني تم جمع int و str وسنبين ال error :

```

hello world.py - C:/Users/hadeel
Python 3.8.2 Shell
File Edit Format Run Options Window Help
x= 6
y= 4
print(x+y)
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world
.py
10
>>> |

```

```

hello world.py - C:/Users/hadeel
Python 3.8.2 Shell
File Edit Format Run Options Window Help
x= "i love summer"
y=5
print(x+y)
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world
.py
Traceback (most recent call last):
  File "C:/Users/hadeel/AppData/Local/Programs/Python/Python38-32/hello world.py", line 3, in <module>
    print(x+y)
TypeError: can only concatenate str (not "int") to str
>>> |

```

في البايثون يمكن ان يكون المتغير (_ Underscore) ويكون لهذا المتغير ان يحفظ قيمة وكذلك تستخدم (_) لإرجاع آخر قيمة تم تنفيذها في Python Prompt / Interpreter والمثالين ادناه يوضحان العملية:

```

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> _=10
>>> print(_)
10
>>> _+_
20
>>> |

```



The screenshot shows the Python 3.8.2 Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell displays the following code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> x=4
>>> y=6
>>> x+y
10
>>> -
10
>>> _+10
20
>>> _*4
80
>>> _/2
40.0
>>> |
```



والآن

بعد تعلمنا بعض الأساسيات للمتغيرات

سنأتي لأنواع المتغيرات



Data Types in Python

أنواع البيانات في بایثون

تعد البيانات وانواعها في البرمجة بشكل عام البایثون بشكل خاص مفهوماً مهماً جداً . في البایثون تنقسم انواع البيانات الى سبعة انواع اساسية منها ما هو معروف في اللغات البرمجية الالخري كـ (str, int , float) ومن جهة اخرى هناك ثلث انواع بيانات اكثراً تقدماً في البایثون مثل(list, tuple and dictionary) حيث سنقوم بشرح كافة الانواع تباعاً. في الجدول ادناه ملخص لأنواع البيانات في البایثون:

نوع	المتغير	ت
str	(Strings)	١
int, float, complex	(Numbers)	٢
bool	(Booleans)	٣
list, tuple	(Sequences) متسلاسلات او مصفوفات	٤
Sets, frozenset	مصفوفات ليس لها حجم ثابت. ولا يمكن حذف قيمها	٥
dict	Dictionaries القاموس	٦
.bytes, bytearray	Binary الثنائي	٧

والآن لنبدأ بشرحها بتفصيل كل نوع بشكل منفصل :



٤. النصوص (Strings)

يمكن تعريف النصوص في البايثون من خلال (اسم متغير = "النص" او اسم المتغير = 'النص') اي بوضع str ما بين علامة اقتباس واحدة single quotes او علامة اقتباس مزدوجة double quotes او ثلث علامات اقتباس والتي تدعى ب(multiline string) كما سنوضح بالأمثلة التالية :

```
str.py - C:/Users/hadeel/Desktop/str.py (3.8.2)
File Edit Format Run Options Window Help
name ='hadeel'
print("Name: ", name)
```

عند التنفيذ سيكون الناتج :

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/hadeel/Desktop/str.py =====
Name: hadeel
>>> |
```

في حال علامتين اقتباس سيكون كالتالي:



```

str.py - C:/Users/hadeel/Desktop/str.py
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/hadeel/Desktop/str.py ==
hadeel
>>> |

```

واخيرا ان كانت ثلاث علامات اقتباس سيكون البرنامج في البايثون كما ادناه:

```

str.py - C:/Users/hadeel/Desktop/str.py (3.8.2)
Python 3.8.2 Shell
File Edit Format Run Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/hadeel/Desktop/str.py ==
Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
>>> |

```

ترتبط لغة البايثون النصوص المتقاورة مع بعضها البعض تلقائياً فمثلا لدينا جملتين تمت كتابتها بالسطر نفسه سيعمل البايثون على دمجهما معاً في النتيجة وسيكون شكل الجملة النهائية كالتالي:

```

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (In tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/hadeel/Desktop/str.py ==
>>> 'had''del'
'haddel'
>>> |

```



فضلا عن فهرسة النصوص من خلال python indexing حيث نلاحظ مثلا ان في الكلمة python الحرف الاول p يأخذ [index[0]] والحرف y يأخذ [index[1]]....الخ. كما في التطبيق التالي:

String	p	y	t	h	o	n
Index[]	0	1	2	3	4	5

الفهرسة من اليسار الى اليمين



```
x = "python"
print(x[0])
print(x[1])
print(x[2])
print(x[3])
print(x[4])
print(x[5])
```

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29)
[cloned] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:/Users/hadeel/Desktop/str.py
p
y
t
h
o
n
```

لكن في حال اردنا ان نقوم بالفهرسة العكسيه اي ان نبدأ من n باتجاه p ستكون الفهرسة [index[-1]] لحرف n و [index[-2]]..... الخ

String	p	y	t	h	o	n
Index[]	-6 or 0	-5	-4	-3	-2	-1

الفهرسة من اليمين الى اليسار





<pre>str.py - C:/Users/hadeel/Desktop</pre>	<pre>Python 3.8.2 Shell</pre>
<pre>File Edit Format</pre>	<pre>File Edit Shell Debug Options Window Help</pre>
<pre>x= "python" print(x[-1]) print(x[-2]) print(x[-3]) print(x[-4]) print(x[-5]) print(x[-6])</pre>	
<pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 64 bit (AMD64)] Type "help", "copyright", "credits" or "license()" for more information. >>> ===== RESTART: C:/Users/hadeel/Desktop/str.py</pre>	
<pre>n o h t y p >>> </pre>	

امثلة اخرى على indexing والتي يمكننا من خلالها فهرسة مدى معين من الاحرف حيث تسمى هذه العملية بالاقطاع (slicing) من خلال وضع (:) النقطتين المتعامدتين فمثلا اقتطاع الحروف من الموقع [2] إلى الموقع [5] سيكون تمثيله [2:5] اي ثلات حروف وهم (tho) سيكون الى [5] اي [5] لا يدخل من ضمنها وكما موضح في الامثلة أدناه:

<pre>str.py - C:/Users/hadeel/Desktop</pre>	<pre>Python 3.8.2 Shell</pre>
<pre>File Edit Format Run</pre>	<pre>File Edit Shell Debug Options Window Help</pre>
<pre>x= "python" print(x[2:5])</pre>	
<pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 64 bit (AMD64)] Type "help", "copyright", "credits" or "license()" for more information. >>> ===== RESTART: C:/Users/hadeel/Desktop/str.py</pre>	
<pre>tho >>> </pre>	

<pre>str.py - C:/Users/hadeel/Desktop</pre>	<pre>Python 3.8.2 Shell</pre>
<pre>File Edit Format Run</pre>	<pre>File Edit Shell Debug Options Window Help</pre>
<pre>x= "python" print(x[-3:-1])</pre>	
<pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 64 bit (AMD64)] Type "help", "copyright", "credits" or "license()" for more information. >>> ===== RESTART: C:/Users/hadeel/Desktop/str.py</pre>	
<pre>ho >>> </pre>	



str.py - C:/Users/hadeel/Desktop/str.py (3.8.2) File Edit Format Run Options Window Help <pre>x= "python" print(x[:3]) اقطاع الحروف من 0 الى 3 (غير مشمل)</pre>	Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3afed, Feb 25 2020, 21:27:00) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information >>> ===== RESTART: C:/Users/hadeel/Desktop/str.py ===== >>>
---	--

str.py - C:/Users/hadeel/Desktop/str.py (3.8.2) File Edit Format Run Options Window Help <pre>x= "python" print(x[-2:]) الحروف من الموضع 2 من اليمين (ممشول) إلى نهاية السلسلة #</pre>	Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3afed, Feb 25 2020, 21:27:00) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information >>> ===== RESTART: C:/Users/hadeel/Desktop/str.py ===== >>>
--	--

str.py - C:/Users/hadeel/Desktop/str.py (3.8.2) File Edit Format Run Options Window Help <pre>x= "python" print(x[:-2]+x[-2:])</pre>	Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 21:27:00) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information >>> ===== RESTART: C:/Users/hadeel/Desktop/str.py ===== python >>>
--	---

والامثلة كثيرة عن الاقطاع يمكن ان تبدأ بتجربتك الخاصة مع البايثون وتجربة String slicing



اما الان فلنأتي لشرح طول النص (String Length) حيث يمكن حساب طول النص من خلال الدالة الداخلية للبایثون والتي تدعى `len()` والتي تعمل على حساب عدد الحروف في النص واظهار النتيجة على شاشة التنفيذ. كما في الامثلة ادناه:

str.py - C:/Users/hadeel	Python 3.8.2 Shell
File Edit Format	File Edit Shell Debug Options Window Help
<pre>x= "python" print(len(x))</pre>	Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 21:27:00) [MSC v.1916 64 bit (AMD64)] Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:/Users/hadeel/str.py ===== 6 >>>

str.py - C:/Users/hadeel	Python 3.8.2 Shell
File Edit Format Run	File Edit Shell Debug Options Window Help
<pre>x= "university" print(len(x))</pre>	Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 21:27:00) [MSC v.1916 64 bit (AMD64)] Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:/Users/hadeel/str.py ===== 10 >>>

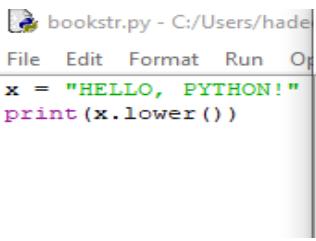
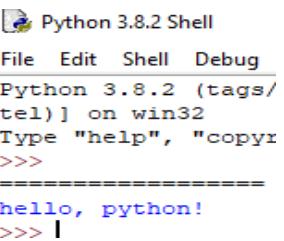
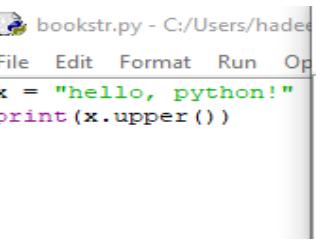
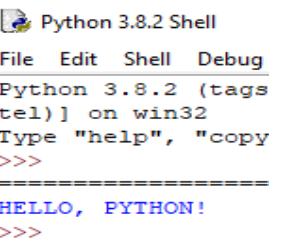
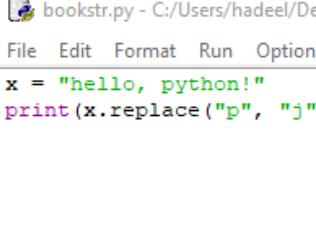
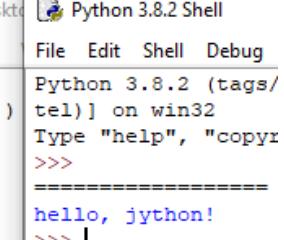
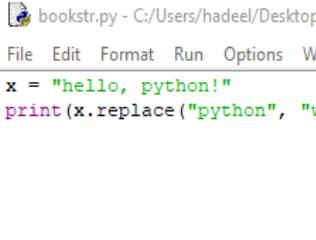
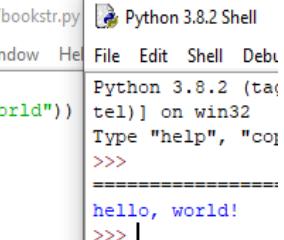
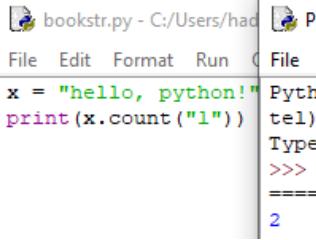
تحتوي البایثون ايضا على مجموعة من الدوال

والتي يمكن استخدامها في النصوص بكل سهولة. وهي كثيرة ومتعددة سنذكر منها

اكثرها شيوعا:





مثال عملي	عن الدالة	اسم الدالة	ت
 	<p>تحوّل هذه الدالة الحروف في السلسلة النصية إلى حروف صغيرة.</p>	lower()	1
 	<p>تحوّل هذه الدالة الحروف في السلسلة النصية إلى حروف كبيرة.</p>	upper()	2
 	<p>تبديل هذه الدالة الجملة او الحروف التي يختارها المستخدم الى جملة وحروف اخرى</p>	replace()	3
 			
 	<p>تعيد هذه الدالة عدد مرات تكرار حرف معين او جملة معين في النصوص</p>	count()	4

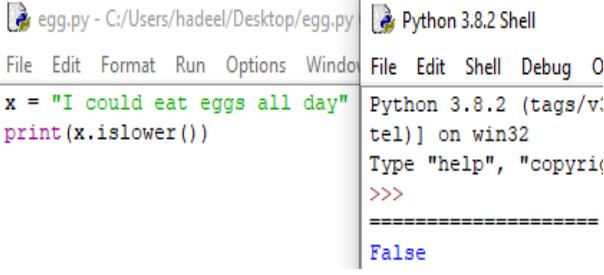
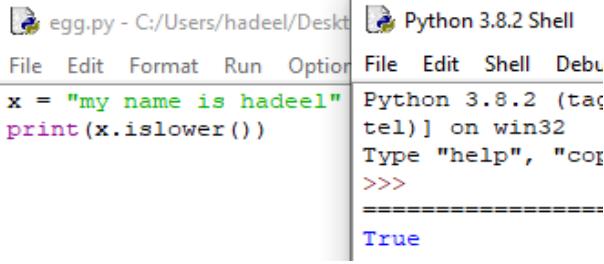


		<p>تعمل هذه الدالة على تقسيم الجملة الى كلمات منفصلة اي الى list</p>	split()	٥
		<p>تحتحقق ما إذا كان النص أرقاماً</p>	isdigit()	٦
		<p>تعمل على إزالة اي مسافات زائدة</p>	rstrip()	٧
		<p>ضم جميع العناصر في مجموعة كنص واحد من خلال استخدام حرف او رمز للتجزئة والذى يعمل كفاصيل</p>	Join()	٨
		<p>تجعل هذه الدالة الأحرف الكبيرة صغرى والكبيرة تحولها الى احرف الصغرى (اي تبديل حالة الحرف)</p>	swapcase	٩ ٠



<pre>x = "Hello, welcome to my book." print(x.find("w")) print(x.find("my")) print(x.find("q"))</pre>	<p>تعد هذه الدالة مشابهة لعمل index حيث أنها تقوم بالبحث عن حرف معين أو جملة وترجع قيمة التواجد الأول للحرف المحددة كما في المثال الموضح وفي حال عدم وجود ذلك الحرف ترجع (-1)</p>	find()	١٠
<pre>x = "Hello, welcome to my book." print(x.title())</pre>	<p>تعمل هذه الدالة على تحويل الجملة الى حالة العنوان من خلال تحويل الحرف الأول في كل كلمة الى حرف كبير</p>	title	١١
<pre>x = "66" print(x.zfill(10))</pre>	<p>تملء السلسلة بالأصفار حتى يبلغ طولها ١٠ أحرف</p>	zfill()	١٢
<pre>x = "I could eat eggs all day" print(x.partition("eggs"))</pre>	<p>تعمل هذه الدالة تقسيم الجملة الى على ثلاثة عناصر:</p> <ul style="list-style-type: none"> • العنصر الأول يحتوي على الجزء قبل الجملة المحددة. • يحتوي العنصر الثاني على الجملة المحددة. • يحتوي العنصر الثالث على الجزء ما بعد الجملة المحددة. 	Partition	١٣



		ملاحظة: تبحث هذه الدالة عن التواجد الأول للجملة المحددة في حال تم تكرارها أكثر من مرة		
		 <pre>x = "I could eat eggs all day" print(x.islower()) False</pre>		
		 <pre>x = "my name is hadeel" print(x.islower()) True</pre>	islower()	١٤
		 <pre>x = "my name is hadeel" print(x.isupper()) False</pre>		
		 <pre>x = "MY NAME IS HADEEL" print(x.isupper()) True</pre>	isupper()	١٥



<pre>x = "MY NAME" print(x.startswith("MY"))</pre>	<pre>x = "MY NAME" print(x.startswith("MY", 0, 6))</pre>	<p>تحقق هذه الدالة مما إذا كانت الـ string تبدأ بكلمة محددة .</p> <p>إضافة إلى ذلك في حال بحثنا في عن string عن كلمة محددة ونحدد القيمة البدائية والنهاية للجملة ستكون النتيجة طبعاً إما True أو ستكون False</p>	startswith h() ١٦
--	--	--	---------------------------------

يحتوي الجدول أعلاه على بعض الدوال المهمة لكن هناك العديد من الدوال الأخرى التي يمكنك ان تبحث عنها ايضاً في حال تم الاحتياج لها .

لكن قبل ان ننتهي من **string** يجب ان نوضح دالة مهمة وهي **str.format()** حيث تستخدم لإدراج ارقام او كلمات معينة خلال الجملة وذلك باستخدام الاقواس المترجة **{}** والامثلة أدناه ستوضح كيفية التعامل مع هذه الدالة:

<pre>x = "My name is {}" print(x.format("hadeel"))</pre>	<pre>Python 3.8.2 (tags/v3.8.2:7b3efca, Feb 24 2020, 17:57:52) [PyPy 6.3.0 (6.3.0+dev-149-gd9f9db)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: My name is hadeel >>></pre>
--	--



 egg.py - C:/Users/hadeel/Desktop/egg.py (3.8.2)	 Python 3.8.2 Shell
<pre> File Edit Format Run Options Window Help x = "My name is {0}, I'm from {1}" print(x.format("hadeel","iraq")) </pre>	<pre> File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" >>> ===== RESTART: C:/... My name is hadeel, I'm from iraq >>> </pre>

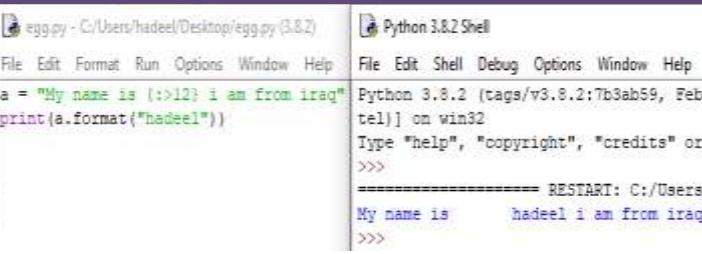
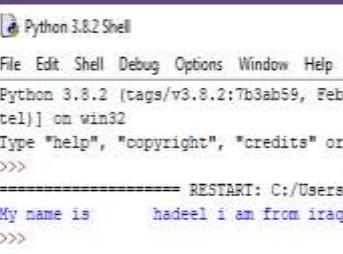
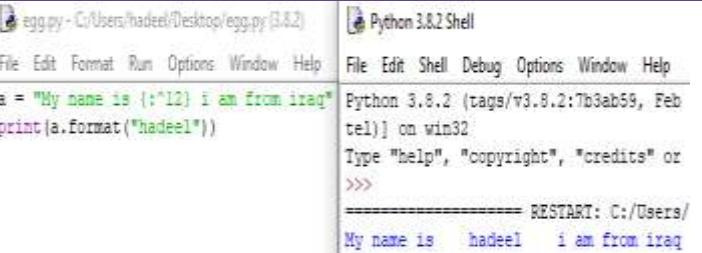
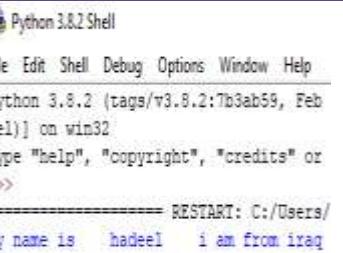
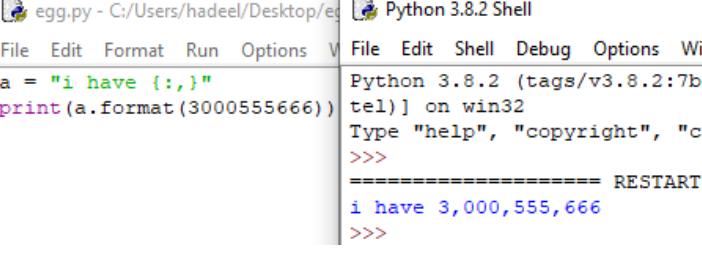
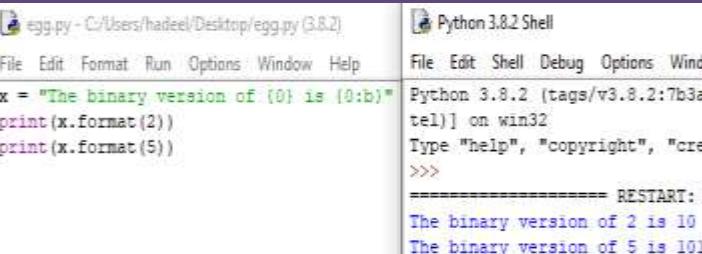
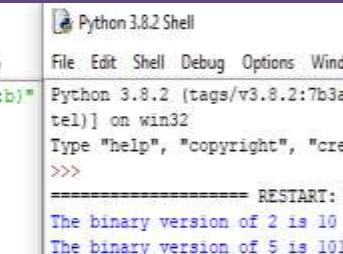
نلاحظ في المثال اعلاه ان الكلمة الاولى "Hadeel" اخذت الموقع 0 و الكلمة "Iraq" استقرت في موقع 1

ولكن يمكن استخدام طريقة اخرى وهي من خلال المتغيرات:

 egg.py - C:/Users/hadeel/Desktop/egg.py (3.8.2)	 Python 3.8.2 Shell
<pre> File Edit Format Run Options Window Help a = "My name is {x}, I'm from {y}" print(a.format(x= "hadeel",y= "iraq")) </pre>	<pre> File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credit" >>> ===== RESTART: C:/... My name is hadeel, I'm from iraq >>> </pre>

فضلا عن ذلك هناك العديد من **Formatting Types** سأذكر خمسة انواع منها بشكل سريع وكما هي القاعدة تقول اذا اردت المزيد من **formatting** ابحث بسهولة عنها مع العم **:Google**



مثال	عملها	انواع Formatting
 <pre>a = "My name is {:<i>12</i>} i am from iraq" print(a.format("hadeel"))</pre>	 <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:/Users/ My name is hadeel i am from iraq >>></pre>	<p>محاذاة الكلمة إلى اليسار من خلال مسافة محددة</p>
 <pre>a = "My name is {>:<i>12</i>} i am from iraq" print(a.format("hadeel"))</pre>	 <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:/Users/ My name is hadeel i am from iraq >>></pre>	<p>محاذاة الكلمة إلى اليمين من خلال مسافة محددة</p>
 <pre>a = "My name is {:<i>12</i>} i am from iraq" print(a.format("hadeel"))</pre>	 <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:/Users/ My name is hadeel i am from iraq >>></pre>	<p>توسيط المسافة للكلمة في المنتصف</p>
 <pre>a = "i have {:,}" print(a.format(3000555666))</pre>	 <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: i have 3,000,555,666 >>></pre>	<p>إضافة فاصلة ما بين الارقام</p>
 <pre>x = "The binary version of {0} is {0:b}" print(x.format(2)) print(x.format(5))</pre>	 <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: The binary version of 2 is 10 The binary version of 5 is 101 >>></pre>	<p>تعمل على تحويل الرقم إلى رقم ثنائي</p>



٢. الارقام (Numbers)

سبق وان تحدثنا عن الارقام int , float في (متغيرات البايثون) وتعلمنا هناك انه نستطيع معرفة نوع المتغير من خلال() type اضافة الى ذلك استطعنا جمع عددين int وايضا عددين float من خلال (+)

لكن الان يجب ان نعترف بان هناك في الحقيقة ثلاث انواع من الارقام وليس اثنان وهي كالتالي:

- رقم صحيح: عدد صحيح (integer) هو عدد صحيح سواء كان موجب أو سالب ،

($x = -156$) ($x = 12345678910$)

- رقم عشري: هو رقم يحتوي على فاصلة . والرقم العشري يمكن ان يكون موجب أو سالب

($x=1.0$) ($x= -10.5$)

- رقم المركب : هو عدد يتكون من جزئين جزء حقيقي وجزء خيالي (j) وسيكون شكل الرقم

كالتالي (4+6j)

الأمثلة ادناه تمثل الانواع الثلاثة في البايثون :

<pre> int.pyw - C:\Users\hadeel.m.taher Python 3.8.2 Shell File Edit Format Run File Edit Shell Debug Options Window x=5 y=4.5 z = 4+6j print(type(x)) print(type(y)) print(type(z)) </pre>	<pre> Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [Clang 9.0.0 (clang-900.0.39.2) on darwin] Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel.m.taher\int.pyw <class 'int'> <class 'float'> <class 'complex'> </pre>
--	--



The screenshot shows two windows side-by-side. On the left is a code editor window titled 'int.pyw - C:\Users\hadeel' containing the following Python code:

```
x=5+6
y=4.0+5.0
z = 4+6j+ 1+4j
print(x)
print(y)
print(z)
```

On the right is a Python 3.8.2 Shell window titled 'Python 3.8.2 Shell' showing the output of the code:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ===== RESTART: C:\Users\hadeel\int.pyw =====
11
9.0
(5+10j)
>>>
```

يمكنك التحويل من نوع إلى آخر باستخدام الطرق `int()` و `float()` و `complex()`

The screenshot shows two windows side-by-side. On the left is a code editor window titled 'int.pyw - C:\Users\hadeel' containing the following Python code:

```
x=5
y=4.5
z = 4+6j
x= float(x)
y= complex(y)
print(x)
print(y)
```

On the right is a Python 3.8.2 Shell window titled 'Python 3.8.2 Shell' showing the output of the code:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ===== RESTART: C:\Users\hadeel\int.pyw =====
5.0
(4.5+0j)
>>>
```

هناك ملاحظة مهمة وهي ان العدد المركب لا يمكن تحويله الى عدد صحيح او عدد عشري كما مبين في المثال اعلاه حيث لم نستطع ان نحول العدد المركب الى صحيح .

لكن يمكن تحويل العدد الصحيح والعشري الى عدد مركب

وفي ختام فقرة الارقام يجب ان نتطرق الى الرقم العشوائي (Random Number) حيث تستطيع ان تستدعي مكتبة `random` ومن خلالها تقوم بتوليد رقم عشوائي وفي حال تكرار التنفيذ ستعطي رقم ضمن المدى المحدد كما في المثال ادناه ستكون الارقام عشوائية من 1 الى 10 لكن من الضروري البدء باستدعاء المكتبة:



 int.pyw - C:\Users\hadeel\Desktop\int.py	 Python 3.8.2 Shell
File Edit Format Run Options Window import random print(random.randrange(1,10))	File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b1 tel) on win32 Type "help", "copyright", "c >>> ===== RESTART: 4

 int.pyw - C:\Users\hadeel\Desktop\int.py	 Python 3.8.2 Shell
File Edit Format Run Options Window import random print(random.randrange(1,10))	File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b1 tel) on win32 Type "help", "copyright", "c >>> ===== RESTART: 5

٣. القيم المنطقية (Boolean)

تمثل القيم المنطقية إحدى القيمتين: صواب أو خطأ تستخدم كثيراً مع if ولكن يمكن استخدامها بشكل منفرد كالمقارنة حيث تعطي True or false أو من خلال الدالة () bool والتي تسمح بتقييم القيمة التي بداخلها وغالباً ما تكون . True

 int.pyw - C:\Users\hadeel\Desktop\int.py	 Python 3.8.2 Shell
File Edit Format Run Options Window print(10 > 9) print(10 == 9) print(10 < 9)	File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ak tel) on win32 Type "help", "copyright", "crec >>> ===== RESTART: C: True False False ---



□

int.pyw - C:\Users\hadeel\Desktop	Python 3.8.2 Shell
File Edit Format Run Options	File Edit Shell Debug Options Window Help
print(bool("python")) print(bool(10))	Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 tel) on win32 Type "help", "copyright", "credits" or "license" >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw True True

□

لـكـن هـنـاكـ حالـاتـ خـاصـةـ عـادـةـ ماـ تـكـونـ النـتـيـجـةـ (False) وـهـيـ كـالتـالـيـ:

int.pyw - C:\Users\hadeel\Desktop	Python 3.8.2 Shell
File Edit Format Run	File Edit Shell Debug Options Window Help
print(bool(())) print(bool([])) print(bool({})) print(bool(0)) print(bool("")) print(bool(False)) print(bool(None))	Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 2 tel) on win32 Type "help", "copyright", "credits" or "license() >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw False False False False False False False False



٤. متسلسلات او مصفوفات list & tuple

القائمة list هي عبارة عن مجموعة مرتبة ومتغيرة بينما ال tuple هي مجموعة مرتبة وغير قابلة للتغيير.

لذا نبدأ ب list :

تتميز هذه القائمة ب الأقواس المربعة [] كما نلاحظ في المثال أدناه:

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	 Python 3.8.2 Shell
<pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c","java"] print(mylist)</pre>	<pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab5f tel) on win32 Type "help", "copyright", "credit" >>> ===== RESTART: C:\U ['python', 'c++', 'c', 'java'] ``` </pre>

: يمكننا الوصول إلى عناصر القائمة list من خلال الرجوع إلى رقم الفهرس (كما هو الحال في

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	 Python 3.8.2 Shell
<pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c","java"] print(mylist[0]) print(mylist[1]) print(mylist[2]) print(mylist[3])</pre>	<pre>File Edit Shell Debug Options W Python 3.8.2 (tags/v3.8.2:7b3ab5f tel) on win32 Type "help", "copyright", "c >>> ===== RESTART: python c++ c java ``` </pre>



وايضا يمكننا تحديد مدى معين من الفهرس (كما في الفهرسة التي تم شرحها مسبقا).

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c","java"] print(mylist[0:2])</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Wi Python 3.8.2 (tags/v3.8.2:7b tel) on win32 Type "help", "copyright", "c: >>> ===== RESTART: ['python', 'c++'] >>></pre>
---	---

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c","java"] print(mylist[-3:-1])</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Wi Python 3.8.2 (tags/v3.8.2:7b tel) on win32 Type "help", "copyright", "c: >>> ===== RESTART: ['c++', 'c'] >>></pre>
---	--

يمكن ايضا تغيير قيمة عنصر معين ، من خلال رقم الفهرس فمثلا يمكننا تغيير لغة C واستبدلها ب SQL

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c","java"] mylist[2]= "SQL" print(mylist)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window I Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits >>> ===== RESTART: C:\Us ['python', 'c++', 'SQL', 'java'] >>> ...</pre>
---	---



لحساب عدد العناصر الموجودة بالقائمة، نستخدم الدالة `len()` و التي سيتضح لنا من خلالها ان عدد العناصر الموجودة في القائمة هي اربعة عناصر فقط.

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c","java"] print(len(mylist))</pre>	 Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:08:11) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw 4 >>></pre>
--	---

يمكن ملء قائمة فارغة بالعناصر من خلال دالة `append()` وايضا يمكن اضافة عنصر لقائمة تحتوي على عناصر من خلال نفس الدالة :

لكي نستعرض دالة `append()` بدأية سنضع قائمة فارغة ثم نقوم بملئها بالعناصر :

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options mylist= [] print(mylist) mylist.append("c++") print(mylist) mylist.append("python") print(mylist)</pre>	 Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:08:11) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw [] ['c++'] ['c++', 'python'] >>></pre>
---	---

نلاحظ ان دالة `append()` تعمل على اضافة العناصر بنهاية القائمة فقط لكن يمكننا ايضا اضافة عناصر محددة بموقع محدد بالاعتماد على الفهرسة اي `insert(index, value)`



int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c", "java"] print(mylist) mylist.insert(1,"SQL") print(mylist)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['python', 'c++', 'c', 'java'] ['python', 'SQL', 'c++', 'c', 'java'] >>> ---</pre>
---	---

من الجهة الأخرى نستطيع حذف عنصر محدد (specified item) ونركز على كلمة **محدد** من القائمة

باستخدام دالة **remove()** كالتالي :

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c", "java"] print(mylist) mylist.remove("python") print(mylist)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['python', 'c++', 'c', 'java'] ['c++', 'c', 'java'] >>></pre>
--	--

اما اذا اردنا حذف عنصر بدون تحديد اي سيقوم البايثون بحذف العنصر الاعلى index بالاعتماد على الفهرسة

(سيكون هذا من خلال دالة **:pop()**)

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c", "java"] print(mylist) mylist.pop() print(mylist)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['python', 'c++', 'c', 'java'] ['python', 'c++', 'c'] >>></pre>
---	--



ونستطيع حذف العناصر جميعها من القائمة وجعل القائمة فارغة تماما وبسهولة من خلال دالة عملية جدا وهي

`clear()`

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	 Python 3.8.2 Shell
<pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c", "java"] print(mylist) mylist.clear() print(mylist)</pre>	<pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw [] >>></pre>

يمكنا الجمع بين قائمتين من خلال `(+)` operator وبسهولة كما في المثال التالي:

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	 Python 3.8.2 Shell
<pre>File Edit Format Run Options list1= ["python", "c++"] list2=["c", "java"] list3= list1+list2 print(list3)</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['python', 'c++', 'c', 'java'] >>></pre>

يمكنا ايضا توسيع القائمة بدمج قائمة اخرى معها اي اضافه عناصر من قائمة الى قائمة ثانية من خلال دالة

`extend()`

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	 Python 3.8.2 Shell
<pre>File Edit Format Run Options list1= ["python", "c++"] list2=["c", "java"] list1.extend(list2) print(list1)</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['python', 'c++', 'c', 'java'] >>></pre>



فضلا عن ذلك يمكن استخدام list constructor الى اخرى وسنلاحظ ان القائمة ستكون بأقواس مدوره () وذلك لأنها ستكون constructor الى القائمة الجديدة :

int.pyw - C:\Users\hadeel\Desktop\int.pyw <pre>File Edit Format Run Options Window list1 = list(("python", "c++")) print(list1)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:42) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['python', 'c++']</pre>
--	---

ولكن هل يمكننا عكس القائمة ؟ الجواب نعم لأن البيشون سهلت تلك العملية من خلال دالة reverse()

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c", "java"] print(mylist) mylist.reverse() print(mylist)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:42) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['python', 'c++', 'c', 'java'] ['java', 'c', 'c++', 'python']</pre>
--	--

واخيرا يمكننا عمل فرز للقائمة أبجديا من خلال دالة sort()

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mylist= ["python", "c++", "c", "java"] mylist.sort() print(mylist)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:42) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['c', 'c++', 'java', 'python']</pre>
---	---



لقد انتهينا من التفاصيل المتعلقة بال list

والان حان دور التعرف على Tuple :-

tuple هي مجموعة مرتبة وغير قابلة للتغيير حيث يتم كتابة tuple بأقواس مستديرة () .

ويمكن القول ان طباع tuple تشبه الى حد ما list وسنلاحظ الان اوجه التشابه والاختلاف من خلال صفاتها التالية :

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mytuple= ("python", "c++", "c", "java") print(mytuple)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:42) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ('python', 'c++', 'c', 'java') >>></pre>
--	---

في البايثون يتم التعرف على العناصر التي بدون اقواس واعطاء النتيجة بانها tuple:

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help mytuple= "python", "c++", "c", "java" print(mytuple)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:42) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ('python', 'c++', 'c', 'java') >>></pre>
--	---

نلاحظ اعلاه ان البايثون اعطى النتيجة ذاتها في الحالتين بأقواس او بدون اقواس وتعرف عليها على انها tuple يمكننا ايضا الوصول إلى اي عنصر داخل tuple من خلال الفهرسة index number واعتقد ان هذه القاعدة سبق وان تم شرحها في list.



<pre>*int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)* File Edit Format Run Options Window Help mytuple= ("python", "c++", "c", "java") print(mytuple[1])</pre>	<pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:00) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw c++ >>></pre>
---	---

قبل ان نكمل في tuple يمكننا توضيح الفرق الرئيسي بين list وtuple هو أن list قابلة للتغيير وtuple ليست كذلك اي انها تحتوي على قيمة ثابتة لا يمكن تغييرها في هذه الحالة اذا اردنا تغيير قيمة فيجب انشاء tuple جديدة .

يمكننا معرفة نوع tuple من خلال الدالة type()

<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)* File Edit Format Run Options Window Help x = ("python", "c++", "c", "java") print(type(x))</pre>	<pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:00) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw <class 'tuple'> >>></pre>
--	---

لكن كيف يميز البايثون tuple من string سترى قوة الملاحظة لديك في المثال التالي :

<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)* File Edit Format Run Options Window Help x =("python") y = ("python",) print(type(x)) print(type(y))</pre>	<pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:00) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw <class 'str'> <class 'tuple'></pre>
--	--



الفرق بينهما واضح جدا وهي الفاصلة حيث نلاحظ ان البايثون ميز بان تكون الجملة بدون فاصلة ولكن بمجرد اضافتها تم تحويل نوعها الى tuple.

والآن سنستكمل شرح الفهرسة بالأمثلة التالية :

The screenshot shows two windows side-by-side. On the left is a code editor window titled "int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8)". It contains the following Python code:`x = ("python", "c++", "c", "java")
print(x[-1])`On the right is a Python 3.8.2 Shell window. The output shows:`Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2019, 17:23:44) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw
java
>>>
...`This demonstrates that the code prints the last element of the tuple, which is "java".

The screenshot shows two windows side-by-side. On the left is a code editor window titled "int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8)". It contains the following Python code:`x = ("python", "c++", "c", "java")
print(x[1:3])`On the right is a Python 3.8.2 Shell window. The output shows:`Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2019, 17:23:44) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw
('c++', 'c')
>>>
...`This demonstrates that the code prints a tuple containing the second and third elements of the original tuple, which are "c++" and "c".

تذكرة دائما أن العنصر الأول يحتوي على فهرس 0 وان نهاية tuple تبدأ بـ-1

The screenshot shows two windows side-by-side. On the left is a code editor window titled "int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8)". It contains the following Python code:`x = ("python", "c++", "c", "java")
print(x[-4:-1])`On the right is a Python 3.8.2 Shell window. The output shows:`Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2019, 17:23:44) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw
('python', 'c++', 'c')
>>>
...`This demonstrates that the code prints a tuple containing the last three elements of the original tuple, which are "python", "c++", and "c".



يمكن ان تحتوي بداخلها على list ، فضلا عن ذلك يمكن ايضا تحويل tuple الى list كالتالي:

int.pyw - C:\Users\hadeel\Desktop\int. <pre>File Edit Format Run Options Window Help x =("python", "c++", [1,2,3]) print(type(x)) print(x)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\ <class 'tuple'> ('python', 'c++', [1, 2, 3]) ---</pre>
---	---

int.pyw - C:\Users\hadeel\int.py <pre>File Edit Format Run Options Window Help x =("python", "c++") y= list(x) print(y)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\ ['python', 'c++'] >>></pre>
--	---

ولمعرفة طول tuple نستخدم الدالة len() كما هو الحال في list

int.pyw - C:\Users\hadeel\int.py <pre>File Edit Format Run Options Window Help x =("python", "c++") print(len(x))</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\ 2 >>></pre>
--	---



وللتتأكد من ان tuple غير قابل للتغيير سنثبت ذلك (حيث سيظهر لنا error)

int.pyw - C:\Users\hadeel\ Desktop	Python 3.8.2 Shell
<pre> File Edit Format Run Options x = ("python", "c++") x[1] = "c" print(x) </pre>	<pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC tel]) on win32 Type "help", "copyright", "credits" or "license()" for more in >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw = Traceback (most recent call last): File "C:\Users\hadeel\Desktop\int.pyw", line 2, in <module> x[1] = "C" TypeError: 'tuple' object does not support item assignment >>> </pre>

ولحذف عناصر نستخدم دالة del() وذلك لحذف tuple بأكملها لأننا لا نستطيع حذف عنصر منها لأنها غير

قابلة للتغيير

int.pyw - C:\Users\hadeel\ Desktop	Python 3.8.2 Shell
<pre> File Edit Format Run Options x = ("python", "c++") del (x) print(x) </pre>	<pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22 tel]) on win32 Type "help", "copyright", "credits" or "license()" for more in >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw = Traceback (most recent call last): File "C:\Users\hadeel\Desktop\int.pyw", line 3, print(x) NameError: name 'x' is not defined >>> </pre>

المثال اعلاه لا يوجد فيه خطأ والسبب اننا قمنا بحذف tuple ولذلك بالنسبة للبايثون لا يوجد شيء يدعى X لهذا

ظهور لنا error. يمكن دمج tuple مع بعضهما البعض كالتالي:

int.pyw - C:\Users\hadeel\Desktop	Python 3.8.2 Shell
<pre> File Edit Format Run Options tuple1 = ("python", "c++") tuple2 = ("c", "java") tuple3 = tuple1 + tuple2 print(tuple3) </pre>	<pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Fe tel]) on win32 Type "help", "copyright", "credits" o >>> ===== RESTART: C:\Users ('python', 'c++', 'c', 'java') >>> ``` </pre>



من الممكن أيضًا استخدام tuple في constructor حيث نلاحظ اتنا استخدمنا الاقواس مرتين :

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	 Python 3.8.2 Shell
<pre>File Edit Format Run Options Window Help x = tuple(("python", "c++", "c","java")) print(x)</pre>	<pre>File Edit Shell Debug Options Window H Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" >>> ===== RESTART: C:\Use ('python', 'c++', 'c', 'java') >>></pre>

Sets, frozenset .

وهي مجموعات ليس لها حجم ثابت. و لا يمكن حذف قيمها . في لغة البايثون يتم كتابة المجموعات (Set) باقواس مترجحة { }.

يمكننا انشاء المجموعات كالاتي:

 int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	 Python 3.8.2 Shell
<pre>File Edit Format Run Options Window x = {"python", "c++", "c","java"} print(x) print(type(x))</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Fe tel) on win32 Type "help", "copyright", "credits" o >>> ===== RESTART: C:\Users {'c', 'python', 'c++', 'java'} <class 'set'> >>></pre>

نلاحظ ان البايثون قد تعرف على المجموعة فقط بعد ان قمنا بتغيير الاقواس الى الاقواس المترجحة .

ملاحظة مهمة جدا يجب الانتباه لها في المجموعات وهي:

لا يمكنك الوصول إلى العناصر في المجموعة بالرجوع إلى فهرس ، وذلك لأن set غير مرتبة لهذا ان العناصر ليس لها فهرس.



نستطيع تضمين string ,number, tuple داخل المجموعة لكن لا يمكن تضمين list داخلها كالتالي:

int.pyw - C:\Users\hadeel\Desktop	Python 3.8.2 Shell
<pre>File Edit Format Run Options x= {"hello",604,(1,2,3)} print(type(x)) print(x)</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\1 <class 'set'> {'hello', 604, (1, 2, 3)}</pre>

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	<pre>>>> /usr/bin/python Type "help", "copyright", "credits" or "license()" for more in >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw = Traceback (most recent call last): File "C:\Users\hadeel\Desktop\int.pyw", line 1, in <module> x= [{"python", "c++"},604,(1,2,3)} print(type(x)) print(x) TypeError: unhashable type: 'list' >>></pre>
---	--

في المجموعة Set لا يمكن أن تحتوي على تكرارات(اي العناصر لا تتكرر داخل المجموعة)

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	Python 3.8.2 Shell
<pre>File Edit Format Run Options Window Help x= {"python", "c++", "c","java", "c++","c"} print(x)</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\1 {'c', 'python', 'c++', 'java'} >>> >>></pre>

على الرغم من الترتيب العشوائي للعناصر داخل المجموعة الا ان العناصر المتكررة تم حذفها عند طباعة النتيجة.

ايضا يمكننا التتحقق داخل المجموعة مما إذا كان احد العناصر موجوداً في المجموعة ام لا من خلال (in) والاجابة

ستكون فقط بنعم او لا



int.pyw - C:\Users\hadeel\Desktop\int.pyw (3) <pre> File Edit Format Run Options Window x = {"python", "c++", "c", "java"} print("java" in x) print("SQL" in x) </pre>	Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw True False >>> </pre>
--	---

في المثال أعلاه نلاحظ ان الجافا(java) موجودة ضمن المجموعة فكانت النتيجة True اما SQL فبحث عنها داخل المجموعة ولم يجدها فأعطتها النتيجة False. ولإضافة عنصر واحد إلى مجموعة ، استخدم طريقة

.add()

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3) <pre> File Edit Format Run Options x= {"python", "c++", "c"} x.add("java") print(x) </pre>	Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw {'c++', 'java', 'python', 'c'} >>> </pre>
--	---

ولتحديد عدد العناصر الموجودة في المجموعة (set)، نستخدم الدالة المتعارف عليها والتي استخدمناها كثيراً جداً

.len()

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3) <pre> File Edit Format Run Options x= {"python", "c++", "c"} print(len(x)) </pre>	Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw 3 >>> </pre>
---	--



فضلا عن يمكننا استخدام الدوال السابقة وهي `remove()` في حال ان العنصر المراد حذفه موجود في المجموعة، ودالة `pop()` لازالة العنصر الاخير الموجود في المجموعة، ودالة `clear()` لإفراغ المجموعة بأكملها، ودالة `del()` لحذف المجموعة وستكون الامثلة التالية للتطبيق العملي لتلك الدوال بالتالي:

int.pyw - C:\Users\hadeel\Desktop <pre>x= ["python", "c++", "c"] x.remove("c") print(x)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2021, 22:43:14) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['python', 'c++']</pre>
--	--

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8) <pre>x= ["python", "c++", "c", "java"] print(x.pop()) print(x)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2021, 22:43:14) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ['c', 'python', 'c++']</pre>
---	---

int.pyw - C:\Users\hadeel\Desktop <pre>x= ["python", "c++", "c"] x.clear() print(x)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2021, 22:43:14) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw set()</pre>
--	--



<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw</pre> <p>File Edit Format Run Options View</p> <pre>x = ["python", "c++", "c"] del x print(x)</pre>	<pre>Python 3.8.2 Shell</pre> <p>File Edit Shell Debug Options Window Help</p> <pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw Traceback (most recent call last): File "C:\Users\hadeel\Desktop\int.pyw", line 3, in <module> print(x) NameError: name 'x' is not defined</pre>
--	---

في مثال دالة `del` لم تتعرف البيايثون على `X` وذلك لأن تم حذفها قبل الطباعة لهذا لم تعد `X` موجودة . وبما أننا الان في المجموعة فمن المؤكد يمكننا عمل اتحاد بين مجموعتين من خلال الدالة `union()` على ان لا تتكرر العناصر :

<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw</pre> <p>File Edit Format Run Options View</p> <pre>x = ["python", "c++"] y= ["c", "java"] z= x.union(y) print(z)</pre>	<pre>Python 3.8.2 Shell</pre> <p>File Edit Shell Debug Options Window Help</p> <pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw {'python', 'java', 'c++', 'c'}</pre>
--	---

ويمكن عمل تحديث لمجموعة مع مجموعة اخرى من خلال `update()` على ان لا تتكرر العناصر ايضا لأن هذه القاعدة أساسية في المجموعات:

<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw</pre> <p>File Edit Format Run Options View</p> <pre>x = ["python", "c++"] y= {"c", "java", "c++"} x.update(y) print(x)</pre>	<pre>Python 3.8.2 Shell</pre> <p>File Edit Shell Debug Options Window Help</p> <pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw {'python', 'c', 'java', 'c++'}</pre>
---	---



لقد انتهينا من المجموعات لكن يجب ان نوضح فكرة المجموعات المجمدة والتي تسمى ب(frozenset) حيث تعد frozenset فئة جديدة لها خصائص مجموعة، ولكن لا يمكن تغيير عناصرها ابدا بمجرد انشائها. كما هو الحال في tuples حيث تعد قوائم ثابتة، لذا فان frozensets هي مجموعات ثابتة. من ناحية أخرى غالبا ما تستخدم تجميد مجموعات frozensets كمفاتيح للقاموس (والذى سيتم شرحها لاحقا)

يمكن تطبيق الدوال التالية على هذا النوع من المجاميع :

copy(), difference(), intersection(), isdisjoint(), issubset(), issuperset(),
.symmetric_difference() and union

لكن لا يمكن تطبيق (add or remove) وذلك لاننا عرفناها سابقا بانها ثابتة (جامدة)

```
int.pyw - C:\Users\hadeel\Desktop\int.pyw
File Edit Format Run Options Window
x =frozenset({"python", "c++"})
print(type(x))

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC
tel] on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw
<class 'frozenset'>
~~~
```

```
int.pyw - C:\Users\hadeel\Desktop\int.pyw
File Edit Format Run Options Window
x =frozenset({"python", "c++"})
x.remove("c++")
print(x)

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC
tel] on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw ==
Traceback (most recent call last):
  File "C:\Users\hadeel\Desktop\int.pyw", line 2, in <module>
    x.remove("c++")
AttributeError: 'frozenset' object has no attribute 'remove'
>>>
```



٦. القاموس Dictionaries

القاموس هو مجموعة غير منظمة وقابلة للتغيير والفرستة. في البيشون تتم كتابة القواميس بأقواس متعلقة key {} ولها ما يعرف ب keys and values حيث يفصل ما بين المفتاح والقيمة نقطتين متعددين () value . وهذه المفاتيح والقيم مهمة جدا في القاموس .

هناك ملاحظة مهمة وقد تكون قاعدة في القاموس :

يمكن أن تكون من أي نوع بيانات حيث من الممكن أن تتكرر ، على عكس keys فيجب أن تكون من النوع غير القابل للتغيير ويجب أن تكون فريدة.

ولإنشاء قاموس:

int.pyw - C:\Users\hadeel\Desktop\int.pyw	Python 3.8.2 Shell
<pre>File Edit Format Run Options W cars= {} # empty dictionary print(type(cars))</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw <class 'dict'> >>> >>></pre>

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	Python 3.8.2 Shell
<pre>File Edit Format Run Options Window Help cars = { "brand": "BMW", "year": "1964"} print(cars) print(type(cars))</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw {'brand': 'BMW', 'year': '1964'} <class 'dict'> ...</pre>



ولمعرفة عدد عناصر القاموس نستخدم دالتنا الشهيرة :

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help cars = { "brand": "BMW", "year": "1964"} print(len(cars))</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\ 2 >>></pre>
--	---

حيث تعد كل (قيمة وفتح) عنصر واحد لهذا قاموسنا الذي أنشأناه يحتوي على عنصرين فقط .

والآن سوف نتعلم كيفية طباعة value لكل key بشكل منفرد كالاتي:

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help nums = {1: "one", 2: "two", 3: "three"} print(nums) print(nums[1]) print(nums[2]) print(nums[3])</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\ {1: 'one', 2: 'two', 3: 'three'} one two three >>></pre>
---	--

حيث تعد الطريقة اعلاه هي الاسهل لان هناك طريقة اخرى وهي من خلال دالة get() حيث نستطيع الحصول على قيمة المفتاح من خلالها:

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help nums = {1: "one", 2: "two", 3: "three"} print(nums.get(1)) print(nums.get(2)) print(nums.get(3))</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\ one two three ~~~</pre>
---	--



طباعة كل القيم الموجودة في القاموس من خلال دالة values():

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} print(nums.values())</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 20:59:52) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw dict_values(['one', 'two', 'three']) >>> >>></pre>
--	--

نلاحظ ان النتيجة ذاتها فقد حصلنا على قيم المفاتيح في القاموس بطرق مختلفة.

من جهة اخرى نستطيع طباعة مفاتيح القاموس من خلال keys():

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} print(nums.keys())</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 20:59:52) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw dict_keys([1, 2, 3]) >>></pre>
--	---

سأتي لمفهوم تغيير(تحديث) القيم ليس المفتاح لنركز معا يمكننا تغيير القيم فقط من خلال اضافة قيمة

جديدة وهذا ما يسمى بتغيير القيم (Change Values):

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre>File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} nums[1] = "ten" print(nums)</pre>	Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 20:59:52) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw {1: 'ten', 2: 'two', 3: 'three'} >>></pre>
---	---



فضلا عن ذلك يمكن اضافة عنصر كامل للقاموس (العنصر يحتوي على مفتاح وقيمة) كالتالي:

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	Python 3.8.2 Shell
<pre>File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} print(nums) nums[4] = "four" print(nums)</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 21:27:18) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\int.pyw {'one': 1, 'two': 2, 'three': 3} {'one': 1, 'two': 2, 'three': 3, 'four': 4} >>></pre>

عند طباعة القاموس الاول كان يحتوي على ٣ عناصر لكن بعد الاضافة وطباعة القاموس لاحظنا ان القاموس احتوى على اربعة عناصر كما في المثال الموضح اعلاه.

بعد اضافة عناصر هل يمكننا ازالة العناصر ياترى وكيف؟

نعم يمكن إزالة العناصر من القاموس باستخدام دالة `pop()` حيث تؤدي هذه الطريقة إلى إزالة عنصر باستخدام المفتاح المقدم وإرجاع القيمة. كذلك يمكن استخدام طريقة `popitem()` لإزالة عنصر عشوائي (مفتاح: قيمة) من القاموس. اضافة الى ذلك يمكن إزالة جميع العناصر مرة واحدة باستخدام طريقة `clear()` واخيرا يمكننا أيضاً استخدام `del` لإزالة العناصر بشكل فردي لعنصر أو القاموس بأكمله.

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)	Python 3.8.2 Shell
<pre>File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} print(nums.pop(2)) print(nums)</pre>	<pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 21:27:18) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. >>> ===== RESTART: C:\Users\hadeel\int.pyw {'one': 1, 'three': 3} {'one': 1}</pre>



<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} print(nums.popitem()) print(nums)</pre>	<pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel)] on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\ (3, 'three') {1: 'one', 2: 'two'} >>></pre>
---	--

<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} nums.clear() print(nums)</pre>	<pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, F tel)] on win32 Type "help", "copyright", "credits" >>> ===== RESTART: C:\User {} ~~~</pre>
--	---

<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} del nums print(nums)</pre>	<pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC tel]] on win32 Type "help", "copyright", "credits" or "license()" for more infor >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw = Traceback (most recent call last): File "C:\Users\hadeel\Desktop\int.pyw", line 3, in <module> print(nums) NameError: name 'nums' is not defined ~~~</pre>
--	---

<pre>int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) File Edit Format Run Options Window Help nums = {1:"one",2:"two", 3:"three"} del nums[2] print(nums)</pre>	<pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC tel]] on win32 Type "help", "copyright", "credits" or "license()" >>> ===== RESTART: C:\Users\hadeel\Desktop\int.pyw = {1: 'one', 3: 'three'} >>> >>></pre>
---	--



The screenshot shows two windows side-by-side. On the left is a code editor window titled 'int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)'. It contains the following Python code:

```
nums = {1:"one",2:"two", 3:"three"}
del nums
print(nums)
```

On the right is a 'Python 3.8.2 Shell' window. The output shows:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29)
[1]: tel]) on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw
Traceback (most recent call last):
  File "C:\Users\hadeel\Desktop\int.pyw", line 3, in <module>
    print(nums)
NameError: name 'nums' is not defined
>>>
```

كما تعلمنا ان `del` قام بحذف القاموس بكلمه فلم يعد البايثون يستطيع ان يتعرف على `nums` و من اتجاه آخر نستطيع عمل تحديث للقاموس من خلال دالة `:update()`

The screenshot shows two windows side-by-side. On the left is a code editor window titled 'int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)'. It contains the following Python code:

```
nums = {1:"one",2:"two", 3:"three"}
nums.update({4:"four"})
print(nums)
```

On the right is a 'Python 3.8.2 Shell' window. The output shows:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29)
[1]: tel]) on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw
{1: 'one', 2: 'two', 3: 'three', 4: 'four'}
>>>
```

يمكننا اختبار ما اذا كان العنصر موجود داخل القاموس ام لا من خلال `in` وتسماى هذه الطريقة (Membership Test) وكما تعلمنا ان نتيجة `in` هي اما True او False

The screenshot shows two windows side-by-side. On the left is a code editor window titled 'int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)'. It contains the following Python code:

```
nums = {1:"one",2:"two", 3:"three"}
print( 1 in nums)
print( 5 in nums)
print( 6 in nums)
print( 3 in nums)
```

On the right is a 'Python 3.8.2 Shell' window. The output shows:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29)
[1]: tel]) on win32
Type "help", "copyright", "credits" or "license()" for more information
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw
True
False
False
True
>>>
```



نلاحظ اعلاه انه تم الاعتماد على المفاتيح في معرفة ما اذا كان العنصر موجود او لا في القاموس ولا يتم الاعتماد ابدا على القيمة.

واخر ما نستطيع تعلمه في القواميس هي كيفية انشاء قواميس متداخلة (*Nested Dictionaries*) حيث يمكن أن يحتوي القاموس على العديد من القواميس داخل معنى اخر (قاموس كبير بداخله قواميس صغيرة) وهذه الحالة تسمى بالقواميس المتداخلة.

يمكن انشائها بطريقتين الاولى هي ان القاموس يحتوي بداخله على ثلاث قواميس كالتالي:

```

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)
File Edit Format Run Options Window Help
nums = {"intnum":{1:"one",2:"two"}, "floatnum":{1:"1.0",2:"2.0"}, "complex":{1:"1+0j", 2:"2+0j"}}
print(nums)

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\hadeel\Desktop\int.pyw =====
{'intnum': {1: 'one', 2: 'two'}, 'floatnum': {1: '1.0', 2: '2.0'}, 'complex': {1: '1+0j', 2: '2+0j'}}
>>>
>>>
>>> |

```

يمكن ترتيبها كالتالي بطريقة اوضح ويمكن قراءتها بسهولة :

```

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)
File Edit Format Run Options Window Help
nums = {
    "intnum": {
        1:"one",
        2:"two"
    },
    "floatnum": {
        1:"1.0",
        2:"2.0"
    },
    "complex": {
        1:"1+0j",
        2:"2+0j"
    }
}
print(nums)

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.
1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\hadeel\Desktop\int.pyw =====
{'intnum': {1: 'one', 2: 'two'}, 'floatnum': {1: '1.0', 2: '2.0'},
 'complex': {1: '1+0j', 2: '2+0j'}}
>>> |

```



اعتقد ان فكرة القواميس المداخلة اصبحت اكثراً وضوحاً. اما الطريقة الثانية دمج ثلاث قواميس موجودة في قاموس واحد كالمثال التالي:

```

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)
File Edit Format Run Options Window Help
intnum={
    1:"one",
    2:"two"
}
floatnum={
    1:"1.0",
    2:"2.0"
}
complexnum={
    1:"1+0j",
    2:"2+0j"
}
nums = {
    "intnum":intnum,
    "floatnum":floatnum,
    "complexnum":complexnum
}
print(nums)

```

Python 3.8.2 Shell

```

File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw ===
=====
{'intnum': {1: 'one', 2: 'two'}, 'floatnum': {1: '1.0', 2: '2.0'}, 'complexnum': {1: '1+0j', 2: '2+0j'}}
>>> |

```

ولكن بقي لدينا موضوع آخر في القواميس وهو Constructor للقاموس حيث يمكن استخدام Constructor لإنشاء قاموس جديد:

ملاحظة صغيرة في المثال أدناه يجب الاستغناء عن النقطتين المتعامدتين (:) واستبدلها بالفارزة (,) نلاحظ ان بعد التنفيذ شكل القاموس طبيعي بنقطتين متعامدتين.

```

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)
File Edit Format Run Options Window Help
nums= dict(name= "ahmed", age= "25")
print(nums)

```

Python 3.8.2 Shell

```

File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: C:\Users\hadeel\Desktop\int.pyw ===
=====
{'name': 'ahmed', 'age': '25'}
>>> |

```

والآن انتهي من القاموس اتمنى ان يكون الشرح وافياً لكم



في اغلب المراجع والدورات يكتفون بهذا القدر من Data Types لكن يوجد ايضا من يضيفها الى انواع البيانات و لا ضير ان نستبين ما هو الثنائي binary ونتعرف كيفية تعامل البايثون معه.





٧. الثنائي Binary

ان النظام الثنائي مهم جدا في اي لغة برمجية لذا فان البايثون تستخدم دالة bin() لتحويل الاعداد الى وسائلي لشرح bytesarray و bytes و bin في مثال بسيط:

```

int.pyw - C:\Users\hadeel\Desktop Python 3.8.2 Shell
File Edit Format Run Options File Edit Shell Debug Options Window Help
a= 3
print(bin(a))
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb
tel) on win32
Type "help", "copyright", "credits" or
>>>
=====
RESTART: C:\Users\hadeel
>>>

```

نلاحظ اعلاه انه تم تحويل الرقم ٣ الى نظام الثنائي وتم اضافة حرف b اختصارا (binary). مثال اخر:

```

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020,
tel) on win32
Type "help", "copyright", "credits" or "license"
>>> bin(6)
'0b110'
>>>
>>>
```

```

والآن لنبدأ ب (Bytes) هي عبارة عن bytes بايتات متسلسلة، مفردة وثابتة (immutable) اي انها غير قابلة للتغيير لذا فأنها تعطي النتيجة بسلسلة بمدى يبدا من 0 الى 256 اي ( $0 \leq x \leq 256$ ). حيث يمكن تقديم bytes باستخدام الدالة () :



|                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  int.pyw - C:\Users\ha...<br>File Edit Format Run<br><pre>print(bytes(5))</pre> |  Python 3.8.2 Shell<br>File Edit Shell Debug Options Window Help<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb<br>tel) on win32<br>Type "help", "copyright", "credits" or<br>>>><br>===== RESTART: C:\Users\ha...<br>b'\x00\x00\x00\x00\x00'<br>>>> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

نلاحظ في المثال اعلاه عند استخدام الدالة bytes() انها اعطت خمس بايتات لنغير الخمسة الى خمسة عشر سيكون كالاتي :

|                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  int.pyw - C:\Users\ha...<br>File Edit Format Run<br><pre>print(bytes(15))</pre> |  Python 3.8.2 Shell<br>File Edit Shell Debug Options Window Help<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v<br>tel] on win32<br>Type "help", "copyright", "credits" or "license()" for more info<br>>>><br>===== RESTART: C:\Users\hadeel\Desktop\int.pyw ===<br>b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'<br>>>> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

هناك العديد من الدوال التابعة لـ bytes منها: bytes.decode(), bytes.count(), bytes.fromhex(), ..... الخ. ان البايتات bytes غالبا ما تكون صالحة للاستعمال مع البيانات ذات الترميز الشهير ASCII.



## يمكن ان فهر بسرعة ايضا الى Bytesarray

هي مشابهة للبيات لكنها مختلفة بانها قابلة للتغيير وليس ثابتة كالبيات حيث ان المثال ادناه يوضح هذه الدالة:

|                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>int.pyw - C:\Users\had</b><br><pre> File Edit Format Run x = 6 arr = bytearray(x) print(arr) </pre> | <b>Python 3.8.2 Shell</b><br><pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\int.pyw ===== bytearray(b'\x00\x00\x00\x00\x00\x00') &gt;&gt;&gt; ... </pre> |
|--------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Python 3.8.2 Shell</b><br><pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information &gt;&gt;&gt; b= bytearray(12) &gt;&gt;&gt; print(b) bytearray(b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00') &gt;&gt;&gt; print(type(b)) &lt;class 'bytearray'&gt; &gt;&gt;&gt; </pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

حيث ان هناك العديد من الدوال التابعة لها منها: .....الخ ( bytearray.index),( bytearray.find),( bytearray.count ) , ( bytearray.hex),( bytearray.fromhex )



**الآن انتهي من انواع البيانات في البايثون**



## Python Operators

### العمليات في البايثون

تعرف العمليات على أنها رموز خاصة تقوم بإجراء العمليات الحسابية أو المنطقية. بحيث يجب أن نعلم ان القيمة التي يعمل عليها عامل التشغيل تسمى المعامل. وان العمليات في البايثون مهمة وضرورية جداً لذا يجب ان نتعرف على الانواع المختلفة وكيفية استخدامها في python

تقسم العمليات الى :

- **العمليات الحسابية (Arithmetic operators):** يتم استخدام العمليات الحسابية مع القيم(values) الرقمية لإجراء العمليات الحسابية الشائعة وسيكون المثال التالي موجز للعمليات الرياضية المعروفة :

|                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)  File Edit Format Run Options Window Help x= 6 y= 4 print(x+y)      # عملية الجمع print(x-y)      # عملية الطرح print(x*y)      # عملية الضرب print(x/y)      # عملية القسمة print(x%y)      # باقي القسمة print(x**y)     # عملية الاس Exponentiation print(x//y)      # تقسيم ينتج عنه عدد صحيح </pre> | <pre> Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2021, 21:24:12) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw 10 2 24 1.5 2 1296 1 &gt;&gt;&gt; </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

في المثال اعلاه هناك سبعة انواع من العمليات الحسابية الأساسية وهي متواجدة في أغلب اللغات البرمجية.

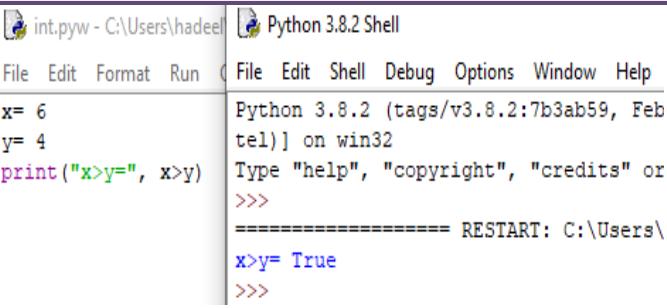
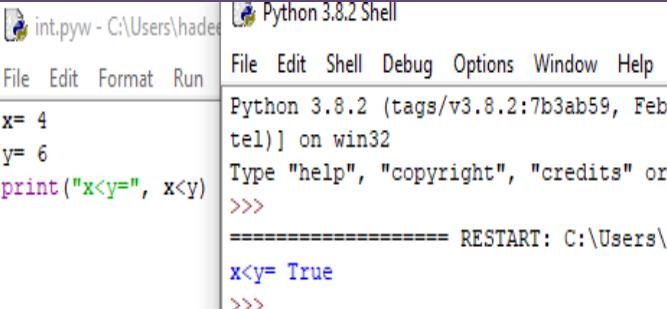
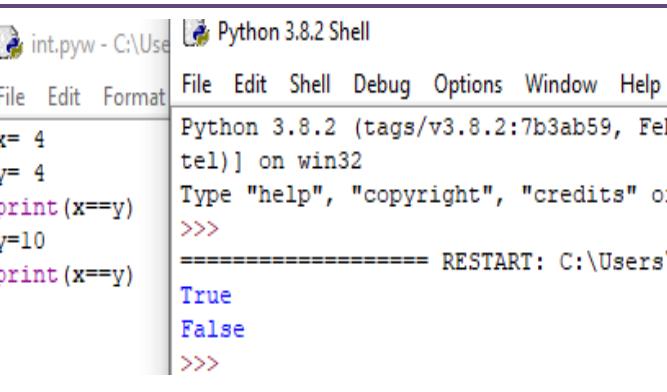


## • عمليات المقارنة (Comparison operators)

المقارنة هي بين شيئين متماثلين وتكون النتيجة دائمًا بصح (True) أو خطأ (False) بحسب الشرط.

باختصار ان عمليات المقارنة تستخدم عوامل المقارنة لمقارنة القيم

وهي الجدول ادناه العمليات (operators) وامثلة عن كيفية تمثيلها في البايثون :

| مثال في البايثون                                                                                                                                 | operator معنى ال operator                                                                                  | العمليات operators |
|--------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|--------------------|
|  <pre>x= 6 y= 4 print("x&gt;y=", x&gt;y) x&gt;y= True</pre>     | <b>أكبر من</b><br><b>إذا كانت القيمة</b><br><b>الأيسر أكبر من</b><br><b>اليمن فالتنتيجة</b><br><b>True</b> | >                  |
|  <pre>x= 4 y= 6 print("x&lt;y=", x&lt;y) x&lt;y= True</pre>    | <b>صغر من</b><br><b>إذا كانت القيمة</b><br><b>الأيسر صغر من</b><br><b>اليمن فالتنتيجة</b><br><b>True</b>   | <                  |
|  <pre>x= 4 y= 4 print(x==y) y=10 print(x==y) True False</pre> | <b>التساوي</b><br><b>إذا كان المعاملين</b><br><b>متساوين فان</b><br><b>النتيجة</b><br><b>True</b>          | ==                 |



|                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                        |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|--------|
| <pre> int.pyw - C:\Users File Edit Format x= 4 y= 4 print(x&gt;=y) y=10 print(x&gt;=y) </pre> <p>Python 3.8.2 Shell</p> <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, F tel) on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\User True False &gt;&gt;&gt; </pre>       | <p><b>أكبر او يساوي من</b><br/> <b>إذا كانت القيمة الأيسر أكبر او</b><br/> <b>تساوي القيمة من</b><br/> <b>اليمين فالنتيجة True</b></p> | $\geq$ |
| <pre> int.pyw - C:\Users File Edit Format x= 4 y= 4 print(x&lt;=y) y=10 print(x&lt;=y) </pre> <p>Python 3.8.2 Shell</p> <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Fe tel) on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\Users\1 True True &gt;&gt;&gt; ...</pre> | <p><b>صغر او يساوي من</b><br/> <b>إذا كانت القيمة الأيسر اصغر او</b><br/> <b>تساوي القيمة من</b><br/> <b>اليمين فالنتيجة True</b></p>  | $\leq$ |



• **العمليات المنطقية(Logical operators):** يتم استخدام العمليات التشغيل المنطقية لدمج العبارات الشرطية. العمليات المنطقية هي : and , or , not ويمكن تمثيلها كالتالي:

| مثال في البايثون                                      | عملها                                                                                                                                                                               | العملية                                                               |
|-------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <br><pre>x = 6 print(x &gt; 4 and x &lt; 10)</pre>    | <br><pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel)] on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\Users\had True &gt;&gt;&gt; &gt;&gt;&gt;</pre> | تكون النتيجة<br>اذا كان True<br>المعاملين صحيحين<br>او الشرطين صحيحين |
| <br><pre>x = 4 print(x &gt; 3 or x &lt; 3)</pre>      | <br><pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, tel)] on win32 Type "help", "copyright", "credit &gt;&gt;&gt; ===== RESTART: C:\U True &gt;&gt;&gt;</pre>                               | تكون النتيجة<br>اذا كان True<br>احد المعاملين او احد الشرطين صحيحين   |
| <br><pre>x = 4 print(not(x &gt; 3 or x &lt; 3))</pre> | <br><pre>Python 3.8.2 (tags/v3.8.2:7b3ab59 tel)] on win32 Type "help", "copyright", "credi &gt;&gt;&gt; ===== RESTART: C:\U False &gt;&gt;&gt;</pre>                                | عكس العملية<br>معنی اذا كان False<br>المعامل فالنتيجة<br>True         |



|                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  int.pyw - C:\Users\hadeel\Desktop\int.pyw                                 |  Python 3.8.2 Shell                                                                                                                                                                                                                         |
| <pre> File Edit Format Run Options Window x = True y = False print('x and y is',x and y)  print('x or y is',x or y)  print('not x is',not x)         </pre> | <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:00) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw x and y is False x or y is True not x is False &gt;&gt;&gt;         </pre> |

• العمليات لإسناد القيم (Assignment operators): تُستخدم لتعيين قيم للمتغيرات

| مثال في الباليثون                                                                                                            | عملها                                                                                                                                                                                                                                                                      | العملية                                 |
|------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
|  int.pyw - C:\User\hadeel\Desktop\int.pyw |  Python 3.8.2 Shell                                                                                                                                                                     | اسناد قيمة الى متغير<br>=               |
| <pre> File Edit Format x = 4 print(x)         </pre>                                                                         | <pre> File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:00) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw 4 &gt;&gt;&gt;         </pre> | تمثل هذه العملية كالاتي:<br>$x = x + 3$ |



|  |  |                                                  |       |
|--|--|--------------------------------------------------|-------|
|  |  | <b>تمثيل هذه العملية كالاتي:</b><br>$x = x - 3$  | $-=$  |
|  |  | <b>تمثيل هذه العملية كالاتي:</b><br>$x = x * 3$  | $*=$  |
|  |  | <b>تمثيل هذه العملية كالاتي:</b><br>$x = x / 3$  | $/=$  |
|  |  | <b>تمثيل هذه العملية كالاتي:</b><br>$x = x \% 3$ | $\%=$ |



|  |  |                                       |     |
|--|--|---------------------------------------|-----|
|  |  | تمثل هذه العملية كالاتي<br>x = x // 3 | //= |
|  |  | تمثل هذه العملية كالاتي<br>x = x ** 3 | **= |
|  |  | تمثل هذه العملية كالاتي<br>x = x & 3  | &=  |
|  |  | تمثل هذه العملية كالاتي<br>x = x   3  | =   |



|                                   |                                                                                                                                                                                                                                                                                                                   |                                           |            |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|------------|
| int.pyw - C:\Users\Hadeel M.Taher | Python 3.8.2 Shell<br>File Edit Shell Debug Options Window<br>Python 3.8.2 (tags/v3.8.2:7b3ab5, Feb 25 2020, 23:08:13) [MSC v.1916 64 bit (AMD64)] on win32<br>Type "help", "copyright", "credits" or "license" for more information.<br>>>><br>===== RESTART: C:\Users\Hadeel M.Taher\int.pyw =====<br>5<br>>>>  | تمثل هذه العمليات كالتالي<br>$x = x^3$    | $\wedge =$ |
| int.pyw - C:\Users\Hadeel M.Taher | Python 3.8.2 Shell<br>File Edit Shell Debug Options Window<br>Python 3.8.2 (tags/v3.8.2:7b3ab5, Feb 25 2020, 23:08:13) [MSC v.1916 64 bit (AMD64)] on win32<br>Type "help", "copyright", "credits" or "license" for more information.<br>>>><br>===== RESTART: C:\Users\Hadeel M.Taher\int.pyw =====<br>0<br>>>>  | تمثل هذه العمليات كالتالي<br>$x = x >> 3$ | $>>=$      |
| int.pyw - C:\Users\Hadeel M.Taher | Python 3.8.2 Shell<br>File Edit Shell Debug Options Window<br>Python 3.8.2 (tags/v3.8.2:7b3ab5, Feb 25 2020, 23:08:13) [MSC v.1916 64 bit (AMD64)] on win32<br>Type "help", "copyright", "credits" or "license" for more information.<br>>>><br>===== RESTART: C:\Users\Hadeel M.Taher\int.pyw =====<br>40<br>>>> | تمثل هذه العمليات كالتالي<br>$x = x << 3$ | $<<=$      |

## • العمليات (Identity Operators):

(not) فقط و تعمل للمقارنة وارجاع قيمة True او False, تعتمد تلك المقارنة على

object

حيث انها يجب ان تتساوى بالقيم وتتساوى بـ object لتعطي نتيجة true, بمعنى ادق انه يتم استخدامها للتحقق من وجود قيمتين (أو متغيرات) على نفس الجزء من الذاكرة. وهذا لا يعني ان اي متغيران متساويان أنهما متطابقان. والمثال ادناه سيوضح عملية المقارنة:



|                                                                                                                                                   |                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\...                                                                                                                     | Python 3.8.2 Shell                                                                                                                                                                                                                                                                    |
| <pre> File Edit Format Run Options x= ["python", "c++"] y= ["python", "c++"] a= x print( a is x) print( a is y) print( x is not y)         </pre> | <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\int.pyw True False True &gt;&gt;&gt;         </pre> |

مثال اخر:

|                                                                                                                       |                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\...                                                                                         | Python 3.8.2 Shell                                                                                                                                                                                                                                                                |
| <pre> File Edit Format Run Options a = 5 b = 5 x = [1,2,3] y = [1,2,3] print(a is not b) print(x is y)         </pre> | <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\int.pyw False False &gt;&gt;&gt;         </pre> |

نلاحظ انه في الطباعة الاولى للمثال اعلاه اعطي قيمة خطأ لأن الرقم خمسة في b, هو ذاته بنفس المكان في الذاكرة لذلك يجب ان تكون النتيجة True . اما في القائمة(list) فهي حالة مختلفة وان تشابهت العناصر داخل القائمتين الا ان لكل واحدة منهم حجز مختلف في الذاكرة

- **عمليات العضوية (Membership Operators):** وان كان التعريف للكلمة غير دقيق لكنه يوضح انها عمليات اختبار عضوية حيث تختبر هل هذا العنصر موجود في قائمة او حرف في جملة .... الخ



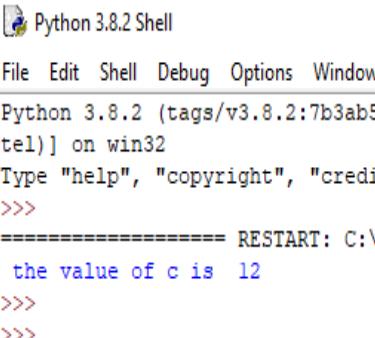
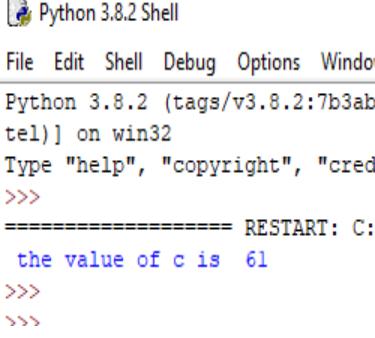
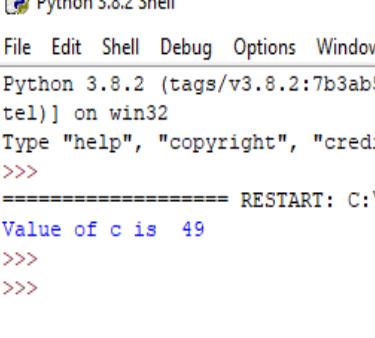
|                                                                             |                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel                                                   | Python 3.8.2 Shell                                                                                                                                                                                                                                                                                               |
| <pre>File Edit Format Run x = [1,2,3] print(1 in x) print(2 not in x)</pre> | <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\int.pyw ===== True False &gt;&gt;&gt; ...</pre> |

|                                                                                                     |                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\D                                                                         | Python 3.8.2 Shell                                                                                                                                                                                                                                                                                                              |
| <pre>File Edit Format Run Op x = "python" print('t' in x) print('z' in x) print('o' not in x)</pre> | <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\int.pyw ===== True False False &gt;&gt;&gt; &gt;&gt;&gt;</pre> |

|                                                                                                 |                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\                                                                      | Python 3.8.2 Shell                                                                                                                                                                                                                                                                                                 |
| <pre>File Edit Format Run O x = {1,2,3,4,5} print(1 in x) print(6 in x) print(3 not in x)</pre> | <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\int.pyw ===== True False False &gt;&gt;&gt;</pre> |



## • عمليات (Bitwise operators): تعمل على تحويل المعامل الى سلسلة من الارقام الثنائية (تستخدم لمقارنة الأرقام (الثنائية)) وهي كالتالي:

| Bitwise عمليات | عملها                                                                                        | مثال                                                                                                                                                                                                                             |
|----------------|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| &              | تحويل الارقام الى النظام الثنائي واجراء عملية and جميع الاحتمالات صفر ماعدا 1 , 1 تعطي 1     |  <pre>a = 60      # 60 = 0011 1100 b = 13      # 13 = 0000 1101 c = 0 c = a &amp; b    # 12 = 0000 1100 print (" the value of c is ", c)</pre> |
|                | تحويل الارقام الى النظام الثنائي واجراء عملية or جميع الاحتمالات 1 ماعدا 0,0 تعطي 0          |  <pre>a = 60      # 60 = 0011 1100 b = 13      # 13 = 0000 1101 c = 0 c = a   b    # 61 = 0011 1101 print (" the value of c is ", c)</pre>    |
| ^              | تحويل الارقام الى النظام الثنائي واجراء عملية ^ تعطي الارقام الثنائية المشابهة 0 والمختلفة 1 |  <pre>a = 60      # 60 = 0011 1100 b = 13      # 13 = 0000 1101 c = 0 c = a ^ b    # 49 = 0011 0001 print ("Value of c is ", c)</pre>        |



|                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                   |                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|-----------------|
|  <pre>a = 60      # 60 = 0011 1100 c= 0 c = a &lt;&lt; 2    # 240 = 1111 0000 print ("Value of c is ", c)</pre>  |  <pre>Python 3.8.2 (tags/v3.8.2:7b3ab3f, Feb 25 2020, 22:46:08) [PyPy 6.3.0 (6.3.0+dev-149-gd3f9f8) on win32] Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw Value of c is 240 &gt;&gt;&gt; &gt;&gt;&gt;</pre> | <p>تسمى<br/>right shift<br/>يتم نقل قيمة<br/>المعاملات اليسرى<br/>إلى اليسار بعدد<br/>البيتات التي يحددها<br/>المعامل الأيمن.</p> | <p>&gt;&gt;</p> |
|  <pre>a = 60      # 60 = 0011 1100 c= 0 c = a &gt;&gt; 2    # 15 = 0000 1111 print ("Value of c is ", c)</pre> |  <pre>Python 3.8.2 (tags/v3.8.2:7b3ab3f, Feb 25 2020, 22:46:08) [PyPy 6.3.0 (6.3.0+dev-149-gd3f9f8) on win32] Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw Value of c is 15 &gt;&gt;&gt;</pre>             | <p>تسمى<br/>left shift<br/>يتم نقل قيمة<br/>المعاملات اليسرى<br/>لليمين بعدد البيتات<br/>التي يحددها المعامل<br/>الأيمن.</p>      | <p>&lt;&lt;</p> |

We finished Python Operators 😊



## Making Choices and Decisions

### صنع الاختيارات والقرارات

الآن بعد ان تعلمنا اساسيات البايثون يمكن جعل البرامج اكثرا ذكاء من خلال اتخاذ القرارات وتشمل هذه الفقرة كل من :

- If statement
- For loop
- While loop

تعرف ايضا بـ (Conditionals and loops) هذه الشروط تعمل على السيطرة (control flow) على تدفق البيانات في البرنامج . بالإضافة إلى ذلك ، سنلقي نظرة أيضاً على المحاولة والاستثناء وبيان ما يجب على البرنامج القيام به عند حدوث خطأ. اذا نبدأ بـ If statement

#### If statement -

لقد لاحظنا مسبقا ان البايثون تدعم الشروط المنطقية المعتادة من الرياضيات اثناء شرحنا لها في العمليات المنطقية. لكن الان نستطيع استخدام الشرط if ووضع عبارة اذا تحقق الشرط تنفذ تلك العبارة .

لنتعرف على تركيبة if statement في البايثون :

if test expression:

    Statement of if

elif test expression:

    Statement of elif

else:

    Statement of else



سنتعلم ايضاً كيفية تمثيل if statement في البايثون خطوة بخطوة:

```

int.pyw - C:\Users\hadeel\Desktop\int.pyw (3)
File Edit Format Run Options Window
a = 60
b = 40
if a> b:
 print(" a is bigger than b")
Python 3.8.2 (tags/v3.8.2:7b3ab5
tel) on win32
Type "help", "copyright", "credi
>>>
=====
RESTART: C:\V
a is bigger than b
>>>
>>>

```

في هذا المثال اعلاه ، لقد استخدمنا متغيرين وهما a و b ، يتم استخدامهما كجزء من if statement لاختبار ما إذا كانت a أكبر من b. بما أن 60 ، و b هي 40 ، فإننا نعلم أن 60 أكبر من 40 ، لذا فإننا نطبع على الشاشة "a أكبر من b".

لكن لنلاحظ قليلاً في المثال اعلاه ان هناك مسافة whitespace (whitespace) ما بعد ال if حيث ان تلك المسافة ما قبل عبارة print هي لتنفيذ شرط if . في بعض اللغات تكون ما بعد if اقواس المترجة { } وما بينهما العبارات التي نريد تنفيذها لكن في البايثون لا توجد اقواس وإنما وضع مسافة فقط

لنجرِّب في حال لم نضع تلك المسافة ماذا سيحصل :

```

int.pyw - C:\Users\hadeel\Desktop\int.
File Edit Format Run Options Window
a = 60
b = 40
if a> b:
print(" a is bigger than b")

```

SyntaxError

expected an indented block

OK



لم يتم تنفيذ الشرط لأنه لا توجد مسافة قبل الطباعة . لكن من جهة اخرى يمكن ان يكون (الطباعة) بجانب الشرط كالاتي:

|                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre> File Edit Format Run Options Window Help a = 40 b = 60 if b &gt; a: print("b is greater than a")         </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:08:11) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw b is greater than a &gt;&gt;&gt;         </pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

والآن في حال ان الشرط الاول لل if غير صحيح لذا يمكن ان نستخدم شرط اخر من خلال elif وهي طريقة البايثون لقول "إذا لم تكن الشروط السابقة صحيحة، فجرب هذا الشرط".

|                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre> File Edit Format Run Options Window Help a = 60 b = 40 if b &gt; a:     print("b is greater than a") elif b &lt; a:     print("b is smaller than a")         </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:08:11) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw b is smaller than a &gt;&gt;&gt; &gt;&gt;&gt;         </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



وفي حال عدم وجود شرط صحيح يمكن تنفيذه ففي هذه الحالة نستخدم Else:

|                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre> File Edit Format Run Options Window Help a = 60 b = 60 if b &gt; a:     print("b is greater than a") elif b &lt; a:     print("b is smaller than a") else:     print("a and b are equal")         </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:08:11) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw a and b are equal &gt;&gt;&gt; &gt;&gt;&gt;         </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



يوجد ايضاً ما يعرف ب Nested If statement والتي يجب ان تذكرها وهي عبارة عن if statement بداخلها if statement آخر :

|                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw (3) <pre> File Edit Format Run Options Window n = 10 if n &gt;= 0:     if n==0:         print("Zero")     else:         print("Positive number") else:     print("Negative number")         </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab tel) on win32 Type "help", "copyright", "cre &gt;&gt;&gt; ===== RESTART: C: Positive number &gt;&gt;&gt; &gt;&gt;&gt; &gt;&gt;&gt;         </pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

يمكن ان تتضمن if statement العمليات المنطقية (logical operator) حيث انها تدمج مع العبارات الشرطية ومثال على ذلك:

|                                                                                                                                                                                                     |                                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre> File Edit Format Run Options Window Help a= 60 b= 40 c= 20 if a&gt; b and a&gt; c:     print(" a is greater than other num")         </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab tel) on win32 Type "help", "copyright", "cre &gt;&gt;&gt; ===== RESTART: C: a is greater than other num &gt;&gt;&gt; ...         </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2) <pre> File Edit Format Run Options Window Help a= 60 b= 40 c= 20 if b&gt; a or b &gt; c:     print(" At least one of the conditions is True ")         </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 1 tel) on win32 Type "help", "copyright", "credits" or ' &gt;&gt;&gt; ===== RESTART: C:\Users\ha At least one of the conditions is True &gt;&gt;&gt; &gt;&gt;&gt;         </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



The screenshot shows two windows. On the left is PyCharm with a script named 'int.pyw' containing the following code:

```
a= 60
b= 40
c= 20
if (not(b>a)):
 print(" a is greater than b ")
```

On the right is the Python 3.8.2 Shell window showing the output of running the script:

```
Python 3.8.2 (tags/v3.8.2:7b3ab5
tel) on win32
Type "help", "copyright", "credi
>>>
=====
RESTART: C:\a
a is greater than b
>>>
>>>
...
```

## For Loops -

تُستخدم الحلقة التكرارية ( for ) في لغة البايثون للتكرار وذلك عبر تسلسل (قائمة، مجموعات، set ، list ، string) أو أي بيانات أخرى قابلة للتكرار. لذا ان تركيبة for في البايثون كالتالي:

for value in sequence:

statement of for

نلاحظ ان المتغير value يأخذ قيمة العنصر داخل المتسلاسل في كل تكرار. حيث تستمرة الحلقة التكرارية حتى نصل إلى العنصر الأخير في التسلسل .

The screenshot shows two windows. On the left is PyCharm with a script named 'int.pyw' containing the following code:

```
x = [1,2,3,4,5,6]
for i in x:
 print(i)
```

On the right is the Python 3.8.2 Shell window showing the output of running the script:

```
Python 3.8.2 (tags/v3.8.2:7b3ab5
tel) on win32
Type "help", "copyright", "credits" or
>>>
=====
RESTART: C:\Users\1
1
2
3
4
5
6
>>>
```



اعلاه نشاهد اتنا لم نعرف المتغير `x` لأن الباليثون من اسهل اللغات الموجودة حاليا فانت لا تحتاج لتعريف المتغيرات قبل استخدامها . ان البرنامج عمل على طباعة جميع الارقام الموجودة بالقائمة وعندما وصل الى النهاية انتهى البرنامج .

```

int.pyw - C:\Users\hadeel\Desktop\int.pyw
File Edit Format Run Options
x = ("python", "c++", "c")
for i in x:
 print(i)

Python 3.8.2 Shell
File Edit Shell Debug Options Window
Python 3.8.2 (tags/v3.8.2:7b3ab
tel) on win32
Type "help", "copyright", "cred
>>>
=====
RESTART: C:
python
c++
c

```

```

int.pyw - C:\Users\hadeel\Desktop\int.pyw
File Edit Format Run Options
x = "python"
for i in x:
 print(i)

Python 3.8.2 Shell
File Edit Shell Debug Options Window
Python 3.8.2 (tags/v3.8.2:7b3ab5!
tel) on win32
Type "help", "copyright", "credi
>>>
=====
RESTART: C:\I
P
Y
T
H
O
N
>>>

```

يمكن استخدام دالة المدى `range()` داخل الحلقة التكرارية حيث ترجع الدالة `range()` سلسلة من الأرقام ، تبدأ من 0 بشكل افتراضي ، وتزداد بمقدار 1 (افتراضياً) ، وتنتهي بالرقم المحدد ما بين قوسى الدالة `range()`

```

int.pyw - C:\Users\hadeel\Desktop\int.pyw
File Edit Format Run Options
for i in range(5):
 print(i)

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb
tel) on win32
Type "help", "copyright", "credits" or
>>>
=====
RESTART: C:\Users\h
0
1
2
3
4

```



يمكن ان نحدد دالة المدى بقيمة بداية مثلا نجعلها تبدا من ٢ بدلا من الصفر وقيمة نهاية وهي ٨ :

```
int.pyw - C:\Users\hadeel\Desktop\int.pyw
for i in range(2, 8):
 print(i)

Python 3.8.2 Shell
File Edit Shell Debug Options Window
Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32
Type "help", "copyright", "credit"
>>>
===== RESTART: C:\U
2
3
4
5
6
7
>>>
```

في المثالين السابقين كانت قيمة الزيادة بين رقم والآخر هو ١ ولكن هل يمكن تغيير هذه القيمة ؟

الجواب : نعم يمكن ذلك حيث نستطيع ان تكون قيمة الزيادة ٢ و ٤ او .... الخ باي قيمة داخل دالة المدى ويمكننا تمثيل ذلك كالتالي:

```
int.pyw - C:\Users\hadeel\Desktop\int.pyw
for i in range(2, 20, 4):
 print(i)

Python 3.8.2 Shell
File Edit Shell Debug Options Window
Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32
Type "help", "copyright", "credit"
>>>
===== RESTART: C:\U:
2
6
10
14
18
>>>
~~~
```





هناك ما يدعى بالتكرار المتداخل (Nested Loops) وهي عبارة عن for loop بداخلها another for loop حيث

تم العملية من خلال تنفيذ loop الداخلية لكل iteration في loop الخارجية

ولسهولة فهم اسلوب عمل الحلقات التكرارية المتداخلة من خلال المثال أدناه:

|                                                                                                                                                                            |                                                                                                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw <pre> File Edit Format Run Options x = ["python", "c++"] for i in range(2,10,4):     for n in x:         print(i,n)       </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab5 tel) on win32 Type "help", "copyright", "credi &gt;&gt;&gt; ===== RESTART: C:\\ 2 python 2 c++ 6 python 6 c++ &gt;&gt;&gt;       </pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

لقد تم تنفيذ البرنامج بالفعل من خلال تنفيذ اللوب الداخلي بالكامل لكل لوب خارجي اي تم تنفيذ القائمة بالكامل لكل رقم بالمدى.

مثال اخر لتوضيح تداخل الحلقات التكرارية :

|                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw <pre> File Edit Format Run Options W m =[1,2,3] n =[1,2,3] for i in m:     for j in n:         print("(",i,j,")")       </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Fe tel) on win32 Type "help", "copyright", "credits" ( &gt;&gt;&gt; ===== RESTART: C:\User: ( 1 1 ) ( 1 2 ) ( 1 3 ) ( 2 1 ) ( 2 2 ) ( 2 3 ) ( 3 1 ) ( 3 2 ) ( 3 3 ) &gt;&gt;&gt; ```       </pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



يمكننا ان نضم if داخل for loop حيث ينفذ الشرط ان كان صحيح في كل دورة لـ

|                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  int.pyw - C:\Users\hadeel\Desktop\int.pyw (3)                                                                                        |  Python 3.8.2 Shell                                                                                                                                                                   |
| <pre> File Edit Format Run Options Window m =[1,-2,3,0,-4] for i in m:     if i &gt; 0:         print(i,"is a positive")     elif i==0:         print(i,"is a zero")     else:         print(i,"is a negative") </pre> | <pre> File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab tel) on win32 Type "help", "copyright", "cred &gt;&gt;&gt; ===== RESTART: C: 1 is a positive -2 is a negative 3 is a positive 0 is a zero -4 is a negative &gt;&gt;&gt; &gt;&gt;&gt; </pre> |

المسافات ضرورية جدا (كالتي بين الـ if و for ) يجب الانتباه لها لأنها قد تكون سبب لعدم تنفيذ برنامجك.

### الحلقة التكرارية (While Loops) -

باستعمال حلقة while نستطيع تنفيذ مجموعة من العبارات طالما أن تعبير الاختبار (الشرط) صحيح

ان التركيبة الاساسية لحلقة while كالاتي:

while test expression:

Body of while

في (while loops) الحلقة التكرارية يتم فحص التعبير اولا وفي حال انه صحيح (true) تستخدم while loops وهكذا يستمر بالتنفيذ الى حين ان يتم تقييم التعبير بخطأ (false). هذه الحلقة التكرارية عادة ما تكون هناك مسافات يجب الانتباه اليها ايضا وهي مهمة جدا لضمان التنفيذ بدون اي اخطاء.



|                                                                                                                      |                                                                                                                                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>int.pyw - C:\Users\hadeel\Des File Edit Format Run Options x = 2 while x &lt;= 10:     print(x)     x+= 2</pre> | <pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC tel]) on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Des\int.pyw = 2 4 6 8 10 &gt;&gt;&gt;</pre> |
|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

في while loop يجب ان نعطي قيمة بدائية للمتغير فضلا عن ذلك انها ستستمر بالتنفيذ الا ان يكتمل الشرط وهو ان X يجب ان يكون اصغر او يساوي 10 حينها سيتوقف البرنامج .

في حال عدم اعطاء قيمة للمتغير سيكون هناك خطأ وسيظهر لك على شاشة التنفيذ :

|                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>int.pyw - C:\Users\hadeel\Des File Edit Format Run Options while x &lt;= 10:     print(x)     x+= 2</pre> | <pre>Python 3.8.2 Shell File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC tel]) on win32 Type "help", "copyright", "credits" or "license()" for more infor &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw = Traceback (most recent call last):   File "C:\Users\hadeel\Desktop\int.pyw", line 1, in &lt;module&gt;     while x &lt;= 10: NameError: name 'x' is not defined &gt;&gt;&gt; &gt;&gt;&gt; &gt;&gt;&gt;</pre> |
|----------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

مصطلاح جديد يجمع ما بين الحلقة التكرارية While loop with else والتي تعودنا على استخدامها في if .



في if الشرطية استخدمنا else في حال كان الشرط غير صحيح نستعين بـ else لكي تعطينا عبارة كالطباعة مثلاً. أما هنا في while تستخدم بنفس القالب اي في حال ان الشرط غير صحيح يمكن استخدام else لتشغيل كتلة من التعليمات البرمجية مرة واحدة

|                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)<br><pre> File Edit Format Run Options Window Help x= 2 while x &lt; 10:     print(x)     x+= 2 else:     print("x is no longer less than 10") </pre> | Python 3.8.2 Shell<br><pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2021, 21:21:18) [MSC v.1916 64 bit (AMD64)] Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\int.pyw ===== 2 4 6 8 x is no longer less than 10 &gt;&gt;&gt; ... </pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

والآن انتهي من الاختبارات والقرارات .

**لكن قبل كل ان نختتم موضوع الاختبارات والقرارات يجب ان نكتب لك ملاحظات مهمة ارجو الاستفادة منها:**

- ان نهاية كل من if و for و while نقطتان متعدمتان (:) ويجب علينا ان لا ننساها ابداً لأن نسيانها سيعطيك خطأ في التنفيذ.
- المسافات ضرورية جداً حيث تم ذكرها أكثر من مرة لأهميتها لتنفيذ برنامجك بسهولة.
- يمكن ان تداخل ال if و while مع for و continue في الشرطية بسهولة.
- لا توجد فارزة منقوطة اثناء كتابة الجمل البرمجية في البيشون على عكس C++
- هناك ما يدعى ب break و continue و for يستخدم في while واستخدامها سهل للغاية الاولى تستخدم لإيقاف الحلقة حتى لو كان الشرط صحيحاً اما continue فتستخدم لإيقاف التكرار الحالي ، والاستمرار مع التالي . ومثال ذلك :



|                                                                                                          |                                                                                                                                                                                                     |
|----------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel                                                                                | Python 3.8.2 Shell                                                                                                                                                                                  |
| File Edit Format Run<br><pre>num = [1,2,3] for x in num:     if x == 2:         break     print(x)</pre> | File Edit Shell Debug Options Window Help<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\int.pyw ===== 1 >>> >>> |

|                                                                                                             |                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel                                                                                   | Python 3.8.2 Shell                                                                                                                                                                           |
| File Edit Format Run<br><pre>num = [1,2,3] for x in num:     if x == 2:         continue     print(x)</pre> | File Edit Shell Debug Options Window<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\int.pyw ===== 1 3 >>> |

|                                                                                                                      |                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Des                                                                                        | Python 3.8.2 Shell                                                                                                                                                                                    |
| File Edit Format Run Options<br><pre>x= 2 while x &lt; 10:     print(x)     if x == 4:         break     x+= 2</pre> | File Edit Shell Debug Options Window Help<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Des\int.pyw ===== 2 4 >>> |

|                                                                                                                         |                                                                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| int.pyw - C:\Users\hadeel\Des                                                                                           | Python 3.8.2 Shell                                                                                                                                                                                       |
| File Edit Format Run Options<br><pre>x= 2 while x &lt; 10:     x+= 2     if x == 8:         continue     print(x)</pre> | File Edit Shell Debug Options Window Help<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or >>> ===== RESTART: C:\Users\hadeel\Des\int.pyw ===== 4 6 10 >>> |



عبارة المروor (pass statement) : تستخدم عبارة pass لمرور عبارات لتجنب ظهور الأخطاء . -

لننفذه ليكون فهم عملها أسهل للقارئ:

|                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)<br>File Edit Format Run Options Window Help<br><pre>x = ['p', 'a', 'y', 't', 'h', 'o', 'n'] for i in x:     pass</pre> |  Python 3.8.2 Shell<br>File Edit Shell Debug Options Window<br>Python 3.8.2 (tags/v3.8.2:7b3af8c, Feb 27 2020, 17:03:43) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information >>> ===== RESTART: C:/Users/hadeel/Desktop/int.pyw ===== >>> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

في برمجة البايثون عبارة pass تكون عبارة المروور عبارة خالية اي لا يوجد اي شيء لفعله لذا في المثال السابق فقط تم مرور جميع عناصر X (تم تنفيذ البرنامج بدون حدوث اي شيء)

|                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  int.pyw - C:\Users\hadeel\Desktop\int.pyw (3.8.2)<br>File Edit Format Run Options Window Help<br><pre>x = 10 y = 20 if y &gt; x:     pass</pre> |  Python 3.8.2 Shell<br>File Edit Shell Debug Options Window Help<br>Python 3.8.2 (tags/v3.8.2:7b3af8c, Feb 27 2020, 17:03:43) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information >>> ===== RESTART: C:/Users/hadeel/Desktop/int.pyw ===== >>> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

الملاحظة الأخيرة في حال تم الخطأ في احدى الخطوات واصبح لدينا infinite loop تكرار الى ما لا نهاية نستخدم CTRL+C keys للخروج منه.



## Python Try, Except

### الاستثناءات في بايثون

الاستثناءات في لغة بايثون يكون من خلال تعبير (try except finally). فكرة الاستثناء هي في حال وجد خطأ برمجي يمكن استثنائه في محاولة واحدة او اثنان او ثلاثة حسب احتياجك الى ان نصل ل finally وذلك لأنه في حال عدم الاستثناء قد يؤدي تنفيذ تلك الشيفرة إلى التسبب في حدوث أخطاء برمجية وبالتالي ايقاف البرنامج ان تركيبة الاستثناءات كالتالي:

```
try:  
    do something  
except:  
    do something else when an error occurs
```

|                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw (3.8.2)<br>File Edit Format Run Options Window Help<br><pre>while True:<br/>    try:<br/>        x= int(input("inter number int : "))<br/>        break<br/>    except ValueError:<br/>        print("that was no vaild int, try again please")</pre> |  Python 3.8.2 Shell<br>File Edit Shell Debug Options Window Help<br><pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020,<br/>tel) on win32<br/>Type "help", "copyright", "credits" or "license"<br/>&gt;&gt;&gt;<br/>===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw =====<br/>inter number int : 5.5<br/>that was no vaild int, try again please<br/>inter number int : p<br/>that was no vaild int, try again please<br/>inter number int : s<br/>that was no vaild int, try again please<br/>inter number int : 8<br/>&gt;&gt;&gt;</pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



|                                                                                                                                               |                                                                                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw |  Python 3.8.2 Shell                                                                                                                                                                     |
| File Edit Format Run Options Window<br><b>try:</b><br>print(m)<br><b>except:</b><br>print("An exception occurred")                            | File Edit Shell Debug Options Window<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32<br>Type "help", "copyright", "credits", "license" for more information.<br>>>><br>===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw =====<br>An exception occurred<br>>>> |

في هذا المثال ظهرت عبارة (except) بسبب ان (m) غير معرفة فكان من المفروض ان البرنامج يجب ان يظهر .except لوجود error

في حال وجود اكثرب من استثناء(except) حيث ان البرنامج سيقوم بطباعة عبارة واحدة اي يأخذ استثناء واحد فقط وفي المثال أدناه تم التعرف على ان الاسم خطأ بسبب المتغير غير معرف سلفاً في البداية.

|                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw                                           |  Python 3.8.2 Shell                                                                                                                                                                              |
| File Edit Format Run Options Window Help<br><b>try:</b><br>print(m)<br><b>except NameError:</b><br>print("Variable x is not defined")<br><b>except:</b><br>print("An exception occurred") | File Edit Shell Debug Options Window<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32<br>Type "help", "copyright", "credits", "license" for more information.<br>>>><br>===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw =====<br>Variable x is not defined<br>>>><br>>>> |

هناك أنواع مختلفة من الاستثناءات منها : ZeroDivisionError و TypeError و ValueError ، حيث تصف رسالة الخطأ نوع الاستثناء.



اما ال except try يتم تنفيذها بغض النظر عن error or not (هل هي try في حال تم تنفيذ finally او). فالنهاية finally (تنفذ).

|                                                                                                                           |                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>try:     print(x) except:     print("Something went wrong") finally:     print("The 'try except' is finished")</pre> | <pre>Python 3.8.2 Shell File Edit Shell Debug Options Window + Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw ===== Something went wrong The 'try except' is finished &gt;&gt;&gt; ...</pre> |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### وملخص موضوع الاستثناءات كالتالي:

Try هي المحاولة لاختبار تعليمات البرمجية معينة في حال ان فيها خطأ سنستخدم except والتي تتيح لك استثناء معالجة الخطأ وخيرا finally تتيح لك تنفيذ التعليمات البرمجية، بغض النظر عن نتيجة المحاولة او الاستثناء.



## Python Functions

### الدوال في البايثون

الدوال في البايثون هي مجموعة من العبارات البرمجية ذات صلة بالبرنامج وتؤدي مهمة معينة. ان الدوال تعمل على ترتيب البرنامج وتجعله اكثر تنظيما حيث تقسم البرنامج الى اجزاء منظمة لتجنب التكرار و يجعل الدالة قابلة لإعادة الاستخدام في حال احتجاجنا لتنفيذها.

لإنشاء دالة نستخدم (def) ببداية تعريف الدالة :

The screenshot shows two windows side-by-side. On the left is a code editor window titled 'tryexception.pyw - C:\Users\ha...'. It contains the following Python code:

```
File Edit Format Run Option
def my_function():
    print("Hello python")

my_function()
```

On the right is the 'Python 3.8.2 Shell' window, which displays the output of running the code. The output shows the Python interpreter's prompt (>>>), the code being run, and the resulting output:

```
File Edit Shell Debug Options Window
Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) ] on win32
Type "help", "copyright", "credits"
>>>
=====
RESTART: C:\Users\Hello python
>>>
```

```

يحتوي المثال اعلاه على المكونات التالية:

- الكلمة المفتاحية (def) التي تحدد بداية الدالة بعدها يجب ان يتبعها اسم تلك الدالة ثم الاقواس المدورة
- .) يكون داخل الاقواس المدورة ما يدعى بـ (Parameter) ويقوم تمرير قيم الى الدالة
- ثم توضع نقطتان (:). لوضع علامة على نهاية رأس الدالة.



- داخل هذه الدالة التي تم تعريفها مسبقا باسم my\_function تأتي عبارات ال statements قد تكون واحدة كما في مثالنا اعلاه وهي مجرد طباعة فقط او اكثرا من statements لكن يجب ان ننتبه الى المسافة البادئة وهي عادة ما تكون (4 مسافات).

- واخيرا استدعاء الدالة حيث نستخدم اسم الدالة متبوعاً بأقواس ولا توجد مسافة بادئة لأن الاستدعاء خارج الدالة.

مثال اخر عن الدالة :

|                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw <pre>File Edit Format Run Options Window def my_func():     x =6     print(" the value is: ",x) my_func()</pre> | Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window H Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 27 2020, 17:03:43) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw the value is: 6 &gt;&gt;&gt; &gt;&gt;&gt;</pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

لكن في حال تم وضع (Parameter) للدالة:

|                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw (3.8.2) <pre>File Edit Format Run Options Window Help def my_function(name):     print("my name is : " + name + ", i am from iraq") my_function("hadeel")</pre> | Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 27 2020, 17:03:43) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw my name is : hadeel, i am from iraq &gt;&gt;&gt; &gt;&gt;&gt;</pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



عند استدعاء الدالة نقوم بتمرير معلومات معينة من خلال الـ Parameter حيث نلاحظ في المثال اعلاه تم استدعاء الدالة من خلال تمرير الاسم الأول ، والذي سيتم استخدامه داخل الدالة لطباعة الاسم داخل جملة الطباعة .

لكن هل يمكن استدعاء الدالة اكثر من مرة ؟ وهل نستطيع تمرير معلومات مختلفة من خلال Parameter كما الاجابة هي نعم يمكننا استدعاء الدالة باي وقت نحتاجها ويمكن تمرير معلومات مختلفة في Parameter كما التالي :

حيث سنستدعي الدالة اكثر من مرة بمعلومات مختلفة في Parameter لنفس الدالة السابقة

|                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw (3.8.2)                                                    |  Python 3.8.2 Shell                                                                                                                                                                                                                              |
| <pre> File Edit Format Run Options Window Help  def my_function(name):     print("my name is : " + name + ", i am from iraq")  my_function("ahmed") my_function("mohammed") my_function("hadeel") </pre> | <pre> File Edit Shell Debug Options Window Help  Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 tel)] on win32 Type "help", "copyright", "credits" or "l &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\ my name is : ahmed, i am from iraq my name is : mohammed, i am from iraq my name is : hadeel, i am from iraq &gt;&gt;&gt; </pre> |

فضلا عن ذلك يمكن اضافة تعليقات توضيحية للمبرمج او المستخدم داخل الدالة :

|                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw (3.8.2)                                                                                                                                              |  Python 3.8.2 Shell                                                                                                                                                                                                                                          |
| <pre> File Edit Format Run Options Window Help  def my_function(name):     """     This function used to     define the person passed in as     a argument     """     print("my name is : " + name + ", i am from iraq")  my_function("ahmed") my_function("mohammed") my_function("hadeel") </pre> | <pre> File Edit Shell Debug Options Window Help  Python 3.8.2 (tags/v3.8.2:7b3ab59, Fe tel)] on win32 Type "help", "copyright", "credits" o &gt;&gt;&gt; ===== RESTART: C:\Users\had my name is : ahmed, i am from iraq my name is : mohammed, i am from iraq my name is : hadeel, i am from iraq &gt;&gt;&gt; &gt;&gt;&gt; &gt;&gt;&gt; </pre> |



وكلما تعلمنا مسبقا ان التعليقات لا تظهر في النتائج لأنها فقط تستخدمنا كتوضيح للمبرمج او المستخدم .

كما يمكن ان تحتوي الدالة على return وكما في اللغات البرمجية الاخرى حيث تعمل return على ارجاع قيمة معينة للدالة . مع الاخذ بنظر الاعتبار ان return قد تكون optional اي اختيارية وليس اجبارية لوضعها في كل دالة . سيكون تركيب الدالة مع return كالتالي :

```
def function_name( parameters ):
    function_suite
    return [expression]
```

|                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  tryexception.pyw - C:\Users\hadeel <pre>File Edit Format Run Options def my_function(x):     return 10 * x  print(my_function(2)) print(my_function(4)) print(my_function(6))</pre> |  Python 3.8.2 Shell <pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59 tel) on win32 Type "help", "copyright", "credit" &gt;&gt;&gt; ===== RESTART: C:\Users 20 40 60</pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

حيث نلاحظ ان الدالة سمحت بارجاع قيمة باستخدام عبارة الإرجاع لكن هل return تعمل على إرجاع قيمة واحدة ام يمكن ان تعمل على ارجاع اكثرا من قيمة ؟ الجواب : كلا لأنه مهما فعلنا فسوف يكون الارجاع قيمة واحدة كالاتي:



|                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>tryexception.pyw - C:\Users\hadeel\Documents\tryexception.pyw</pre> <pre>File Edit Format Run Options Window Help def mvar(x, y, z):     return x, y, z var= mvar(2,4,6) print (var)</pre> | <pre>Python 3.8.2 Shell</pre> <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Documents\tryexception.pyw ===== (2, 4, 6) &gt;&gt;&gt;</pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

قام بإرجاع قيمة واحدة على شكل tuple على الرغم أننا أدخلنا قيم ثلاثة مختلفة . فضلاً عن ذلك يمكن ارجاع جملة string وليس فقط ارقام وكما في المثال التالي:

|                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>tryexception.pyw - C:\Users\hadeel\Documents\tryexception.pyw</pre> <pre>File Edit Format Run Options Window Help def re_string():     return "python :)" print(re_string())</pre> | <pre>Python 3.8.2 Shell</pre> <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Documents\tryexception.pyw ===== python :) &gt;&gt;&gt;</pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

قد يتبرد إلى ذهنك في حال قمنا بوضع return مرتين في داخل الدالة function هل يتم تنفيذ الاثنان معاً ؟  
الحقيقة ان البايثون ستنفذ return واحدة فقط :

|                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>tryexception.pyw - C:\Users\hadeel\Documents\tryexception.pyw</pre> <pre>File Edit Format Run Options Window Help def re_string():     return "python :)"     return "python :)" print(re_string())</pre> | <pre>Python 3.8.2 Shell</pre> <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license" for more information. &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Documents\tryexception.pyw ===== python :) &gt;&gt;&gt;</pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



واخيرا يمكن ان تكون دالة الطباعة في الدالة مع وجود الارجاع (اي لا مانع من وجود الاثنين معا):

|                                                                                                                          |                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  tryexception.pyw - C:\Users\had        |  Python 3.8.2 Shell                                                                                                           |
| <pre>File Edit Format Run Option def re_string():     print("python :)")     return "python :)" print(re_string())</pre> | <pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\had python :) python :) &gt;&gt;&gt;</pre> |

يمكن ايضا استدعاء الدالة لأكثر من مرة (اي متداخلة) حيث يمكن تنفيذها باي وقت نحتاج اليها كما في المثال

ادناه:

|                                                                                                                                    |                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  tryexception.pyw - C:\Users\hadeel\Desktop\trye |  Python 3.8.2 Shell                                                                                                         |
| <pre>File Edit Format Run Options Window Help def my_function(x):     return 5 * x print(my_function(my_function(3)))</pre>        | <pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\trye 75 &gt;&gt;&gt;</pre> |

ومن الطبيعي القول اننا نستطيع وضع while و if و for التي تم شرحها فيما سبق وفي ادناه بعض الامثلة

لإضافتها داخل الـ function



|                                                                                                                                                                                                       |                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\D<br><pre> File Edit Format Run Options Window Help def abs_number(n):     if n &gt;= 0:         return n     else:         return -n print(abs_number(-4)) </pre> | Python 3.8.2 Shell<br><pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\tryexception.pyw ===== 4 &gt;&gt;&gt; ``` </pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                                  |                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\D<br><pre> File Edit Format Run Options Window Help def str_function():     for x in range(0,5):         if x == 3:             break         print(x)  str_function() </pre> | Python 3.8.2 Shell<br><pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\tryexception.pyw ===== 0 1 2 </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                   |                                                                                                                                                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\D<br><pre> File Edit Format Run Options Window Help def str_function():     x=1     while x in range(0,5):         x+=1         print(x) str_function() </pre> | Python 3.8.2 Shell<br><pre> File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\tryexception.pyw ===== 2 3 4 5 </pre> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



|                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw</pre> <pre>File Edit Format Run Options Window</pre> <pre>def pro():     i= 1     for i in range(100):         if i% 2 != 0:             print(i, " ", , end='')             i= i + 1 pro()</pre> | <pre>Python 3.8.2 Shell</pre> <pre>- □ X</pre> <pre>File Edit Shell Debug Options Window Help</pre> <pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32</pre> <pre>Type "help", "copyright", "credits" or "license()" for more information.</pre> <pre>&gt;&gt;&gt;</pre> <pre>===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw =====</pre> <pre>1 3 5 7 9 11 13 15 17 19 21 23 25</pre> <pre>27 29 31 33 35 37 39 41 43 45 47 49</pre> <pre>51 53 55 57 59 61 63 65 67 69 71 73</pre> <pre>75 77 79 81 83 85 87 89 91 93 95</pre> <pre>97 99</pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

يمكننا ايضا ادراج اي نوع من أنواع البيانات إلى دالة (سلسلة، رقم، قائمة، قاموس ، إلخ)، وسيتم التعامل داخل الدالة كالتالي:

|                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw</pre> <pre>File Edit Format Run Options Window</pre> <pre>def my_function(lang):     for x in lang:         print(x)  pro_lang = ["python", "c++", "c"] my_function(pro_lang)</pre> | <pre>Python 3.8.2 Shell</pre> <pre>File Edit Shell Debug Options Window</pre> <pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32</pre> <pre>Type "help", "copyright", "credits" or "license()" for more information.</pre> <pre>&gt;&gt;&gt;</pre> <pre>===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw =====</pre> <pre>python</pre> <pre>c++</pre> <pre>c</pre> <pre>&gt;&gt;&gt;</pre> <pre>...</pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

الآن استخدمنا القائمة list التي قمنا بإضافتها خارج الدالة وهي اللغات البرمجية ثم قمنا بإدراجها داخل الدالة ثم طباعتها بمعنى ان (pro\_lang= lang) . اي في البيشون يمكن كتابة القائمة قبل الدالة function وفي الأمثلة أدناه تمثيل ذلك حيث قمنا بكتابة القائمة قبل الدالة ثم طباعته ما نفعله في القائمة كما تم شرحه سابقا مثل طول القائمة او كيفية اضافة عنصر بنهاية القائمة وطباعة القائمة الجديدة بعد الاضافة :



|                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw                                                                                                                                                                                                                                                          | Python 3.8.2 Shell                                                                                                                                                                                                                                                                                                                             |
| <pre>File Edit Format Run Options Window Help my_list= [ "python","c++","c","java"] def my_function(list):     print(list)     print(list[0])     print(list[-1])     print(list[3:5])     print(list[:3])     print(list[3:])     print(len(list))     list.append("z")     print(list)  my_function(my_list)</pre> | <pre>File Edit Shell Debug Options Window Help Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb tel) on win32 Type "help", "copyright", "credits" or &gt;&gt;&gt; ===== RESTART: C:\Users\hade ['python', 'c++', 'c', 'java'] python java ['java'] ['python', 'c++', 'c'] ['java'] 4 ['python', 'c++', 'c', 'java', 'z'] &gt;&gt;&gt; &gt;&gt;&gt;</pre> |

|                                                                                                                                                                                                             |                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw                                                                                                                                                 | Python 3.8.2 Shell                                                                                                                                                                                  |
| <pre>File Edit Format Run Options Window Help dic_function = {1:"one", 2:"two",3:"three"} def my_function(collection):     for key in collection:         print(float(key)) my_function(dic_function)</pre> | <pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\ 1.0 2.0 3.0 &gt;&gt;&gt;</pre> |

|                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw (3.8.2)                                                                                                                                                                                                                                            | Python 3.8.2 Shell                                                                                                                                                                                   |
| <pre>File Edit Format Run Options Window Help def my_function(mon, ):     months = {"January":1, "February":2,"March":3,"Apral":4,               "May":5,"June":6,"July":7,"Augest":8, "Septemper":9,               "October":10,"November":11, "Desember":12}     print(months[mon]) my_function("May")</pre> | <pre>File Edit Shell Debug Options Window Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credit &gt;&gt;&gt; ===== RESTART: C:\Users\ 5 &gt;&gt;&gt; &gt;&gt;&gt;</pre> |



|                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw (3.8.2)<br>File Edit Format Run Options Window Help<br><pre>def my_function(mon, ):     months = {"January":1, "February":2,"March":3,"Apral":4,               "May":5,"June":6,"July":7,"Augest":8, "Septemper":9,               "October":10,"November":11, "Desember":12}     print(months[mon]) my_function("May") my_function("July") my_function("October") my_function("Apral")</pre> | Python 3.8.2 Shell<br>File Edit Shell Debug Options<br><pre>Python 3.8.2 (tags/v3.8.1:7b3ab59, tel) on win32 Type "help", "copyright", "credits", "version" &gt;&gt;&gt; ===== RESTART: 5 7 10 4 &gt;&gt;&gt; &gt;&gt;&gt;</pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel\Desktop\tryexception.pyw (3.8.2)<br>File Edit Format Run Options Window Help<br><pre>def my_function(my_collection):     for x in my_collection:         print(x) ddic = {1: 'Admin',2: 'Editor',3: 'Reader'} #dictionary atup = ("a","b","c","d","b") #tuple str1 = "hadeel"      # string nlis = [23,64,12,24] # list  my_function(ddic) my_function(atup) my_function(str1) my_function(nlis)</pre> | Python 3.8.2 Shell<br>File Edit Shell Debug Options Window<br><pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits", "version" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\Desktop\tryexception.pyw 1 2 3 a b c d b h a d e e 1 23 64 12 24 &gt;&gt;&gt;</pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



اما الان فيجب ان نوضح مفهوم مهم في اغلب اللغات البرمجية وهي ان الدالة تستدعي نفسها وهذا ما يعرف بالـ **Recursion** (a function that calls itself) تكون تركيبة عملها كالتالي:

```
def recurse():
    .....
    recurse()
    recurse()
```

التكرار له فائدة حيث أنه يمكننا تكرار البيانات للوصول إلى نتيجة. مع الواجب الحذر من التعامل مع تكرار البيانات. والمثال أدناه وهو عن factorial فمثلا fact(3) سيكون حسابه بالرياضيات  $3*2*1=6$  لنشرح طريقة التكرار باستخدام التكرار بالخطوات التالية للسهولة :

```
x = factorial(3)
def factorial(n):
    if n == 1:
        return 1
    else:
        3
        return n * factorial(n-1)

def factorial(n):
    if n == 1:
        return 1
    else:
        2
        return n * factorial(n-1)

def factorial(n):
    if n == 1:
        return 1
    else:
        1
        return n * factorial(n-1)

3*2=6
is returned
2*1=2
is returned
1
is returned
```



اما برمجيا ستكون كالتالي :

|                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| python.pyw - C:/Users/hadeel/Desktop/python.pyw (3.8.2) <pre> File Edit Format Run Options Window Help  def my_fact(n):     if n == 1:         return 1     else:         return (n * my_fact(n-1))  x = 6 print("The factorial of", "(", x ,")", "is : ", my_fact(x)) </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help  Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:/User The factorial of ( 6 ) is : 720 &gt;&gt;&gt; &gt;&gt;&gt; </pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

مثال اخر عن تكرار الدالة لذاتها recursive Python function في حال ادخال رقم مثلا ٦ يكون جموعه ٦+٥+٤+٣+٢+١ (هذا البرنامج شبيه لـ factorial لكن هنا جمع وليس ضرب)

|                                                                                                                                                                                                                              |                                                                                                                                                                                                                                          |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| python.pyw - C:\Users\hadeel\Desktop\pyt <pre> File Edit Format Run Options Window Help  def sumofn(n):     if n==1:         return 1     return (n+sumofn(n-1))  print(sumofn(2)) print(sumofn(6)) print(sumofn(10)) </pre> | Python 3.8.2 Shell <pre> File Edit Shell Debug Options Window Help  Python 3.8.2 (tags/v3.8.2:7b3ab59, F tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users 3 21 55 &gt;&gt;&gt; &gt;&gt;&gt; </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



**Function** هي عبارة عن: كود بسيط يستخدم لتنفيذ مهمة معينة ويمكن استدعائه باي وقت وبأي مكان داخل البرنامج. ومن خلال الشرح اعلاه يمكننا تقسيم الدالة (function) الى نوعان :

- الاول هو Built-in functions اي الدوال المبنية من قبل البايثون وابسط مثال لهذا النوع هي `print()`

حيث يمكننا كتابة `text` داخل الدالة سيتم طباعته على الشاشة بمجرد التنفيذ. اضافة الى دوال جاهزة اخرى مثل `append` , `abs()`, `min()`, `max()`, `sort()` .... الخ من الدوال.

- النوع الثاني هو User-defined functions هي الدوال التي تكتب من قبل المستخدم اي قوم ببناء الدالة بنفسه وسوف نوضح النوعين بمثال بسيط وهو العدد المطلق ( اي يمكننا عمل مطلق لعدد معين من خلال دالة تم بنائها من قبل البايثون وهي `abs()` ومن جهة الاخر يمكننا بناء دالة تعمل بالطريقة ذاتها تم أنشائها من قبل المستخدم والمثال أدناه سيوضح الفرق :

|                                                                                                |                                                                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel                                                             | Python 3.8.2 Shell                                                                                                                                                                                                                    |
| <pre>File Edit Format Run Options</pre> <pre>print(abs(-1)) print(abs(2)) print(abs(-3))</pre> | <pre>File Edit Shell Debug Options Window</pre> <pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\tryexception.pyw ===== 1 2 3 &gt;&gt;&gt;</pre> |

|                                                                                                                                                                  |                                                                                                                                                                                                                                       |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tryexception.pyw - C:\Users\hadeel                                                                                                                               | Python 3.8.2 Shell                                                                                                                                                                                                                    |
| <pre>File Edit Format Run Options</pre> <pre>def my_abs(x):     if x&gt;=0:         print(x)     else:         print(-(x)) my_abs(-1) my_abs(2) my_abs(-3)</pre> | <pre>File Edit Shell Debug Options Window</pre> <pre>Python 3.8.2 (tags/v3.8.2:7b3ab59, tel) on win32 Type "help", "copyright", "credits" &gt;&gt;&gt; ===== RESTART: C:\Users\hadeel\tryexception.pyw ===== 1 2 3 &gt;&gt;&gt;</pre> |



## ملاحظات مهمة في الـ Function

**ملاحظة الاولى :** هل ما نكتبه داخل اقواس الدالة هو Arguments او Parameters ؟ في الحقيقة ان لها نفس المعنى اي ببساطة انها يستخدمان لتمرير المعلومات (information) داخل function لكن في حال ان اردنا تعریفها من وجهة نظر الدالة فيمكن القول ان :

.function : هو المتغير (variable) المذكور داخل الأقواس في تعريف Parameter -

Argument : هي القيمة التي يتم إرسالها إلى الدالة عندما يتم استدعاؤها.

فلذا ارجوا التساهل معى في حال وجود Argument او العكس في الكتاب لأنه تم اخذ المعنى العام .. والآن من باب الامانة تم ذكر المعنيين .

**ملاحظة الثانية :** هناك بعض function لا يوجد فيها محتوى (فارغة) ولضمان عدم حصول error خطأ يتم كتابة pass داخل الدالة .

```

File Edit Format Run Option File Edit Shell Debug Options Window +
def myfunction():
    pass
Python 3.8.2 (tags/v3.8.2:7b3ab59,
tel) on win32
Type "help", "copyright", "credits"
>>>
===== RESTART: C:\Users\1
>>>

```

**ملاحظة الثالثة :** المسافات و المسافات ثم المسافات في الدوال وجودها مهم جدا في البايثون وهذه الملاحظة مكررة لكن لأهميتها نعيدها ونكررها حيث شاهد في المثالين ادناه أهمية المسافة فالأول عند استدعاء function يتم طباعته function و دالة print() اما في المثال الثاني لم يتم استدعاء function نلاحظ انه تم تنفيذ البرنامج بدون error من خلال دالة الطباعة فقط حيث تم تنفيذها باعتبارها خارج عن الـ function .

لكن في حال تم اضافة مسافة (4 spaces) و لم يتم استدعاء function سوف لن يتم طباعته اي شيء على الشاشة كما في المثال الاخير :



|                                                                                    |                                                                                                                                                             |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| python.pyw - C:\Users\hadeel\Desktop                                               | Python 3.8.2 Shell                                                                                                                                          |
| File Edit Format Run Options                                                       | File Edit Shell Debug Options Window                                                                                                                        |
| def my_print():<br>print("hello python")<br>print("just hello!")<br><br>my_print() | Python 3.8.2 (tags/v3.8.2:7b3ab<br>tel) on win32<br>Type "help", "copyright", "cre<br>>>><br>===== RESTART: C:<br>just hello!<br>hello python<br>>>><br>... |

|                                                                  |                                                                                                                                          |
|------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| python.pyw - C:\Users\hadeel\Desktop                             | Python 3.8.2 Shell                                                                                                                       |
| File Edit Format Run Options                                     | File Edit Shell Debug Options Window                                                                                                     |
| def my_print():<br>print("hello python")<br>print("just hello!") | Python 3.8.2 (tags/v3.8.2:7b3ab<br>tel) on win32<br>Type "help", "copyright", "cred<br>>>><br>===== RESTART: C:\\\<br>just hello!<br>>>> |

|                                                                  |                                                                                                                                                  |
|------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| python.pyw - C:\Users\hadeel\Desktop                             | Python 3.8.2 Shell                                                                                                                               |
| File Edit Format Run Options View                                | File Edit Shell Debug Options Window Help                                                                                                        |
| def my_print():<br>print("hello python")<br>print("just hello!") | Python 3.8.2 (tags/v3.8.2:7b3ab59, F<br>tel) on win32<br>Type "help", "copyright", "credits" ,<br>>>><br>===== RESTART: C:\Users'\<br>>>><br>>>> |



## Lambda in Python

### لبايثن في Lambda

يمكن تعريف Lambda بانها تعريف لدالة ليس لها اسم . غالبا ما تكون هذه الدالة من سطر واحد وتعمل على ارجاع قيمة معينة عند استدعائها من خلال تعريف الـ function باستخدام الكلمة function ولا نستخدم الكلمة def التي تعودنا استخدامها في تعريف الدوال..

من مميزات lambda انها تحتوي على اي عدد من arguments لكن تحتوي على تعبير واحد(expression) فقط ويمكن تمثيل تركيبتها كالتالي :

```
lambda [arg1 [,arg2,....arg n]]:expression
```

|                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  python.pyw - C:\Users\hadeel\Desktop<br>File Edit Format Run Options<br><pre>b = lambda a : a + 10 print(b(6))</pre> |  Python 3.8.2 Shell<br>File Edit Shell Debug Options Window Help<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32<br>Type "help", "copyright", "credits" or "license" for more information >>> ===== RESTART: C:\Users\hadeel\Desktop\python.pyw ===== 16 >>> >>> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  python.pyw - C:\Users\hadeel\Desktop<br>File Edit Format Run Options<br><pre>x = lambda a , b : a + b print(x(5,5))</pre> |  Python 3.8.2 Shell<br>File Edit Shell Debug Options Window Help<br>Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32<br>Type "help", "copyright", "credits" or "license" for more information >>> ===== RESTART: C:\Users\hadeel\Desktop\python.pyw ===== 10 >>> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



يمكن لهذه الدالة استخدامها بداخل دوال اخرى وهذه هي حدى ميزاتها ايضا :

The screenshot shows two windows side-by-side. On the left is a code editor window titled "python.pyw - C:\Users\hadeel\Desktop\p". It contains the following Python code:

```
def my_function(b):
    return lambda a : a * b
x= my_function(10)
print(x(20))
```

On the right is the "Python 3.8.2 Shell" window. It shows the Python interpreter's prompt and the output of the code execution:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:46:08) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
=====
RESTART: C:\Users\hadeel\Desktop\python.pyw
200
>>>
>>>
```

. my\_function تم دمجها او استخدامها بداخل دالة lambda



هناك العديد من المواضيع المتنوعة في البايثون منها Python Class و Python Module و Python – File و Python Inheritance .... الخ. لكن قد يكون الأفضل شرحها في المرحلة القادمة بعد يتم تعلم ما تم شرحه سابقا.

وفي النهاية أتمنى أن تكون هذه الصفحات القليلة

مفيدة لكم

والحمد لله رب العالمين والصلوة والسلام على سيد المرسلين محمد (صلى الله عليه وسلم )

*Hadeel*



Python 3.8.2 Shell

```
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb
tel) on win32
Type "help", "copyright", "credits" or
>>> print("شكرا لكم ..... تحياتي")
شكرا لكم ..... تحياتي
>>>
>>>
```

