

SEPTEMBER 1 – 7, 2019



المبادرة السعودية للمطورين

تعلم .. فكر .. حاول .. أبداع

المبادرة السعودية للمطورين

مسار Python

مشرفي المسار:

عبدالله عوده – انتصار النصار – رؤى كردي – ليلى المصعبي



ملاحظات قبل بدء الدروس:

- على المتدربين نشر كل يوم الجزئية التي تم كتابتها من النص البرمجي في الـ **Github** تحت **Topic** بعنوان **saudidevorg** كما تم توضيحه في دروس الـ **Github** سابقاً

- على المتدربين نشر كل يوم مقدار التقدم وصورة لما تم تعلمه وتطبيقه على **Twitter** تحت الهاشتاقات:
#المبادرة_السعودية_للمطورين
_100#يوم_برمجة
#100DaysOfCode

تمنياتنا لك بالتوفيق
المبادرة السعودية للمطورين

اليوم الثالث عشر

القوائم في لغة البايثون

Python Lists

➤ List

القائمة

القائمة تعتبر أحد هياكل البيانات في لغة بايثون وهي عبارة عن مصفوفة ليس لها حجم ثابت، يمكن تخزين قيم مختلفة الأنواع بداخلها، يمكن أن تتكون من سلاسل نصية، أرقام، ومن قوائم أخرى أيضا. يمكنك تبديل القيم متى ما شئت.

A **list** is a collection which is ordered and changeable. Allows duplicate members.

In **Python** lists are written with square brackets `[]`.

لتعريف القائمة نستخدم الرمز `[]` فالعناصر تكون بين القوسين المربعين، حتى يفهم البايثون أنك تريد تحديد قائمة. يمكنك تمرير القيم بداخله، ويجب عليك وضع فاصلة بين كل عنصر.

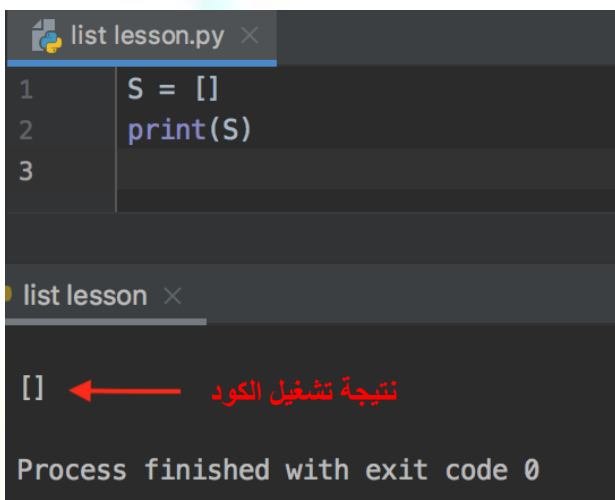
Example

أفضل طريقة لفهم القوائم (القائمة) هي العمل عليها، انظر للأمثلة

Create a **List**

```
S=[]  
print (S)
```

في المثال التالي قمنا بتعريف قائمة فارغة اسمها S ، لا تحتوي على أي عنصر.

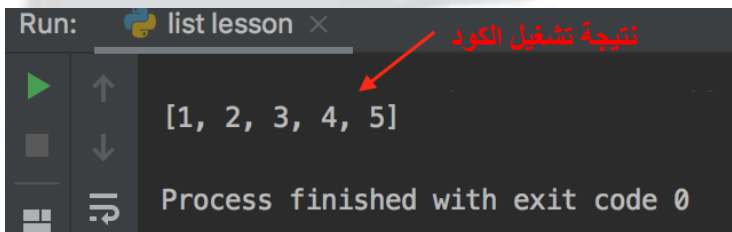


```
list lesson.py x  
1 S = []  
2 print(S)  
3  
  
list lesson x  
[ ] ← نتيجة تشغيل الكود  
Process finished with exit code 0
```

في هذا المثال قمنا بتعريف قائمة اسمها **numbers** تحتوي على أعداد صحيحة فقط.

Example

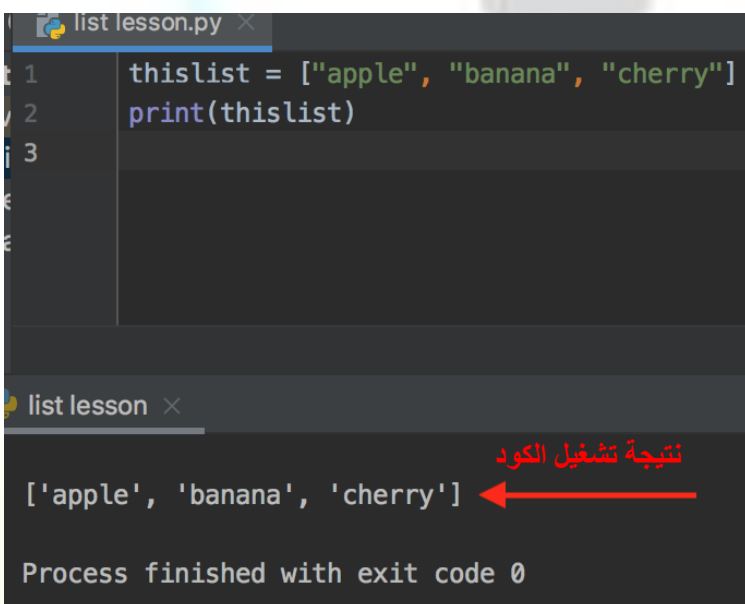
```
numbers = [1, 2, 3, 4, 5]
print(numbers)
```



Example

هنا قمنا بتعريف قائمة اسمها **thislist** تحتوي على نصوص.

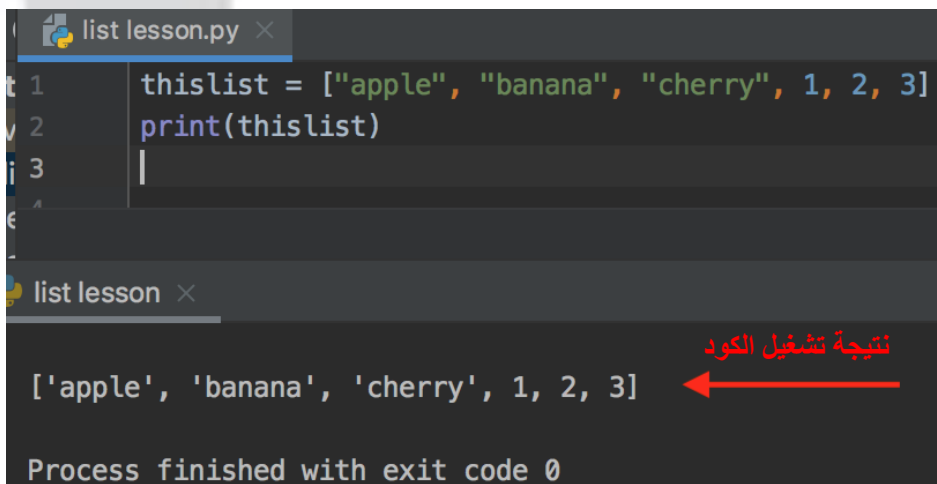
```
thislist = ["apple", "banana", "cherry"]
print(thislist)
```



Example

هنا قمنا بتعريف قائمة اسمها `thislist` تحتوي على نصوص وأعداد صحيحة.

```
thislist = ["apple", "banana", "cherry", 1, 2, 3]
print(thislist)
```



The screenshot shows a code editor with a file named `list lesson.py`. The code contains two lines: `thislist = ["apple", "banana", "cherry", 1, 2, 3]` and `print(thislist)`. Below the code, the output is displayed: `['apple', 'banana', 'cherry', 1, 2, 3]`. A red arrow points from the text "نتيجة تشغيل الكود" (Code execution result) to the output. At the bottom, it says "Process finished with exit code 0".

مُبرمجنا، قم بإنشاء قائمة تحتوي على الأعداد العشرية فقط

➤ Access Items

الوصول لعناصر القائمة

للوصول لأي عنصر في القائمة عليك استخدام رقم الـ **index** الخاص بالعنصر الذي تريد.

You access the **list** items by referring to the **index** number.

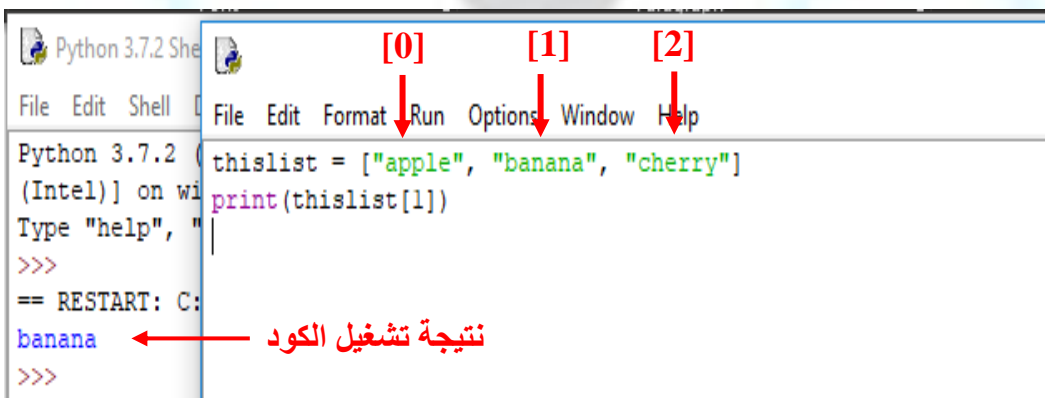
- إذا أردت الوصول لعناصر القائمة ابتداءً من أول عنصر تم إدخاله (من اليسار إلى اليمين) فإن الـ **index** لأول عنصر هو صفر **0**
- للوصول لعناصر القائمة من آخر عنصر تم إدخاله (من اليمين لليسار) فإن الـ **index** لآخر العنصر سيكون **-1**

Example

Print the second item of the **list**

هنا قمنا بطباعة قيمة **ثاني** عنصر في القائمة

```
thislist = ["apple", "banana", "cherry"]  
print(thislist[1])
```



```
Python 3.7.2 Shell  
File Edit Shell  
Python 3.7.2 (Intel) on win32  
Type "help", "copyright()", or "credits()" for more  
>>>  
== RESTART: C:\Python37\Shell ==  
banana  
>>>
```


عرض جميع القيم في القائمة

➤ Loop Through a List

You can loop through the **list** items by using a **for** loop.

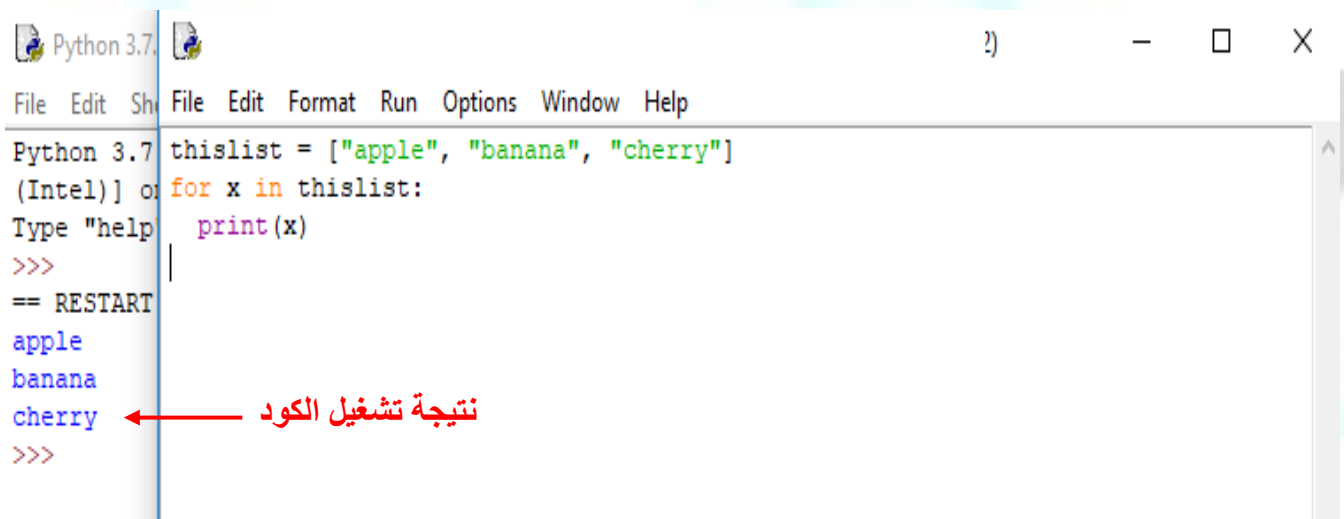
Example

Print all items in the **list**, one by one.

```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
    print(x)
```

هنا قمنا بتعريف قائمة اسمها **thislist** تحتوي على نصوص ثم قمنا بطباعة جميع القيم الموجودة في القائمة باستخدام الحلقة **for**

في كل مرة سيتم وضع قيمة عنصر من عناصر **thislist** في المتغير **x** ثم سيتم طباعتها سيتم شرح الحلقة **for** في دروس قادمة بإذن الله.



```
Python 3.7.0
File Edit Shell View Help
Python 3.7.0 (tags/v3.7.0:1bf3651, Oct 4 2018) [AMD64] on win32
Type "help()" for more.
>>> thislist = ["apple", "banana", "cherry"]
>>> for x in thislist:
>>>     print(x)
>>>
== RESTART ==
apple
banana
cherry
>>>
```

قم بإنشاء قائمة تحتوي على أعداد فقط، عشرية أم صحيحة كانت ثم قم بطباعة جميع العناصر كما في المثال السابق.

➤ Change Item Value

تغيير قيمة العنصر

To change the value of a specific item, refer to the **index** number.

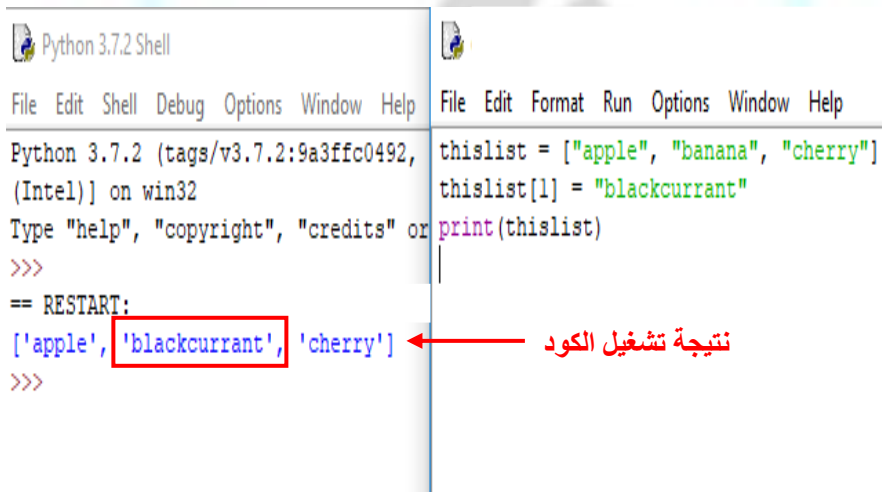
لتغيير قيمة عنصر ما، يتم إسناد القيمة الجديدة لرقم الـ **index** للعنصر المراد تغييره.

Example

Change the second item.

في المثال هنا قمنا بتغيير قيمة العنصر **الثاني**

```
thislist = ["apple", "banana", "cherry"]  
thislist[1] = "blackcurrant"  
print(thislist)
```



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, (Intel)) on win32  
Type "help", "copyright", "credits" or >>>  
== RESTART:  
['apple', 'blackcurrant', 'cherry']  
>>>
```

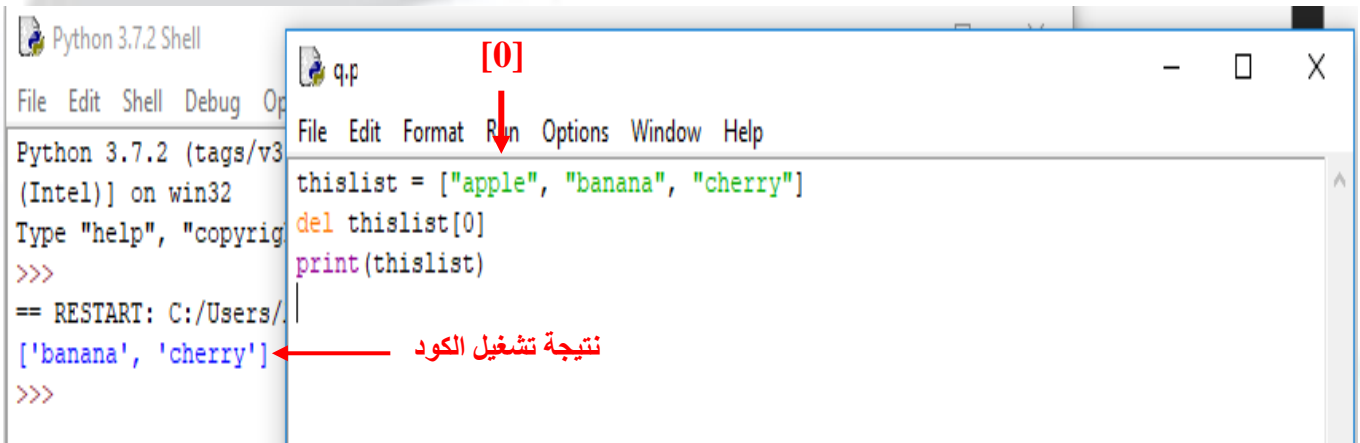
Example

تُستخدم **del** لحذف القائمة كاملة من الذاكرة أو لحذف عنصر محدد فيها.

The **del** keyword removes the specified index.

```
thislist = ["apple", "banana", "cherry"]  
del thislist[0]  
print(thislist)
```

هنا قمنا بحذف عنصر محدد وهو العنصر الأول



The screenshot shows two windows. The left window is a 'Python 3.7.2 Shell' with the following output:
Python 3.7.2 (tags/v3.7.2 [0] on win32
Type "help", "copyright", "credits()", and "quit()" for more
>>>
== RESTART: C:/Users/...
['banana', 'cherry']
>>>
The right window is a code editor with the following code:
thislist = ["apple", "banana", "cherry"]
del thislist[0]
print(thislist)
A red arrow points from the text '[0]' to the index '0' in the code. Another red arrow points from the text 'نتيجة تشغيل الكود' to the output '['banana', 'cherry']' in the shell.

Example

هنا في هذا المثال قمنا بحذف القائمة/المصفوفة كاملة

The **del** keyword can also delete the **list** completely.

```
thislist = ["apple", "banana", "cherry"]  
del thislist  
#this will cause an error because you have successfully deleted "thislist".
```

The list will be deleted.

```
NameError: name 'thislist' is not defined
```

نتيجة تشغيل الكود

ملاحظة:

عند حذفك لعنصر ما في القائمة بواسطة **del** فإن العناصر ستترتب من جديد في بايثون، وتتحدث أرقام الـ **index** تلقائياً لكل عنصر.

```

[0] [1] [2] [3] [4]
1 thislist = ["A", "B", "C", "D", "E"]
2 print(thislist)
3

```

نتيجة تشغيل الكود

```

['A', 'B', 'C', 'D', 'E']

```

هنا لدينا خمسة عناصر

```

[0]
1 thislist = ["A", "B", "C", "D", "E"]
2 del thislist[0]
3 print(thislist)

```

نتيجة تشغيل الكود

```

['B', 'C', 'D', 'E']

```

عملية حذف العنصر الأول

```

[0] [1] [2] [3]
[محذوف]
1 thislist = ["A", "B", "C", "D", "E"]
2 del thislist[0]
3 del thislist[1]
4 print(thislist)
5

```

سيتم حذف العنصر C بعد تشغيل الكود

نتيجة تشغيل الكود

```

['B', 'D', 'E']

```

بعد ما تم حذف العنصر الأول A أصبح عدد العناصر أربعة
وهنا أصبح العنصر B يحمل الـ **index** رقم 0

رائع!
أتممت درسك

روابط قد تساعدك

Check the links below

- [Python lists](#)
- [Python Programming #5 - Introduction to Lists](#)

طبق ما تعلمته في هذا الدرس
ولا تنسى مشاركتنا أكوادك

اليوم الرابع عشر

القوائم في لغة البايثون **Python** Lists 2

تقسيم القوائم إلى أجزاء / تجزئة القائمة

في هذا المثال قمنا بتعريف قائمة وهي `thislist` ولكي نقوم بطباعة جزء من عناصر هذه القائمة نضع رقم الـ **index** للعناصر

هنا قمنا بطباعة قيم العناصر `B` , `C` وذلك ابتداءً من العنصر `B` الذي يحمل الـ **index 1** إلى ما قبل العنصر `D` الذي يحمل الـ **index 3**

العنصر الذي يحمل الـ **index 3** لن تتم طباعته

[1] [2] [3]



```
1 thislist = ["A", "B", "C", "D", "E"]
2 print(thislist[1:3])
```

list lesson ×

نتيجة تشغيل الكود ← ['B', 'C']

Process finished with exit code 0

قم بتطبيق مثال آخر على التقطيع/التجزئة للقوائم

أفضل طريقة للفهم هي التطبيق

➤ Check if Item Exists

التحقق من وجود عنصر في القائمة

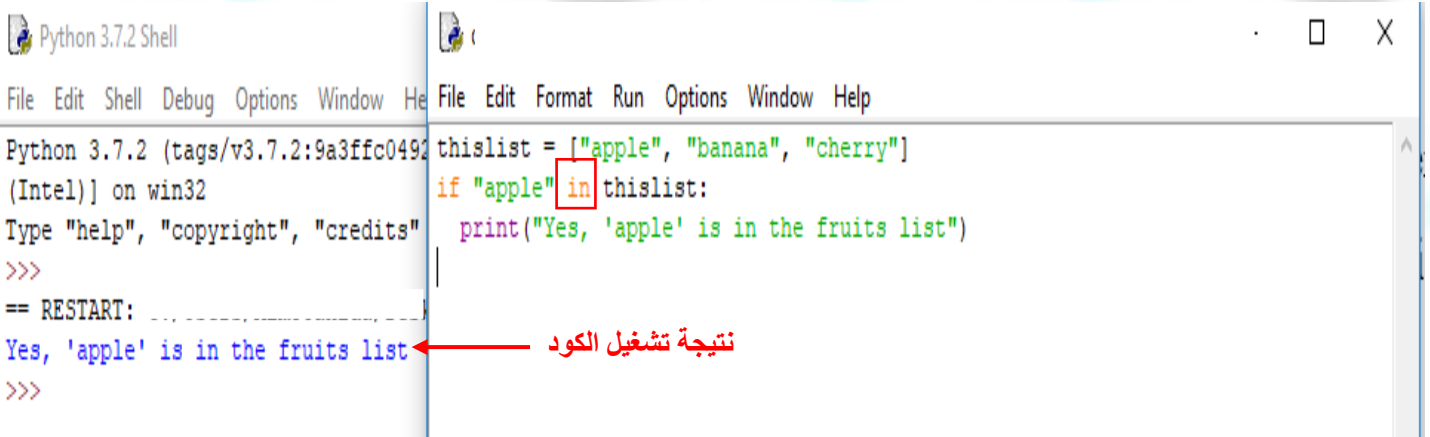
للبحث عن قيمة معينة داخل القائمة نستخدم العامل **in**

To determine if a specified item is present in a **list** use the **in** keyword.

Example

Check if "apple" is present in the **list**.

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
    print("Yes, 'apple' is in the fruits list")
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492 thislist = ["apple", "banana", "cherry"]
(Intel)] on win32 if "apple" in thislist:
Type "help", "copyright", "credits" print("Yes, 'apple' is in the fruits list")
>>>
== RESTART:
Yes, 'apple' is in the fruits list
>>>
```

نتيجة تشغيل الكود

➤ Repeat Item in List

To repeat an item in a **list**, use the ***** operator.

لتكرار قيمة معينة نستخدم العامل *

Example

```
1 thislist = ["بايثون"] * 4
2 print(thislist)
```

list lesson ×

```
['بايثون', 'بايثون', 'بايثون', 'بايثون']
```

في هذا المثال قمنا بإنشاء قائمة **thislist**

ووضعنا فيها أربع قيم نصية "بايثون" باستخدام العامل *

ثم كتبنا أمر الطباعة للقائمة **thislist**

ليطبع لنا كلمة "بايثون" أربع مرات

نتيجة تشغيل الكود

➤ + Operator in List

To add 2 **lists** or more into one **list**.

يستخدم هذا العامل + لدمج أكثر من قائمة في قائمة واحدة.

Example

```
1 thislist1 = ["Ahmad", "Khalid", "Omar"]
2 thislist2 = ["Sara", "Hind", "Batool"]
3
4 thislist3 = thislist1 + thislist2
5
6 print(thislist3)
7
```

list lesson ×

```
['Ahmad', 'Khalid', 'Omar', 'Sara', 'Hind', 'Batool']
```

ألق نظرة على هذا المثال

نتيجة تشغيل الكود

قم بتطبيق مثال لدمج قائمتين إحداهما تحتوي على أعداد عشرية والأخرى أعداد صحيحة.

أتممت درسك بنجاح!
تابع التقدّم

طبّق ما تعلمته في هذا الدرس
ولا تنسى مشاركتنا أكوادك

اليوم الخامس عشر

دوال القوائم في لغة البايثون

Python Lists 3

List Methods

➤ List Length

عدد عناصر القائمة

استخدم الدالة **len()** لمعرفة عدد عناصر القائمة، فهي تقوم بترجيع عدد صحيح وهو عدد العناصر.

To determine how many items a **list** has, use the **len()** method.

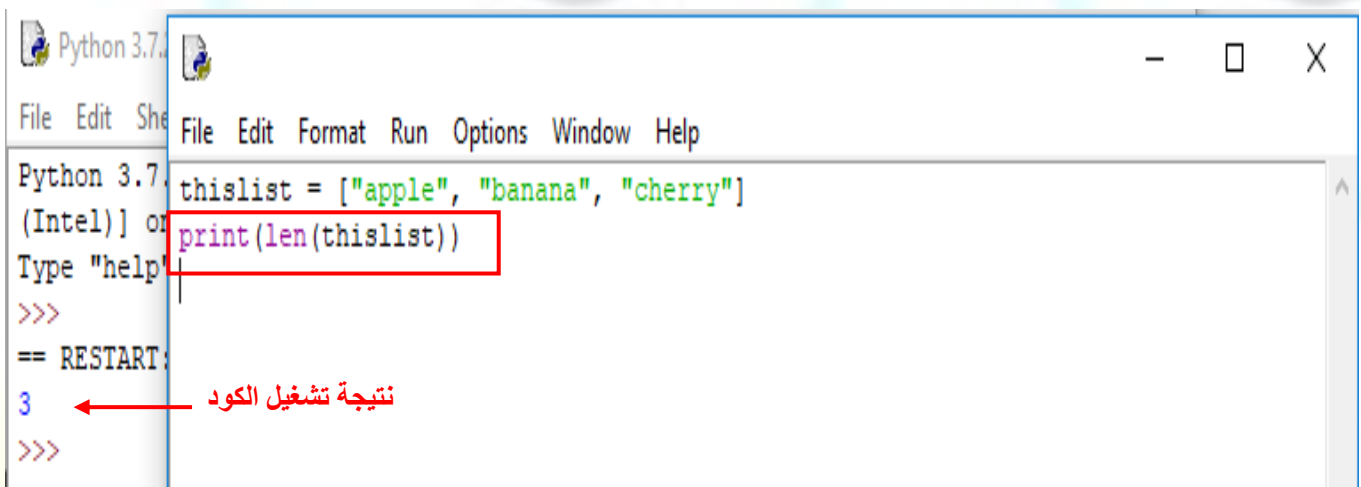
Example

Print the number of items in the **list**.

```
thislist = ["apple", "banana", "cherry"]  
print(len(thislist))
```

في هذا المثال القائمة تحتوي على ٣ عناصر (قيم نصية).

نستدعي الدالة **len()**، ثم نكتب بداخل الأقواس () الذي هو مكان المُعاملات (الباراميتير)، نمرر أو نكتب فيه اسم القائمة التي نريد معرفة عدد عناصرها.



```
Python 3.7.4 Shell  
File Edit Shell  
Python 3.7.4 Shell  
(Intel) on ...  
Type "help()" for more  
>>>  
== RESTART: ==  
3  
>>>
```

نتيجة تشغيل الكود

➤ Add Items

إضافة عنصر جديد في القائمة

استخدم الدالة **append()** إذا أردت إضافة عنصر جديد في آخر القائمة.

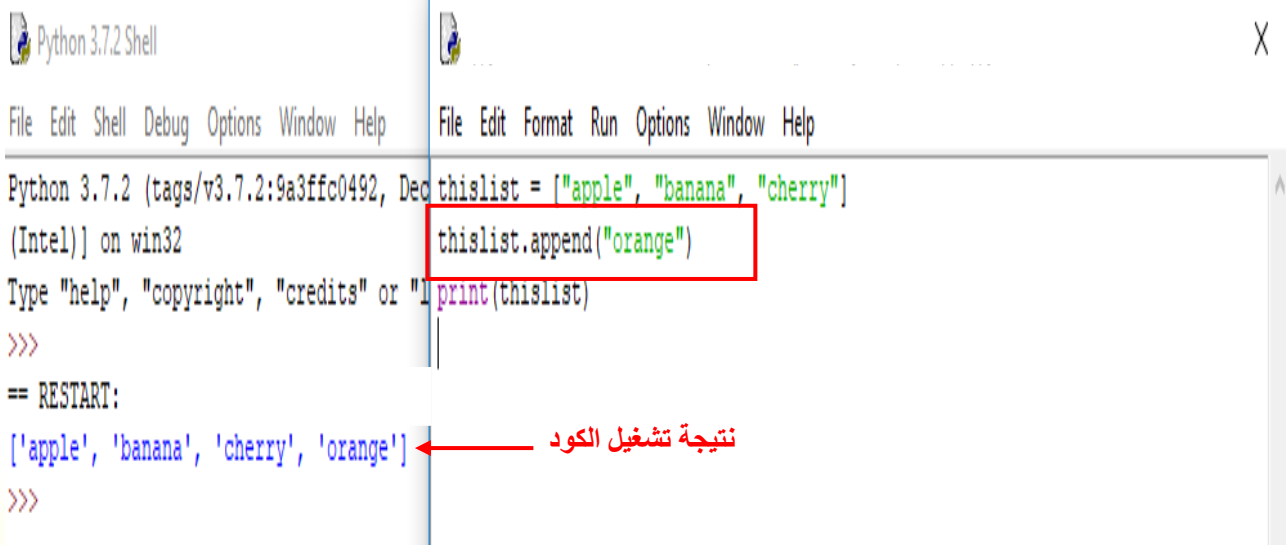
To add an item to the end of the **list**, use the **append()** method.

اكتب اسم القائمة التي تريد إضافة عنصر في آخرها ثم نقطة . ثم نستخدم الدالة **append()** ونكتب العنصر الذي نريد إضافته داخل الأقواس () مكان المُعاملات (الباراميتري)، كما هو موضح بالمثال التالي:

Example

Using the **append()** method to append an item.

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 11 2019) on win32
Type "help", "copyright", "credits" or "license()"
>>> thislist = ["apple", "banana", "cherry"]
>>> thislist.append("orange")
>>> print(thislist)
== RESTART: ==
['apple', 'banana', 'cherry', 'orange']
>>>
```

نتيجة تشغيل الكود

- To add an item at the specified index, use the **insert()** method. إضافة عنصر جديد في مكان محدد في القائمة

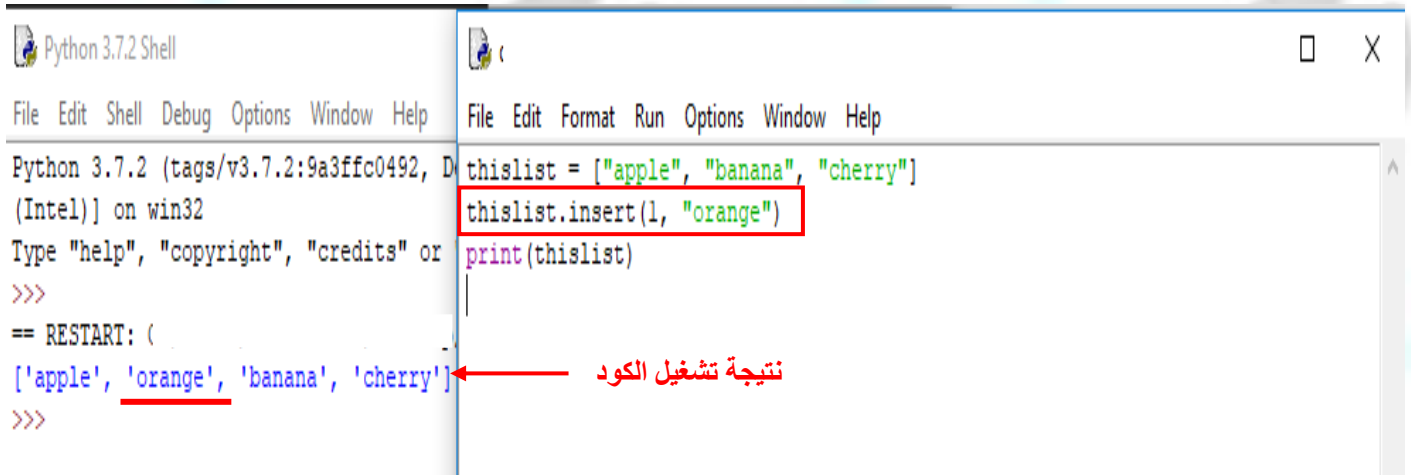
Example

هنا في هذا المثال قمنا بإضافة عنصر جديد في مكان معين ومحدد باستخدام دالة **insert()**

Insert an item as the second position.

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(1, "orange")
print(thislist)
```

نستدعي الدالة ثم بـمكان المُعاملات (الباراميتر) هنا داخل الأقواس () نمرّر/نكتب رقم الـ **index** للمكان الذي نريد أن نضيف فيه ثم فاصلة , ثم نكتب/نمرّر العنصر المراد إضافته.



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, D
(Intel)) on win32
Type "help", "copyright", "credits" or
>>>
== RESTART: (
['apple', 'orange', 'banana', 'cherry']
>>>
```

نتيجة تشغيل الكود

➤ Remove Item

حذف عنصر من القائمة

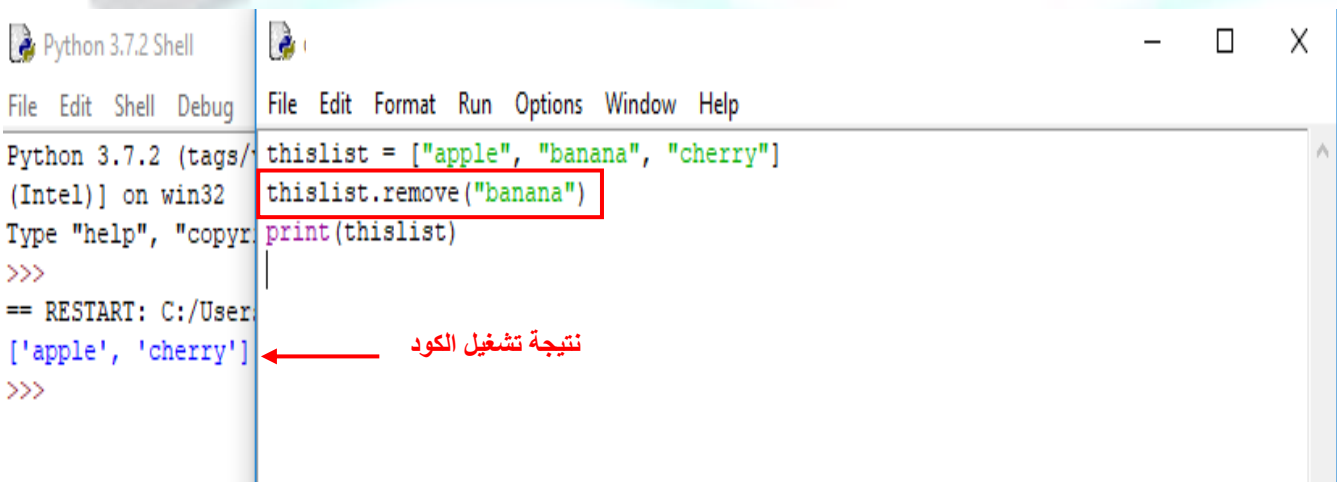
There are several methods to remove items from a **list**.

هناك دالتين لحذف العنصر من القائمة

Example

The **remove()** method removes the specified item. الدالة الأولى هي **remove()** تقوم بحذف عنصر محدد في القائمة

```
thislist = ["apple", "banana", "cherry"]  
thislist.remove("banana")  
print(thislist)
```



```
Python 3.7.2 Shell  
File Edit Shell Debug  
Python 3.7.2 (tags/thislist = ["apple", "banana", "cherry"]  
(Intel)) on win32 thislist.remove("banana")  
Type "help", "copyr: print(thislist)  
>>>  
== RESTART: C:/User:  
['apple', 'cherry']  
>>>
```

نتيجة تشغيل الكود

في حال كان هناك أكثر من عنصر لهما نفس القيمة، فإن الدالة تقوم بحذف أول عنصر يتم إيجاده في القائمة.

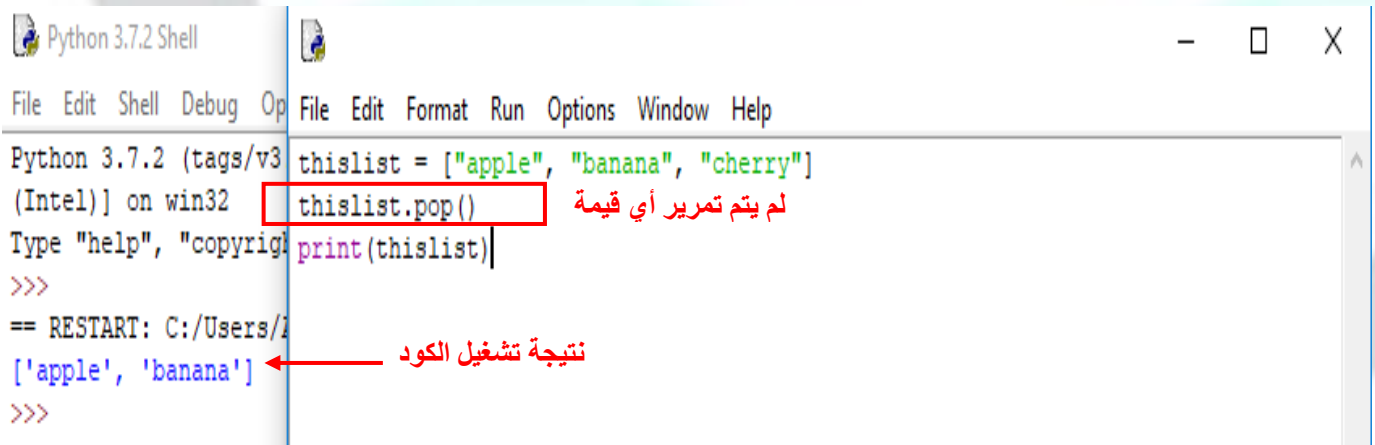
جرب أيها المبرمج!

Example

The **pop()** method removes the specified index. **pop()** دالة الحذف الثانية هي تقوم بحذف عنصر محدد في القائمة (or the last item if index is not specified). تقوم بحذف العنصر لكن عن طريق استدعاء رقم الـ **index** لهذا العنصر.

وفي حال قمت باستدعاء هذه الدالة بدون أن تمرر لها أي رقم في (الباراميتر)، فإنها تحذف آخر عنصر موجود في القائمة، انظر المثال التالي:

```
thislist = ["apple", "banana", "cherry"]
thislist.pop()
print(thislist)
```



```
Python 3.7.2 Shell
File Edit Shell Debug Op File Edit Format Run Options Window Help
Python 3.7.2 (tags/v3.7.2:1) on win32
thislist = ["apple", "banana", "cherry"]
thislist.pop()
print(thislist)
>>>
== RESTART: C:/Users/...
['apple', 'banana']
>>>
```

لم يتم تمرير أي قيمة

نتيجة تشغيل الكود

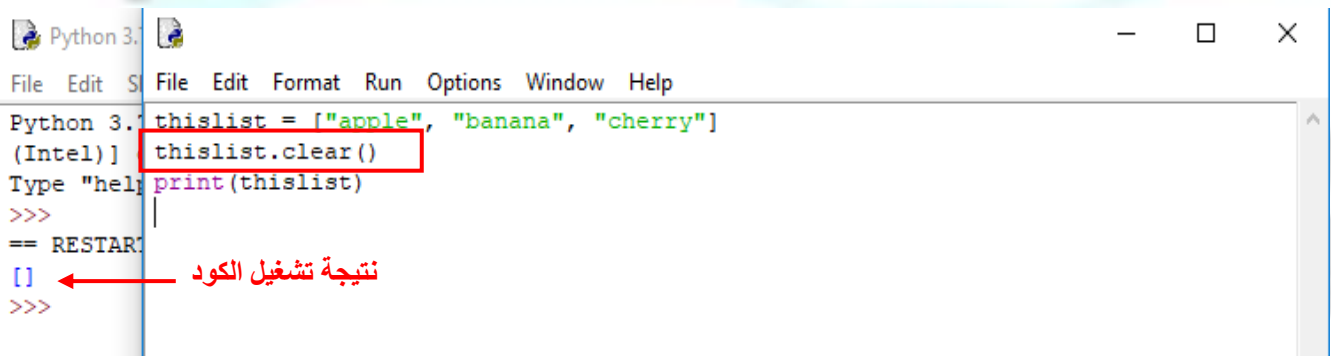
قم بتطبيق كلا الحالتين للدالة **pop()**

Example

The **clear()** method empties the **list**.

تُستخدم دالة **clear()** لحذف جميع العناصر في القائمة.

```
thislist = ["apple", "banana", "cherry"]  
thislist.clear()  
print(thislist)
```



```
Python 3.7  
File Edit Shell File Edit Format Run Options Window Help  
Python 3.7 thislist = ["apple", "banana", "cherry"]  
(Intel) ] thislist.clear()  
Type "help" for help print(thislist)  
>>>  
== RESTART ==>  
[] ← نتيجة تشغيل الكود  
>>>
```

نسخ القائمة

➤ Copy a List

You cannot copy a **list** simply by typing **list2 = list1**, because: **list2** will only be a reference to **list1**, and changes made in **list1** will automatically also be made in **list2**.

There are ways to make a copy, one way is to use the built-in List method **copy()**.

لا يمكنك نسخ قائمة بهذه الطريقة **list2 = list1**

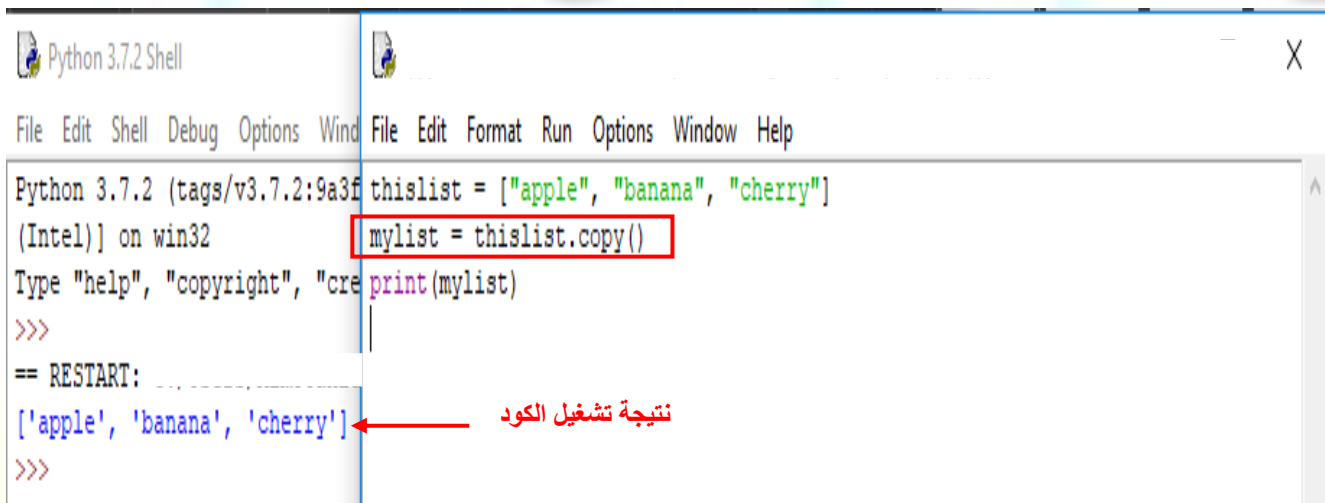
إذا أردت أن تقوم بعمل نسخ للقائمة استخدم الدالة **copy()** فهي تقوم بترجييع نسخة من القائمة التي استدعتها

Example

Make a copy of a **list** with the **copy()** method.

```
thislist = ["apple", "banana", "cherry"]  
mylist = thislist.copy()  
print(mylist)
```

في هذا المثال قمنا بنسخ القائمة **thislist** ونسخنا قيمها إلى القائمة **mylist**



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window File Edit Format Run Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3f503, Dec 12 2019, [AMD64]) on win32  
Type "help", "copyright", "credits()" or "quit()" for more  
>>> thislist = ["apple", "banana", "cherry"]  
>>> mylist = thislist.copy()  
>>> print(mylist)  
>>>  
== RESTART: ==  
['apple', 'banana', 'cherry']  
>>>
```

نتيجة تشغيل الكود

Example

لإيضاح فكرة النسخ أكثر، انظر الأمثلة التالية:

```
list lesson.py x
1 thislist = ["Ahmad", "Omar", "Khalid"]
2 mylist = thislist.copy()
3 thislist.pop(0)
4 print(mylist)
5 print(thislist)
```

نتيجة تشغيل الكود

```
list lesson x
['Ahmad', 'Omar', 'Khalid']
['Omar', 'Khalid']
```

لاحظ :

هذا يعني أنه من بعد عملية النسخ
أي تغيير سيطراً على القائمة `thislist`
لن يؤثر على قيم القائمة `mylist`

Example

ماذا لو قمنا بالنسخ بطريقة الإسناد `list2 = list1`، لنرى ماذا سيحدث:

```
list lesson.py x
1 thislist = ["Ahmad", "Omar", "Khalid"]
2 mylist = thislist
3 thislist.pop(0)
4 print(mylist)
5 print(thislist)
```

نتيجة تشغيل الكود

```
list lesson x
['Omar', 'Khalid']
['Omar', 'Khalid']
```

لاحظ :

أي تغيير سيطراً على القائمة `thislist`
سيؤثر على قيم القائمة `mylist`

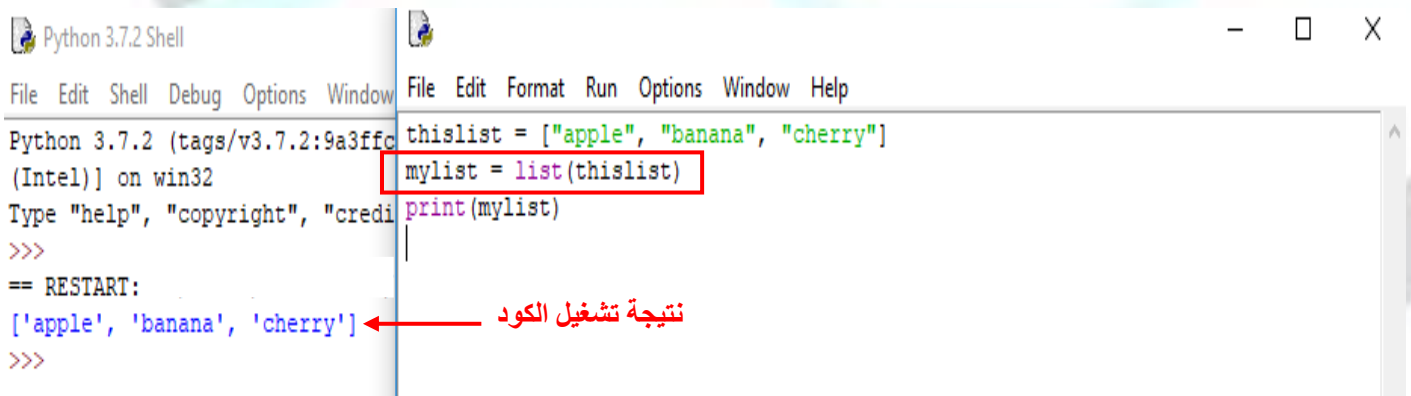
- Another way to make a copy is to use the built-in method **list()**.

طريقة أخرى لعمل نسخ للقائمة، وذلك باستخدام الدالة **list()**

Example

Make a copy of a **list** with the **list()** method.

```
thislist = ["apple", "banana", "cherry"]  
mylist = list(thislist)  
print(mylist)
```



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window  
Python 3.7.2 (tags/v3.7.2:9a3ffc0, Dec 11 2019) on win32  
Type "help", "copyright", "credits()" or "quit()" for more  
>>>  
== RESTART: ==  
['apple', 'banana', 'cherry']  
>>>
```

نتيجة تشغيل الكود

تابع التقدّم ..

بناء/تكوين قائمة

The **list()** Constructor

It is also possible to use the **list()** constructor to make a new **list**.

يمكنك إنشاء قائمة جديدة باستخدام الدالة **list()**

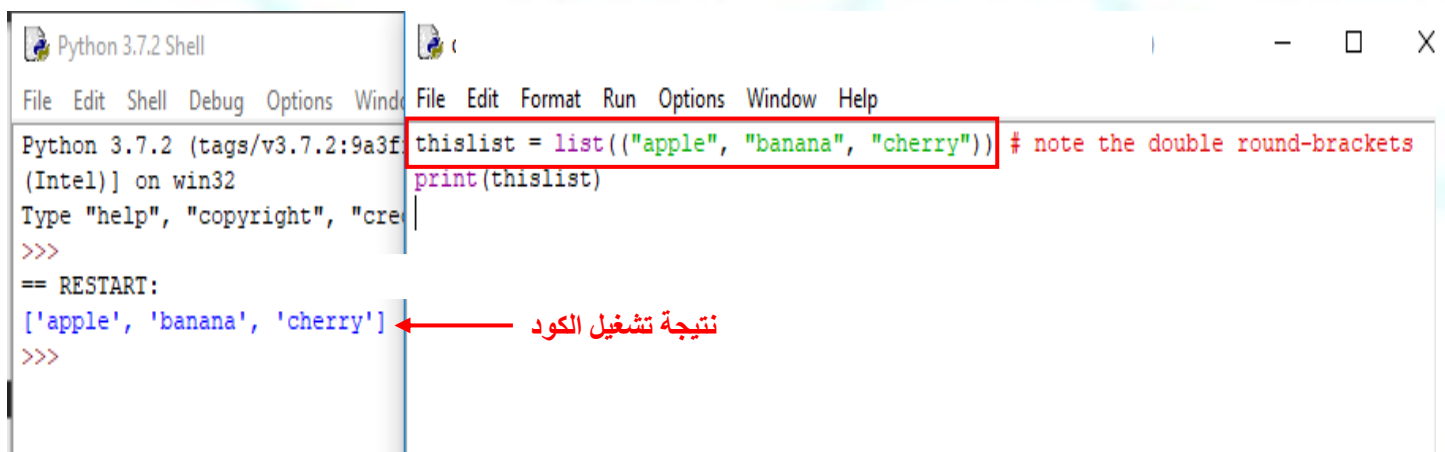
Example

Using the **list()** constructor to make a **list**.

```
thislist = list(("apple", "banana", "cherry")) # note the  
double round-brackets  
print(thislist)
```

لاحظ هنا طريقة كتابة القائمة ستختلف باستخدام الدالة **list()**

نستدعي الدالة **list()** ثم نضع بداخلها العناصر داخل أقواس مستديرة () بدلا من الأقواس المربعة الخاصة بالقوائم []



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3f503, Dec 12 2019, [AMD64]) on win32  
Type "help", "copyright", "credits()" or "quit()" for more  
>>>  
== RESTART:  
['apple', 'banana', 'cherry']  
>>>
```

نتيجة تشغيل الكود

➤ List Methods

دوال القائمة في لغة بايثون

بايثون تحتوي على دوال جاهزة يمكنك استخدامها للتعامل مع القوائم وهي ما ناقشنا أغلبها في هذا الدرس.

Python has a set of built-in methods that you can use on **lists**.

Method	Description
<u>append()</u>	Adds an element at the end of the list
<u>clear()</u>	Removes all the elements from the list
<u>copy()</u>	Returns a copy of the list
<u>count()</u>	Returns the number of elements with the specified value
<u>extend()</u>	Add the elements of a list (or any iterable), to the end of the current list
<u>index()</u>	Returns the index of the first element with the specified value
<u>insert()</u>	Adds an element at the specified position
<u>pop()</u>	Removes the element at the specified position
<u>remove()</u>	Removes the item with the specified value
<u>reverse()</u>	Reverses the order of the list
<u>sort()</u>	Sorts the list

طبّقها لتتأكد من فهمك.

أتممت درسك يا مبرمج الغد!

روابط قد تساعدك

Check the links below

- [20 Python lists](#)
- [في بايثون list القوائم](#)
- [Python Programming #5 - Introduction to Lists](#)
- [07 - Python - Beginners Tutorial - Lists](#)
- [#5 Python Tutorial for Beginners | List in Python](#)
- [Learn Python in Arabic #18 - 18 Python list متغير قائمة Python شرح بايثون بالعربي](#)
- [Learn Python in Arabic #19 Python list append extend add del متغير قائمة الاضافة و الحذف](#)
- [Learn Python Programming - 12 - Lists of Things](#) start at time 3:36
- [Lists | الدرس 5: تعلم بايثون 3 | القوائم](#)
- [القوائم - Lists 7 - تعلم البرمجة بلغة بايثون](#)
- [Python tutorial List with For القوائم](#)
- [Python tutorial List row and cloumn اكثر من بعد](#)
- [التعامل مع القوائم والسلاسل النصية في لغة بايثون](#)

طبق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم السادس عشر

الصفوف (الأزواج المرتبة) في لغة البايثون

Python Tuples

➤ Tuple

الصف

الصفوف تعتبر أيضا أحد هياكل البيانات في لغة بايثون وهي تشبه القوائم إلى حد ما لكنها تستخدم الأقواس () بدلا من []. تختلف أيضا بأنها غير قابلة للتغيير فلا يمكن تغيير قيمها فهي عبارة عن مصفوفة لها حجم ثابت بعكس القوائم. يمكن تخزين قيم مختلفة الأنواع بداخلها، لكن لا يمكن تبديل القيم فيها.

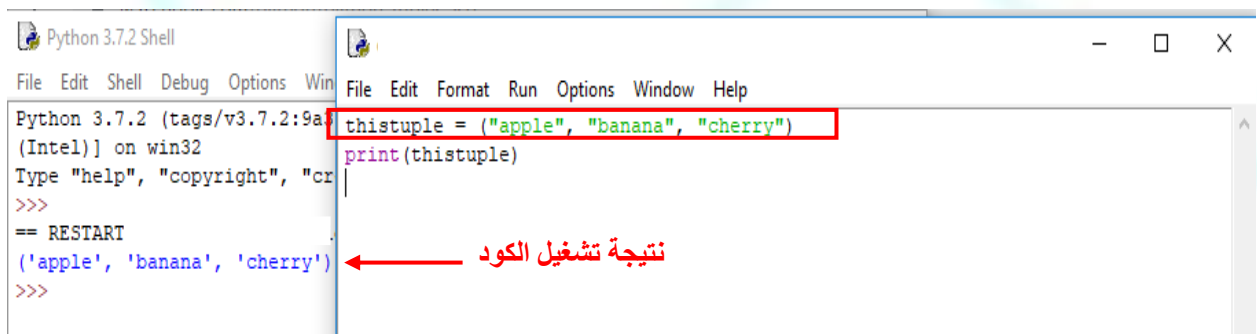
A **tuple** is a collection which is ordered and unchangeable.

In **Python** tuples are written with round brackets () .

Example

لتعريف/إنشاء صف تكتب العناصر بين القوسين () ويجب عليك وضع فاصلة , بين كل عنصر. Create a **Tuple**.

```
thistuple = ("apple", "banana", "cherry")  
print(thistuple)
```



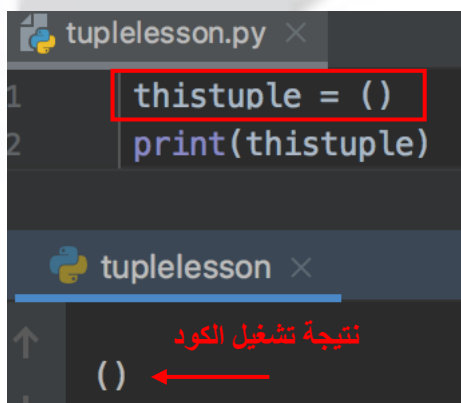
```
Python 3.7.2 Shell  
File Edit Shell Debug Options Win  
Python 3.7.2 (tags/v3.7.2:9a3c743d, Jul 9 2019, 16:08:01) [AMD64] on win32  
Type "help", "copyright", "credits()" or "quit()" to interact with the interpreter.  
>>>  
== RESTART ==>  
>>> thistuple = ("apple", "banana", "cherry")  
>>> print(thistuple)  
>>> ('apple', 'banana', 'cherry')  
>>>
```

نتيجة تشغيل الكود

Example

An empty **tuple** in **Python**.

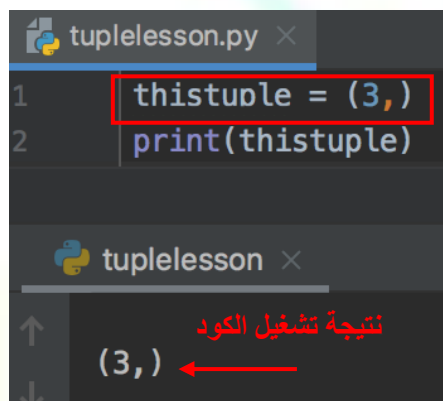
```
thistuple = ()
print (thistuple)
```



في المثال التالي قمنا بتعريف **tuple** فارغ ، لا يحتوي على أي عنصر.

Example

```
thistuple = (3,)
print (thistuple)
```



في المثال التالي قمنا بتعريف **tuple** يحتوي على عنصر واحد فقط.

إذا كان الـ **tuple** يحتوي على عنصر واحد فقط، يجب عليك وضع فاصلة , بعد القيمة كما في المثال

حتى يتم التفريق بين **tuple** و المتغير العادي في بايثون.

Example

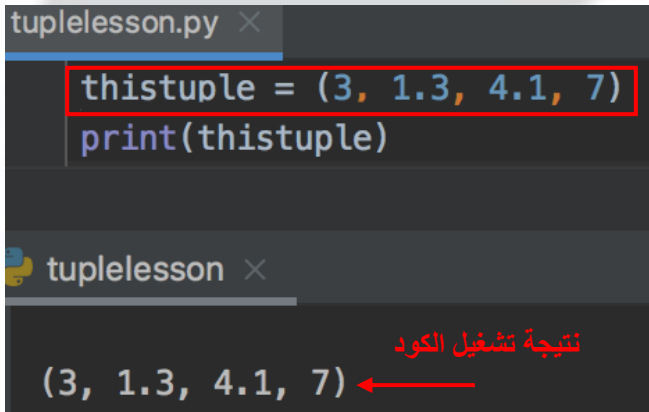
For multiple items.

```
thistuple = (3, 1.3, 4.1, 7)
print(thistuple)
```

في المثال التالي قمنا بتعريف tuple يحتوي على أعداد صحيحة وعشرية

هنا لا حاجة لوضع فاصلة , إضافية بعد آخر قيمة كما في المثال السابق

لأن ال tuple تحتوي على أكثر من عنصر



```
tuplelesson.py ×
thistuple = (3, 1.3, 4.1, 7)
print(thistuple)

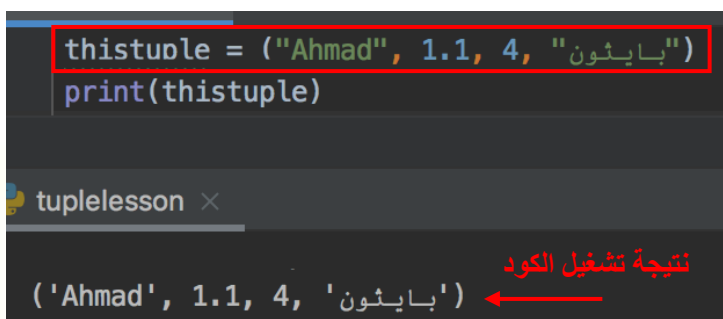
tuplelesson ×
نتيجة تشغيل الكود
(3, 1.3, 4.1, 7) ←
```

Example

The data inside a **tuple** can be of one or more data types.

```
thistuple = ("Ahmad", 1.1, 4, "بايثون")
print(thistuple)
```

في المثال التالي قمنا بتعريف tuple يحتوي على عدة أنواع من أنواع البيانات: أعداد صحيحة، أعداد عشرية، ونصوص.



```
tuplelesson.py ×
thistuple = ("Ahmad", 1.1, 4, "بايثون")
print(thistuple)

tuplelesson ×
نتيجة تشغيل الكود
('Ahmad', 1.1, 4, 'بايثون') ←
```

➤ Access Tuple Items

الوصول لعناصر الـ tuple

للوصول لأي عنصر في tuple عليك استخدام رقم الـ **index** الخاص بالعنصر، باستخدام الأقواس المربعة []

You can access **tuple** items by referring to the **index** number, inside square brackets [].

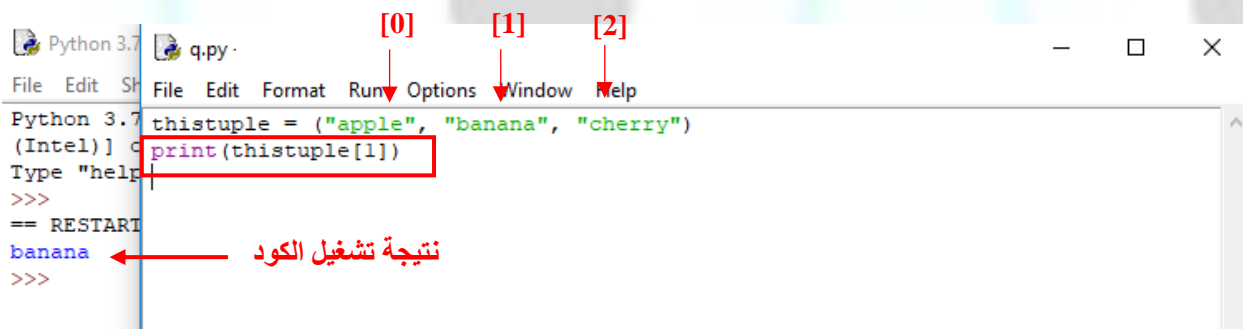
Example

يمكنك الوصول للعناصر الموجودة في tuple كما في القوائم تمامًا.

Return the item in position 1.

```
thistuple = ("apple", "banana", "cherry")
print(thistuple[1])
```

هنا قمنا بعرض قيمة العنصر الثاني



```
Python 3.7 q.py
File Edit Shell Format Run Options Window Help
Python 3.7 thistuple = ("apple", "banana", "cherry")
(Intel) | print(thistuple[1])
Type "help()
>>>
== RESTART ==>
banana
>>>
```

تغيير قيمة العناصر في الـ tuple

➤ Change Tuple Values

Once a **tuple** is created, you cannot change its values. **Tuples** are unchangeable.

في الـ **tuple** لا يمكنك تغيير القيم التي تسندها إليها أثناء إنشائها، فهي غير قابلة للتغيير أو التعديل عليها.

عرض جميع القيم في الـ tuple

➤ Loop Through a Tuple

You can loop through the **tuple** items by using a **for** loop.

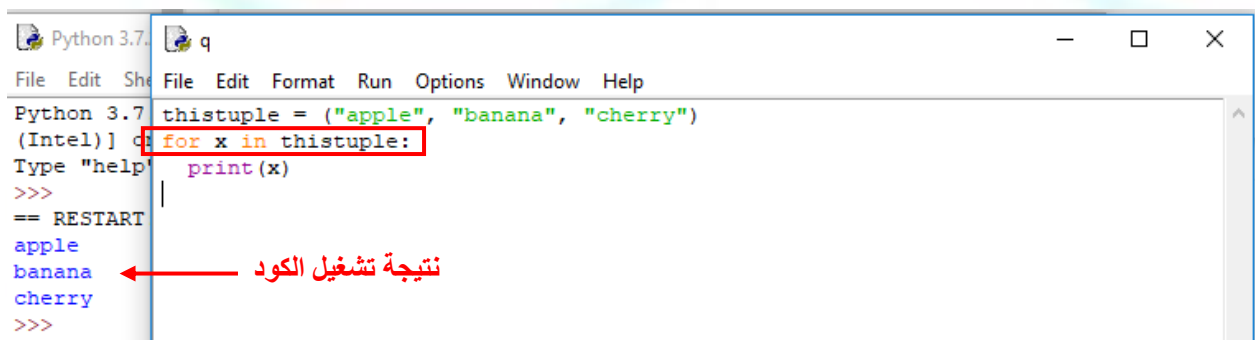
Example

هنا قمنا بتعريف **tuple** تحتوي على نصوص
ثم قمنا بطباعة جميع القيم باستخدام الحلقة **for**

Iterate through the items and print the values.

```
thistuple = ("apple", "banana", "cherry")
for x in thistuple:
    print(x)
```

في كل مرة سيتم وضع قيمة عنصر من عناصر الـ **tuple** في المتغير **x** ثم سيتم طباعتها
سيتم شرح الحلقة **for** في دروس قادمة بإذن الله.



```
Python 3.7.4 Shell
File Edit Shell File Edit Format Run Options Window Help
Python 3.7.4 Shell
>>> thistuple = ("apple", "banana", "cherry")
>>> for x in thistuple:
>>>     print(x)
>>>
== RESTART ==
apple
banana
cherry
>>>
```


➤ Add Items

إضافة عنصر جديد في الـ tuple

Note: you cannot add items to it.

Once a **tuple** is created, **tuples** are unchangeable.

في الـ tuple القيم غير قابلة للتغيير أو التعديل عليها.

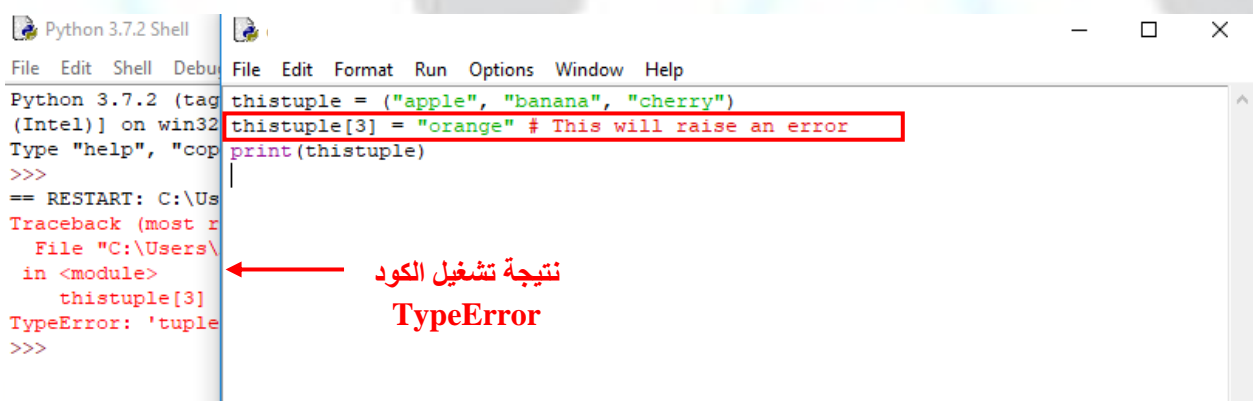
Example

You cannot add items to a **tuple**.

```
thistuple = ("apple", "banana", "cherry")
thistuple[3] = "orange" # This will raise an error
print(thistuple)
```

هنا لن تتمكن من إضافة عنصر رابع للـ tuple سيظهر لك **error**

كما هو موضح بالمثال التالي



```
Python 3.7.2 Shell
File Edit Shell Debug
Python 3.7.2 (tags/v3.7.2:1a3f604, Dec 14 2019) on win32
Type "help()", "copyright()", "credits()", "license()", "sys.version()", "sys.platform()"
>>>
== RESTART: C:\Users\...
Traceback (most recent call last):
  File "C:\Users\...
    in <module>
      thistuple[3]
TypeError: 'tuple' object does not support item assignment
>>>
```

نتيجة تشغيل الكود
TypeError

➤ Remove Item

حذف العناصر في الـ tuple

Note: You cannot remove items in a **tuple**.

- **Tuples** are unchangeable, so you cannot remove items from it,
- but you can delete the **tuple** completely.

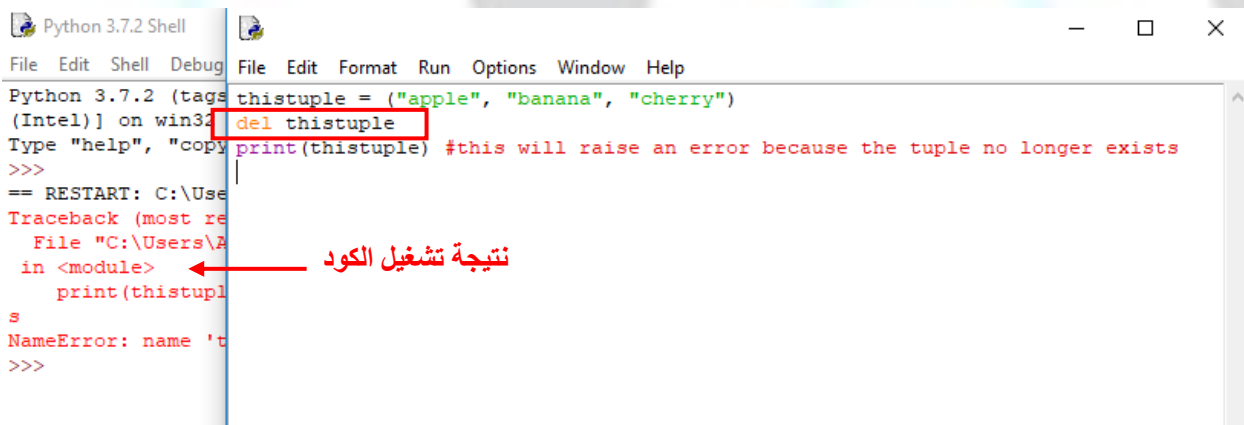
لا يمكنك حذف عنصر في الـ tuple
تستطيع فقط حذف كامل الـ tuple

Example

في هذا المثال قمنا بحذف الـ tuple كاملاً.

The **del** keyword can delete the **tuple** completely.

```
thistuple = ("apple", "banana", "cherry")
del thistuple
print(thistuple) #this will raise an error because the tuple no longer exists
```



```
Python 3.7.2 Shell
File Edit Shell Debug File Edit Format Run Options Window Help
Python 3.7.2 (tags/b3d9000f, Sep 17 2019, 12:50:34) [AMD64] on win32
Type "help", "copyright()", "credits()", or "quit()" for more
>>> thistuple = ("apple", "banana", "cherry")
>>> del thistuple
>>> print(thistuple) #this will raise an error because the tuple no longer exists
Traceback (most recent call last):
  File "C:\Users\user\AppData\Local\Programs\Python\Python37-32\Python.exe", line 1, in <module>
    print(thistuple)
NameError: name 'thistuple' is not defined
>>>
```

نتيجة تشغيل الكود

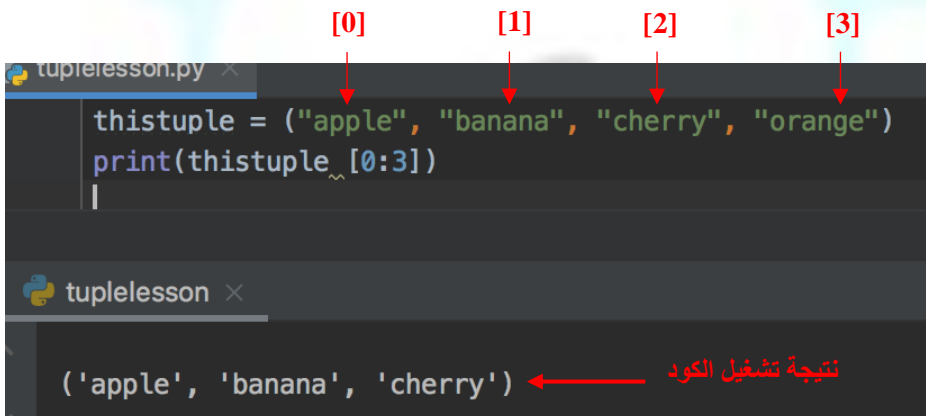
تقسيم الـ tuple إلى أجزاء / تجزئة الـ tuple

إذا أردنا عرض أو طباعة جزء من عناصر الـ tuple نقوم بعرض العناصر كما في القوائم عن طريق رقم الـ **index**

Example

```
thistuple = ("apple", "banana", "cherry", "orange")  
print(thistuple [0:3])
```

في هذا المثال قمنا بتعريف tuple ثم قمنا بطباعة العناصر الثلاثة الأولى فقط.



```
tuplelesson.py ×  
thistuple = ("apple", "banana", "cherry", "orange")  
print(thistuple [0:3])  
|  
  
tuplelesson ×  
  
( 'apple', 'banana', 'cherry' ) ← نتيجة تشغيل الكود
```

أتممت درسك بنجاح!
واصل التعلم

طبّق ما تعلمته في هذا الدرس
ولا تنسى مشاركتنا أكوادك

اليوم السابع عشر

الصفوف (الأزواج المرتبة) في لغة البايثون

Python Tuples 2

➤ Check if Item Exists

التحقق من وجود عنصر في الـ tuple

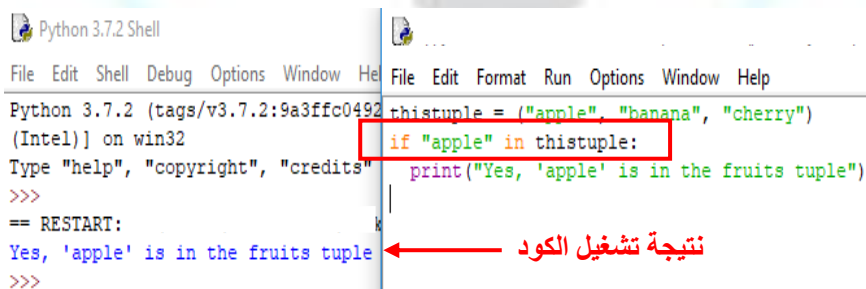
To determine if a specified item is present in a **tuple** use the **in** keyword.

Example

في المثال التالي استخدمنا العامل **in** للبحث عن قيمة معينة في الـ tuple

Check if "apple" is present in the **tuple**.

```
thistuple = ("apple", "banana", "cherry")
if "apple" in thistuple:
    print("Yes, 'apple' is in the fruits tuple")
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 11 2019) [AMD64] on win32
Type "help", "copyright", "credits()" or "quit()" for more
>>> thistuple = ("apple", "banana", "cherry")
>>> if "apple" in thistuple:
>>>     print("Yes, 'apple' is in the fruits tuple")
>>>
== RESTART: ==
Yes, 'apple' is in the fruits tuple
>>>
```

➤ Repeat Item

To repeat an item in a **tuple**, use the ***** operator.

لتكرار قيمة معينة نستخدم العامل *

Example

في هذا المثال قمنا بإنشاء **tuple**

ووضعنا فيه ثلاث قيم نصية "بايثون" باستخدام العامل *

لاحظ: يجب عليك وضع الفاصلة بعد القيمة التي داخل القوس لأنها هنا تعتبر كعنصر واحد، حتى يميز مترجم بايثون أنك تريد تعريف **tuple**

```
thistuple = ("بايثون",) * 3
print(thistuple)
```

نتيجة تشغيل الكود

```
('بايثون', 'بايثون', 'بايثون')
```

جرب ألا تضع الفاصلة واطبع النتيجة ..

➤ + Operator in Tuple

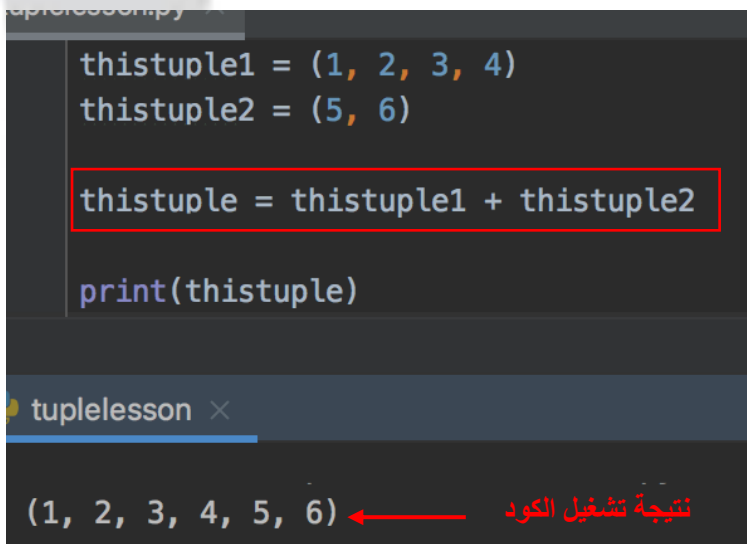
To add 2 **tuple**s or more into one **tuple**.

يُستخدم هذا العامل + لدمج أكثر من **tuple** كما في القوائم.

Example

ألق نظرة على هذا المثال

```
thistuple1 = (1, 2, 3, 4)
thistuple2 = (5, 6)
thistuple = thistuple1 + thistuple2
print(thistuple)
```



```
thistuple1 = (1, 2, 3, 4)
thistuple2 = (5, 6)
thistuple = thistuple1 + thistuple2
print(thistuple)
```

tuplelesson ×

(1, 2, 3, 4, 5, 6) ← نتيجة تشغيل الكود

طبّق هذا المثال لتتأكد من فهمك

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

➤ Tuple Length

عدد عناصر الـ tuple

استخدم الدالة `len()` كما في درس القوائم سابقًا، لمعرفة عدد العناصر، فهي تقوم بترجيع عدد صحيح.

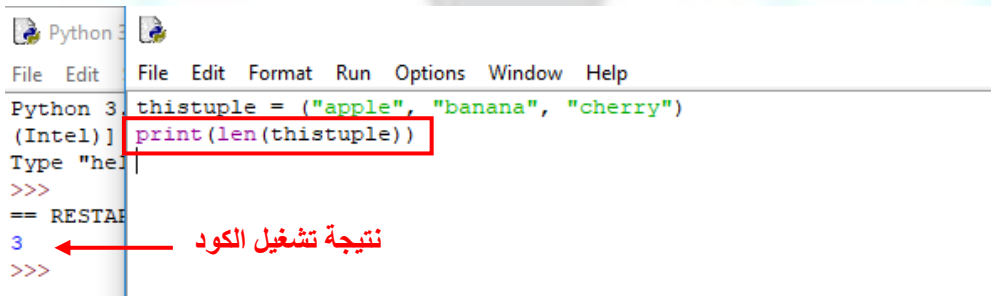
To determine how many items a **tuple** has, use the `len()` method.

Example

Print the number of items in the **tuple**.

```
thistuple = ("apple", "banana", "cherry")  
print(len(thistuple))
```

في هذا المثال الـ tuple يحتوي على ٣ عناصر.



```
Python 3.10.0 (tags/v3.10.0:2016-10-24) [AMD64] on win32  
File Edit Format Run Options Window Help  
Python 3.10.0 thistuple = ("apple", "banana", "cherry")  
(Intel) print(len(thistuple))  
Type "help()" for more  
>>>  
3  
== RESTART ==>>>
```

نتيجة تشغيل الكود

➤ The **tuple()** Constructor

بناء tuple / صف

يمكنك أيضا إنشاء tuple باستخدام الدالة **tuple()**.

It is also possible to use the **tuple()** constructor to make a **tuple**.

Example

Using the **tuple()** method to make a **tuple**.

في هذا المثال قمنا بتعريف متغير وهو **thistuple**

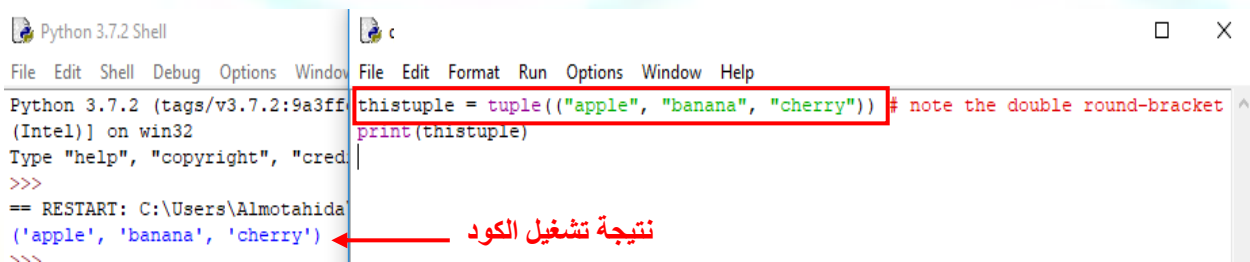
وقمنا باستخدام الدالة **tuple()** ووضعنا فيها مجموعة عناصر وهنا العناصر تمثل **tuple**

لذلك تلاحظ أن هناك أقواس () مضاعفة.

١/ الأقواس الأولى تمثل مكان المُعامِلات (الباراميتِر) للدالة **tuple()**.

٢/ الأقواس الثانية تمثل الـ **tuple**.

```
thistuple = tuple(("apple", "banana", "cherry")) # note the double round-brackets
print(thistuple)
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3fff, Dec 12 2019) on win32
Type "help", "copyright", "credits()" or "quit()" for more
>>> thistuple = tuple(("apple", "banana", "cherry")) # note the double round-bracket
>>> print(thistuple)
('apple', 'banana', 'cherry')
```

نتيجة تشغيل الكود

باختصار فإن الدالة **tuple()** تقوم بترجييع/تحويل مجموعة العناصر المرسله لها إلى **tuples**

قم بتطبيق هذا المثال وستتضح لك الفكرة أكثر والغرض من هذه الدالة **tuple()**.

```
thislist = [3, 4, 5, 6, "A", "B"]  
thistuple = tuple(thislist)  
print(thistuple)
```

حيث أن الدالة **tuple()** هنا تستقبل مجموعة عناصر القائمة

وبالتالي الدالة **tuple()** ستقوم بتحويل هذه العناصر من قائمة إلى صف **tuple**

➤ Tuple Methods

هنا أيضاً دوال جاهزة في بايثون للتعامل مع الـ **tuple**

Method	Description
<u>count()</u>	Returns the number of times a specified value occurs in a tuple
<u>index()</u>	Searches the tuple for a specified value and returns the position of where it was found

ابحث وتعرف على وظائفها وقم بتطبيق أمثلة عليها لتكون قد غطيت أغلب الدوال الخاصة بالـ **tuple**.

رائع!
أتممت درسك الأخير لهذا الأسبوع

روابط قد تساعدك

Check the links below

- [Python tuple](#)
- [Python Programming #7 - Tuples](#)
- [Learn Python Programming - 34 - Tuples](#)
- [السلاسل Python lists and tuples -8](#)
- [08 - Python - Beginners Tutorial - Tuples](#)
- [Learn Python in Arabic #17 - 17 شرح بايثون بالعربي Python in Arabic tuple متغير صف](#)

طبّق ما تعلمته في هذا الدرس
ولا تنسى مشاركتنا أكوادك

اليوم الثامن عشر & اليوم التاسع عشر

تحدي الأسبوع (يتم حله ورفعته على Github)

أولاً:

قم بإنشاء قائمة وطبق عليها ٤ دوال مما تعلمته خلال هذا الأسبوع

Method	Description
<code>append()</code>	Adds an element at the end of the list
<code>clear()</code>	Removes all the elements from the list
<code>copy()</code>	Returns a copy of the list
<code>count()</code>	Returns the number of elements with the specified value
<code>extend()</code>	Add the elements of a list (or any iterable), to the end of the current list
<code>index()</code>	Returns the index of the first element with the specified value
<code>insert()</code>	Adds an element at the specified position
<code>pop()</code>	Removes the element at the specified position
<code>remove()</code>	Removes the item with the specified value
<code>reverse()</code>	Reverses the order of the list
<code>sort()</code>	Sorts the list

ثانياً:

قم بكتابة نص برمجي (كود) لتتأكد عن طريقه من وجود العنصر **python** في `tuple = ("java", "python", "swift")`

انتظرونا في دروس الأسبوع القادم