

SEPTEMBER 8 – 14, 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 العشرون

المجموعات في لغة البايثون

Python Sets

➤ Set

المجموعة

A **set** is a collection which is unordered and unindexed.

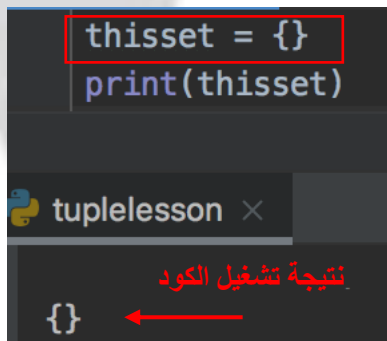
In **Python** sets are written with curly brackets {}.

هي عبارة عن مصفوفة أيضاً، ليس لها حجم ثابت. غير منظمة/ غير مرتبة، وعناصرها ليس لها رقم الـ **index** كما كان للقوائم والصفوف. لإنشاء/تعريف مجموعة نستخدم الأقواس المعقوفة {} ونكتب القيم ثم نفصل بينهم بالفاصلة ،

Example: Empty Set.

```
thisset = {}
print(thisset)
```

هنا مثال لمجموعة فارغة

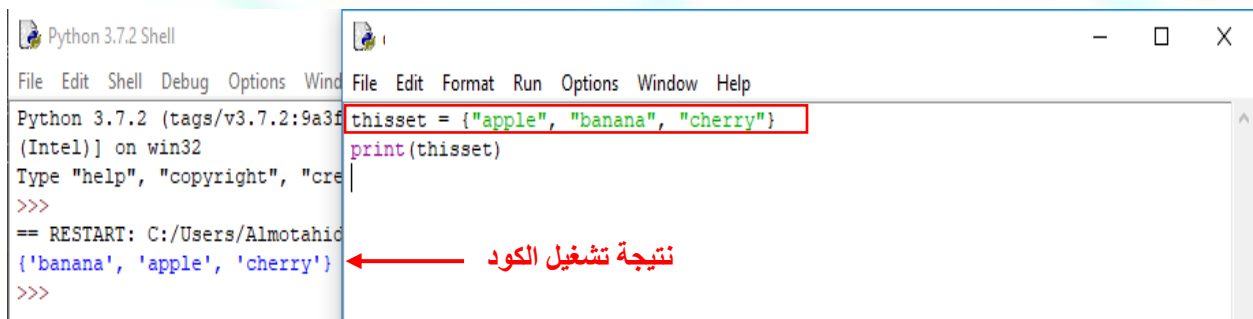


Example: Create a Set.

```
thisset = {"apple", "banana", "cherry"}
print(thisset)
```

هنا قمنا بإنشاء مجموعة وأسندنا إليها قيم نصية.

المجموعات غير مرتبة كما قلنا، لذا لا نعلم كيف سيتم ترتيب العناصر أثناء لطباعة



Note: Sets are unordered, so you cannot be sure in which order the items will appear.

Example

في هذا المثال

```
thisset = {"Ahmad", "Ahmad", 1, 2, 1, 5}
print(thisset)
```

أردنا توضيح أن المجموعات لا تحتوي على قيم مكررة، قيمها تكون دائما فريدة. أي أنها لا تقوم بتخزين نفس القيمة أكثر مرة. ولا ننسى أيضا أنها غير مرتبة لذلك ظهرت النتيجة بهذه الصورة.

```
thisset = {"Ahmad", "Ahmad", 1, 2, 1, 5}
print(thisset)
```

tuplelesson ×

نتيجة تشغيل الكود ← {1, 2, 5, 'Ahmad'}

➤ Access Items

الوصول لعناصر المجموعة

You cannot access items in a **set** by referring to an index, since **sets** are unordered the items has no index.

لا يمكن الوصول للعناصر عن طريق الـ **index** كما كنا نفعل مع القوائم والصفوف، لأن المجموعات غير منظمة.

But you can loop through the **set** items using a **for** loop, or ask if a specified value is present in a **set**, by using the **in** keyword.

لكن يمكننا الوصول لعناصر المجموعة باستخدام الحلقة **for** لعرض جميع القيم في المجموعة. أو باستخدام العامل **in** للبحث عن قيمة معينة، كما في القوائم والصفوف.

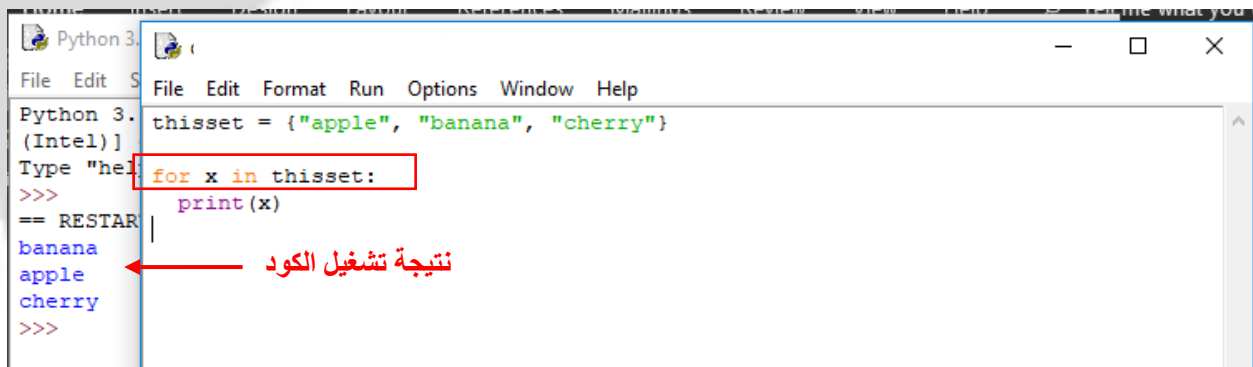
Example: Loop through the **set** and print the values.

```
thisset = {"apple", "banana", "cherry"}

for x in thisset:
    print(x)
```

في هذا المثال قمنا بتعريف مجموعة **thisset** تحمل قيم نصية
ثم قمنا بطباعة وعرض القيم باستخدام الحلقة **for**

العامل **in** يُستخدم هنا للمرور على القيم عند استخدامه في حلقة **for**



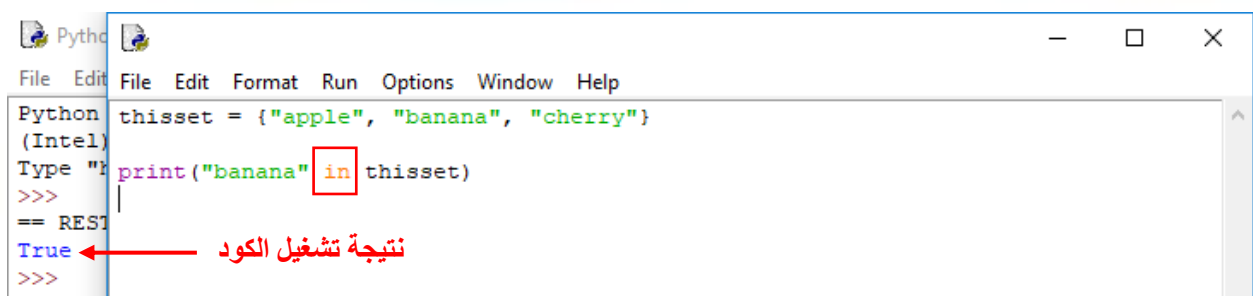
نتيجة تشغيل الكود

Example: Check if "banana" is present in the set.

```
thisset = {"apple", "banana", "cherry"}

print("banana" in thisset)
```

هنا استخدمنا العامل **in** للبحث عن قيمة معينة في المجموعة **thisset**



نتيجة تشغيل الكود

تعديل/تغيير عناصر المجموعة

➤ Change Items

Once a **set** is created, you cannot change its items, but you can add new items.

عند إنشائك لمجموعة فإنك لن تستطيع تغيير قيم العناصر فيما بعد، لكن يمكنك فقط إضافة عناصر جديدة.

إضافة عنصر/عناصر للمجموعة

➤ Add Items

To add one item to a **set** use the **add()** method.

الدالة **add()** لإضافة عنصر واحد

To add more than one item to a **set** use the **update()** method.

الدالة **update()** لإضافة أكثر من عنصر

Example

Add an item to a **set**, using the **add()** method.

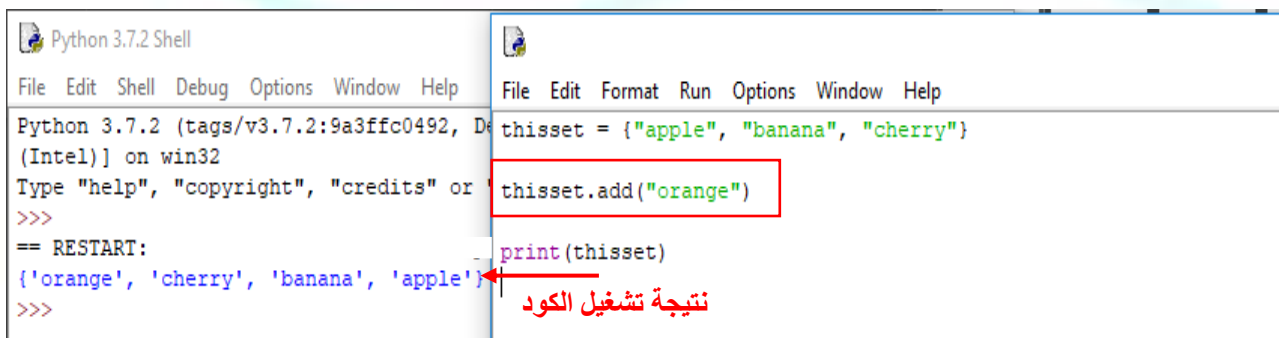
```
thisset = {"apple", "banana", "cherry"}

thisset.add("orange")

print(thisset)
```

هنا أضفنا عنصر جديد للمجموعة **thisset**

باستخدام الدالة **add()**



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 10 2019, [AMD64] on win32)
Type "help", "copyright", "credits" or "quit()"
>>>
== RESTART:
{'orange', 'cherry', 'banana', 'apple'}
>>>
```

ملاحظة: الدالة **add()** لا تقبل سوى عنصر واحد في كل مرة.

Example

Add multiple items to a **set**, using the **update()** method.

```
thisset = {"apple", "banana", "cherry"}
```

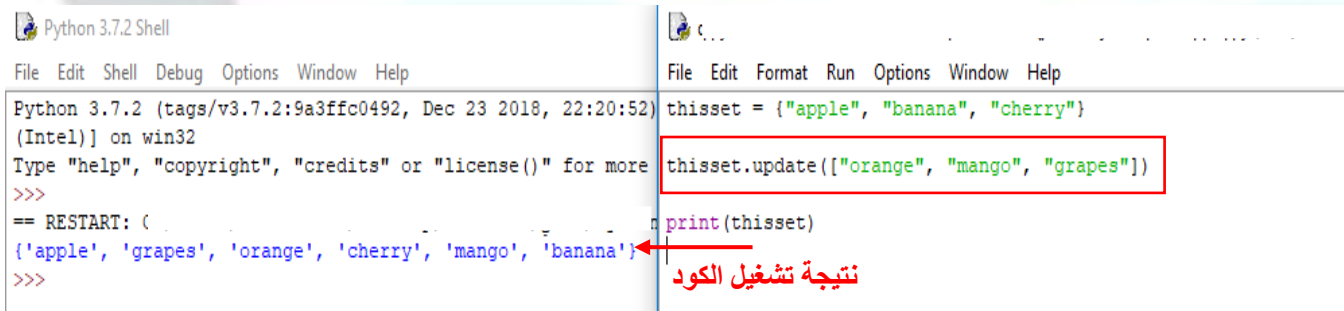
هنا أضفنا عدة عناصر مرة واحدة للمجموعة **thisset**

```
thisset.update(["orange", "mango", "grapes"])
```

باستخدام الدالة **update()**

```
print(thisset)
```

لاحظ: أن العناصر المضافة يجب أن تكون داخل قوسين مربعين []



The screenshot shows two windows. The left window is a Python 3.7.2 Shell with the following output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52)
(Intel) on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
== RESTART: (
{'apple', 'grapes', 'orange', 'cherry', 'mango', 'banana'}
>>>
```

The right window is a code editor showing the following code:

```
thisset = {"apple", "banana", "cherry"}
thisset.update(["orange", "mango", "grapes"])
print(thisset)
```

A red box highlights the `thisset.update(["orange", "mango", "grapes"])` line in the code editor. A red arrow points from the text "نتيجة تشغيل الكود" (Code execution result) to the output of the `print` statement in the shell.

أفضل طريقة للفهم هي التطبيق

أتمت درسك بنجاح!

طبّق ما تعلمته في هذا الدرس

ولا تنسى مشاركتنا أكوادك

اليوم الواحد والعشرون

المجموعات في لغة البايثون

Python Sets 2

➤ Get the Length of a Set

طول المجموعة/عدد عناصر المجموعة

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

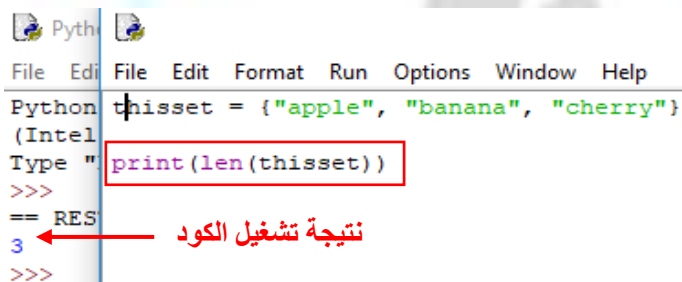
استخدم الدالة **len()** لمعرفة عدد عناصر المجموعة، فهي تقوم بترجيع عدد صحيح كما في الدروس السابقة القوائم والصفوف.

Example

Get the number of items in a **set**.

```
thisset = {"apple", "banana", "cherry"}  
  
print(len(thisset))
```

في هذا المثال المجموعة تحتوي على ٣ عناصر.



```
Python  
File Edit Format Run Options Window Help  
Python thisset = {"apple", "banana", "cherry"}  
(Intel print(len(thisset))  
Type ">>>  
== RES  
3  
>>>
```

نتيجة تشغيل الكود

➤ Remove Item

حذف عنصر في المجموعة

لحذف عنصر من مجموعة ما استخدم إحدى هاتين الدالتين **remove()** أو **discard()**.

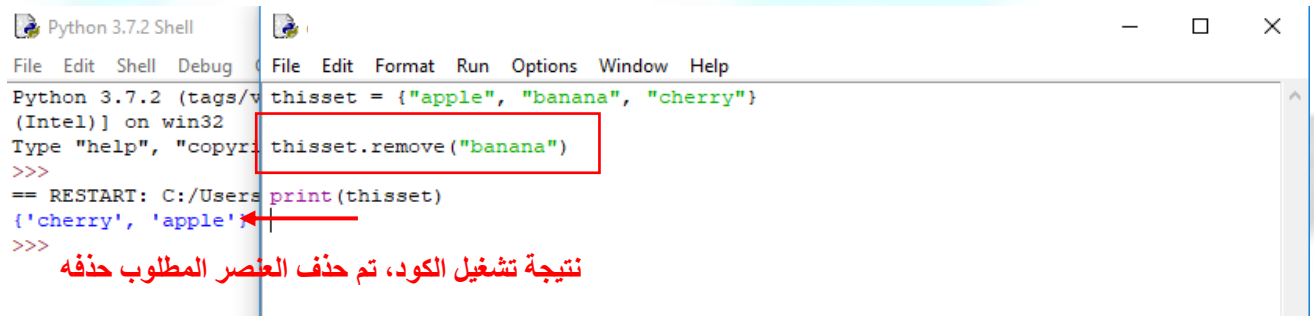
To remove an item in a **set**, use the **remove()**, or the **discard()** method.

Example

Remove "banana" by using the **remove()** method.

```
thisset = {"apple", "banana", "cherry"}  
  
thisset.remove("banana")  
  
print(thisset)
```

في هذا المثال استخدمنا دالة **remove()** لحذف عنصر محدد في المجموعة وهو "banana"



```
Python 3.7.2 Shell  
File Edit Shell Debug File Edit Format Run Options Window Help  
Python 3.7.2 (tags/v3.7.2:1 c3ae93d, Oct 9 2019) on win32  
Type "help", "copyright()", "credits()", or "quit()" for more help  
>>> thisset = {"apple", "banana", "cherry"}  
>>> thisset.remove("banana")  
>>> print(thisset)  
{'cherry', 'apple'}  
>>>
```

نتيجة تشغيل الكود، تم حذف العنصر المطلوب حذفه

انتبه!

في حال لم يتم إيجاد العنصر المراد حذفه، والذي تم تمريره مكان (الباراميتر) في الدالة، سيُظهر لك البرنامج رسالة خطأ **error**.

Note: If the item to remove does not exist, **remove()** will raise an **error!**.

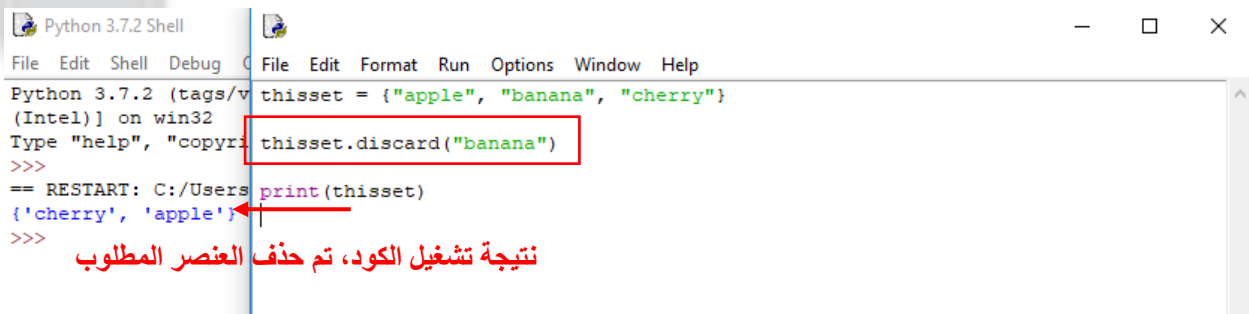
جَرِّبْ يا مبرمج!

Example

Remove "banana" by using the **discard()** method.

```
thisset = {"apple", "banana", "cherry"}  
  
thisset.discard("banana")  
  
print(thisset)
```

أيضاً في هذا المثال استخدمنا الدالة **discard()** لحذف عنصر محدد في المجموعة وهو "banana"



```
Python 3.7.2 Shell  
File Edit Shell Debug C File Edit Format Run Options Window Help  
Python 3.7.2 (tags/v3.7.2:1a7ff97, Jul 10 2019, [AMD64]) on win32  
Type "help", "copyright()", "credits()", "license()", "quit()", or "exit()" for more details.  
>>> thisset = {"apple", "banana", "cherry"}  
>>> thisset.discard("banana")  
>>> print(thisset)  
{'cherry', 'apple'}  
>>>
```

نتيجة تشغيل الكود، تم حذف العنصر المطلوب

لاحظ! هنا في هذه الدالة إذا لم يتم إيجاد العنصر المراد حذفه، لن يُظهر لك البرنامج رسالة خطأ **error** عند التنفيذ.

Note: If the item to remove does not exist, **discard()** will **NOT** raise an error.

الدالة **pop()** تُستخدم لحذف العنصر الأخير، لكن بما أن المجموعة عبارة عن جمع غير منظم/مرتب، إذاً سوف تقوم الدالة بحذف عنصر ما في المجموعة بشكل عشوائي.

You can also use the **pop()** method to remove an item, but this method will remove the last item. Remember that **sets** are unordered, so you will not know what item that gets removed.

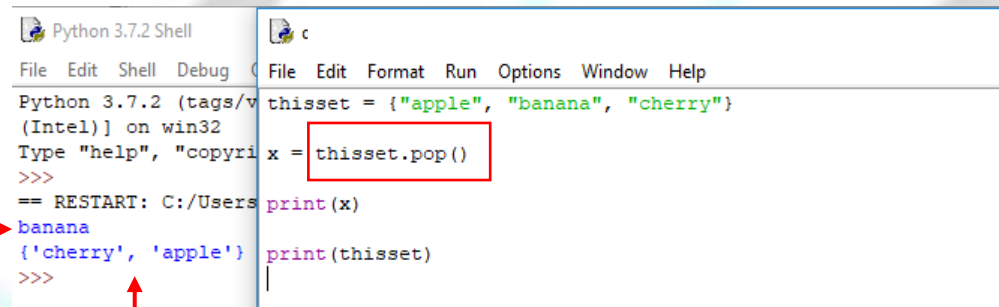
الدالة ترجع العنصر الذي تم حذفه من المجموعة. The return value of the **pop()** method is the removed item.

Example

قم بتطبيق هذا المثال لتتضح لك وظيفة هذه الدالة

Remove the last item by using the **pop()** method.

```
thisset = {"apple", "banana", "cherry"}
x = thisset.pop()
print(x)
print(thisset)
```



→ العنصر الذي تم اختياره للحذف

نتيجة تشغيل الكود بعد حذف العنصر

Note: Sets are unordered, so when using the **pop()** method, you will not know which item that gets removed.

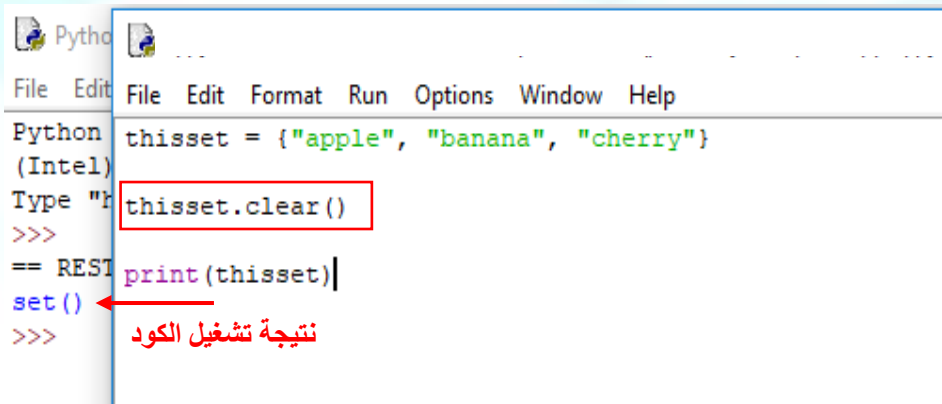
لاحظ! بما أن المجموعات غير مرتبة، فإننا لا نعلم أي العناصر سيتم حذفها عند استخدام الدالة **pop()**

Example

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

دالة **clear()** تُستخدم لحذف جميع عناصر المجموعة.

```
thisset = {"apple", "banana", "cherry"}  
  
thisset.clear()  
  
print(thisset)
```



```
Python ...  
File Edit Format Run Options Window Help  
Python (Intel)  
Type "r" to run the file  
>>> thisset = {"apple", "banana", "cherry"}  
>>> thisset.clear()  
>>> print(thisset)  
set()  
>>> نتيجة تشغيل الكود
```


Example

The **del** keyword will delete the **set** completely.

جملة **del** ستقوم بحذف المجموعة كاملة من الذاكرة.

```
thisset = {"apple", "banana", "cherry"}  
  
del thisset  
  
print(thisset)
```

بالتالي سيظهر لك خطأ **error** عند التشغيل

#this will raise an error because the set no longer exists

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

← نتيجة تشغيل الكود

➤ The **set()** Constructor

بناء/تكوين مجموعة باستخدام الدالة **set()**.

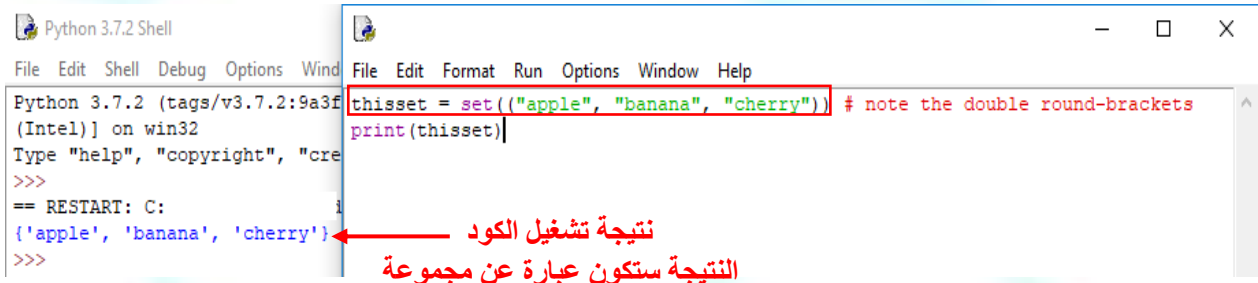
It is also possible to use the **set()** constructor to make a **set**.

Example

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

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

نستدعي الدالة **set()** ثم نضع بداخلها العناصر/القيم داخل أقواس مستديرة () بدلا من أقواس المجموعة {}



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3f503, Dec 12 2019) on win32
Type "help", "copyright", "credits()" or "quit()" for more
>>> thisset = set(("apple", "banana", "cherry")) # note the double round-brackets
print(thisset)
{'apple', 'banana', 'cherry'}
>>>
```

نتيجة تشغيل الكود
النتيجة ستكون عبارة عن مجموعة

➤ Set Methods

دوال المجموعة في لغة بايثون

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

بايثون تحتوي على دوال جاهزة يمكنك استخدامها للتعامل مع المجموعات وقد ناقشنا بعضها في هذا الدرس.

Method	Description
<u>add()</u>	Adds an element to the set
<u>clear()</u>	Removes all the elements from the set
<u>copy()</u>	Returns a copy of the set
<u>difference()</u>	Returns a set containing the difference between two or more sets
<u>difference_update()</u>	Removes the items in this set that are also included in another, specified set
<u>discard()</u>	Remove the specified item
<u>intersection()</u>	Returns a set, that is the intersection of two other sets
<u>intersection_update()</u>	Removes the items in this set that are not present in other, specified set(s)
<u>isdisjoint()</u>	Returns whether two sets have a intersection or not
<u>issubset()</u>	Returns whether another set contains this set or not
<u>issuperset()</u>	Returns whether this set contains another set or not
<u>pop()</u>	Removes an element from the set
<u>remove()</u>	Removes the specified element
<u>symmetric_difference()</u>	Returns a set with the symmetric differences of two sets
<u>symmetric_difference_update()</u>	inserts the symmetric differences from this set and another
<u>union()</u>	Return a set containing the union of sets
<u>update()</u>	Update the set with the union of this set and others

أتممت درسك بنجاح!
تابع التقدّم

روابط قد تساعدك

Check the links below

- [Python set](#)
- [في بايثون set المجموعات](#)
- [#6 Python Tutorial for Beginners | Tuple | Set in Python](#)
- [Python Tutorial for Beginners 15 - Python Sets](#)
- [Sets in Python || Python Tutorial || Learn Python Programming](#)

طبّق ما تعلمته في هذا الدرس
ولا تنسى مشاركتنا أكوادك

اليوم الثاني والعشرون

القواميس (المعاجم) في لغة البايثون

Python Dictionaries

➤ Dictionary

القاموس

القاموس هو جمعٌ غير مرتب ومتغير، وكل عنصر له مفتاح وقيمة، والمفاتيح تستخدم للوصول إلى القيم، والقواميس مفهرسة بواسطة المفتاح. نعرّف القاموس بواسطة الرمز {} الأقواس المعقوفة

A **dictionary** is a collection which is unordered, changeable and indexed.

In **Python** dictionaries are written with curly brackets {}, and they have keys and values.

Example

an Empty **dictionary**.

```
thisdict = {}  
print(thisdict)
```

tuplelesson ×

نتيجة تشغيل الكود → {}

لا بد أنك لاحظت، أنه لا فرق بين القاموس الفارغ والمجموعة الفارغة.

لكن بمجرد أن تضيف العناصر أو على الأقل عنصراً واحداً فقط، ستفرق **البايثون** بين القاموس و المجموعة. لأن لكل واحد منهم كتابة معينة، أثناء التعريف.

dictionary name = {keys : values}

نكتب القاموس بهذا الشكل {القيمة : المفتاح} و يتم الإسناد إلى متغير

Example

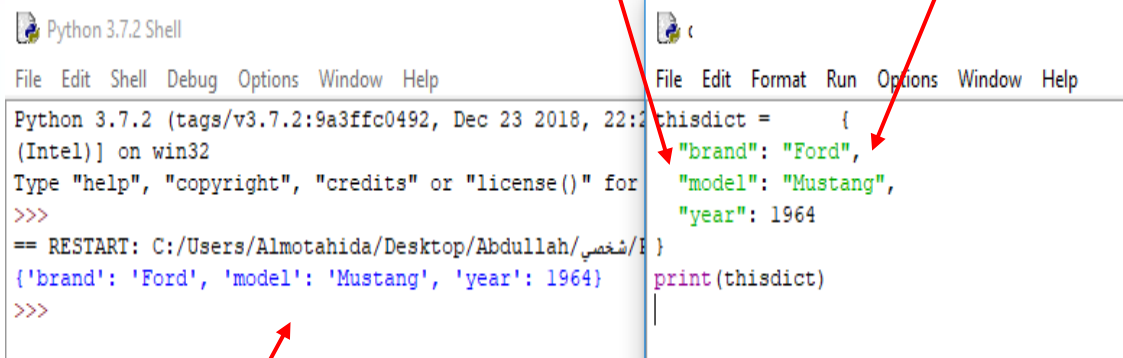
Create and print a **dictionary**.

هنا قمنا بتعريف قاموس **thisdict** يتكون من ٣ عناصر وتعريف كل عنصر على سطر منفرد للترتيب فقط.

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
print(thisdict)
```

كل عنصر يجب أن يملك قيمتين، المفتاح والقيمة، وبين كل مفتاح وقيمة نكتب الرمز :
يتم الفصل بين العناصر بواسطة الفاصلة ,

المفاتيح هنا وضعناها كنصوص، والقيم نصوص وأرقام.



The screenshot shows a Python 3.7.2 Shell window with the following code and output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:21:04) [Intel] on win32  
Type "help", "copyright", "credits" or "license()" for more  
>>>  
== RESTART: C:/Users/Almotahida/Desktop/Abdullah/Python/Python 3.7.2 Shell  
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}  
>>>
```

مفاتيح

قيم

نتيجة تشغيل الكود

➤ Accessing Items

الوصول للعناصر في القاموس

You can access the items of a **dictionary** by referring to its key name, inside square brackets [].

للاوصول للعناصر نستخدم المفتاح الخاص بالعنصر، ونضع هذا المفتاح داخل أقواس مربعة [].

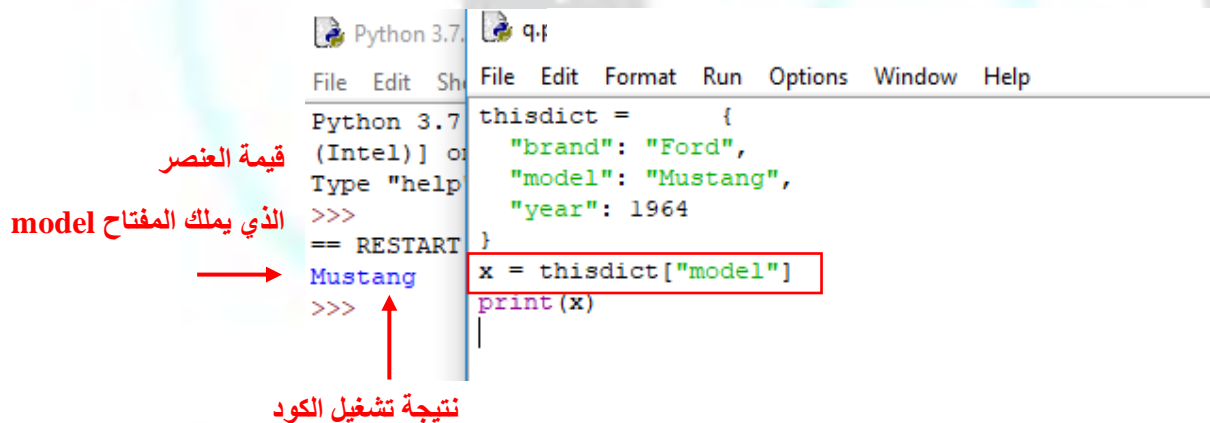
Example

Get the value of the "model" key.

هنا في هذا المثال

نريد عرض وطباعة قيمة العنصر الذي يملك المفتاح **model**

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
x = thisdict["model"]
print(x)
```



يمكنك أيضا استخدام دالة `get()` للوصول وعرض قيمة العنصر، وستعطيك نفس النتيجة.

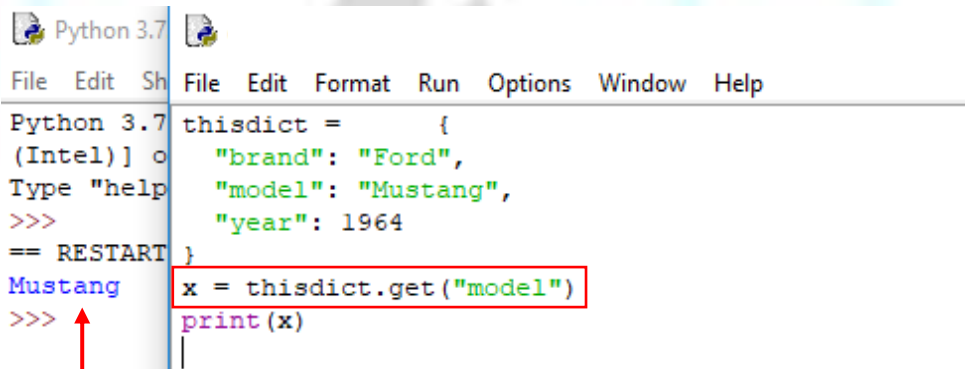
There is also a method called `get()` that will give you the same result.

Example

Get the value of the "model" key.

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
x = thisdict.get("model")  
print(x)
```

هنا قمنا بعرض وطباعة قيمة العنصر الذي يملك المفتاح `model`
باستخدام الدالة `get()`



```
Python 3.7  
File Edit Shell File Edit Format Run Options Window Help  
Python 3.7 (Intel) o  
Type "help" >>>  
== RESTART ==  
Mustang  
>>>
```

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
x = thisdict.get("model")  
print(x)
```

نتيجة تشغيل الكود

تغيير قيمة العناصر في القاموس

➤ Change Values

You can change the value of a specific item by referring to its key name.

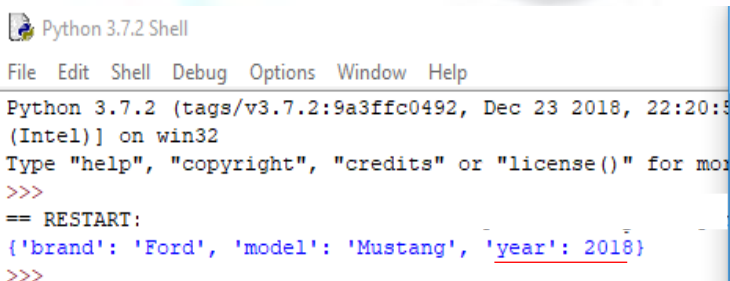
يمكنك تغيير قيمة عنصر محدد في القاموس باستخدام المفتاح الخاص بهذا العنصر

Example

Change the "year" to 2018.

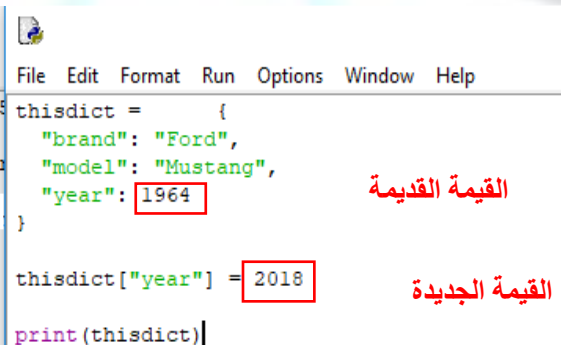
```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
thisdict["year"] = 2018
```

هنا قمنا بتبديل/تغيير قيمة العنصر الذي يملك المفتاح year



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:58) on win32  
(Intel)] Type "help", "copyright", "credits" or "license()" for more  
>>>  
== RESTART:  
{'brand': 'Ford', 'model': 'Mustang', 'year': 2018}  
>>>
```

نتيجة تشغيل الكود



```
File Edit Format Run Options Window Help  
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
thisdict["year"] = 2018  
print(thisdict)
```

القيمة القديمة

القيمة الجديدة

➤ Loop Through a Dictionary

You can loop through a **dictionary** by using a **for** loop.

When looping through a **dictionary**, the return value are the keys of the **dictionary**, but there are methods to return the values as well.

باستخدام الحلقة **for** سنقوم بعرض جميع المفاتيح في القاموس.
قيمة الإرجاع ستكون هي المفاتيح، لكن يوجد هناك طرق أيضا لعرض القيم.

Example

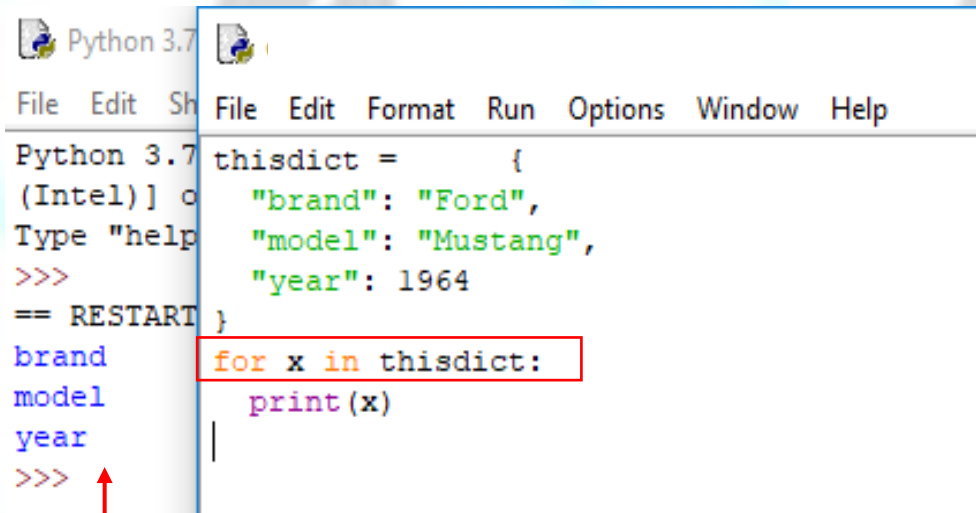
Print all key names in the **dictionary**, one by one.

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
for x in thisdict:
    print(x)
```

هنا قمنا باستخدام الحلقة **for** لعرض وطباعة جميع المفاتيح الموجودة في القاموس **thisdict**

→ طباعة جميع المفاتيح

↑ نتيجة تشغيل الكود



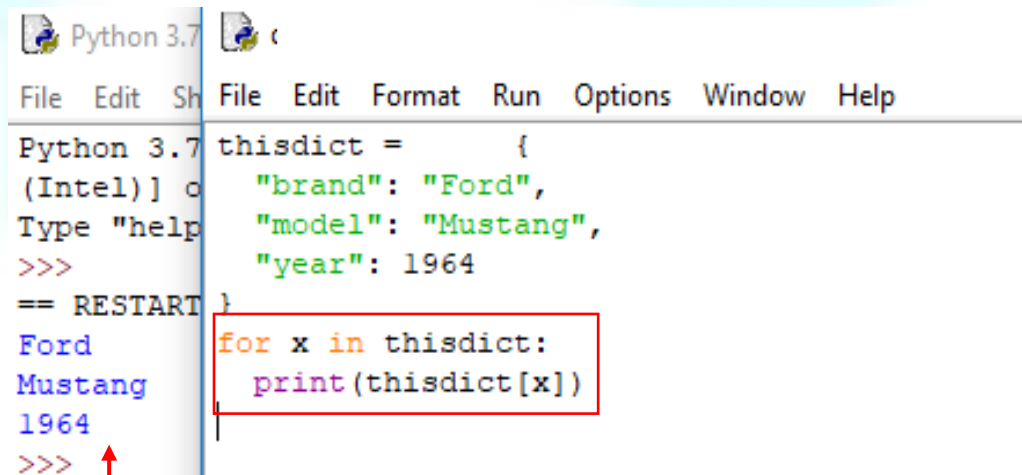
```
Python 3.7
File Edit Shell
Python 3.7 (Intel) o
Type "help
>>>
== RESTART ==>
brand
model
year
>>>
```

Example

Print all values in the **dictionary**, one by one

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
for x in thisdict:
    print(thisdict[x])
```

أما لعرض جميع القيم في القاموس نستخدم هذه الطريقة في الطباعة
وباستخدام الحلقة **for**



```
Python 3.7 Shell
File Edit Shell File Edit Format Run Options Window Help
Python 3.7 thisdict = {
(Intel)] o     "brand": "Ford",
Type "help"    "model": "Mustang",
>>>          "year": 1964
== RESTART == }
Ford
Mustang
1964
>>>
```

طباعة جميع القيم

نتيجة تشغيل الكود

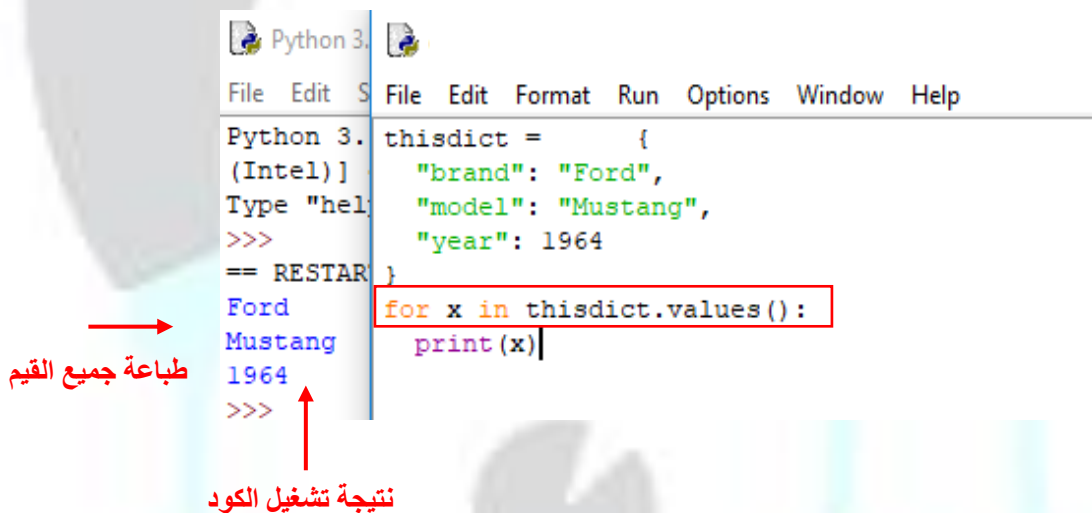
أيضا يمكنك عرض القيم في القاموس بواسطة الدالة **values()**

You can also use the **values()** function to return values of a **dictionary**.

تقوم الدالة **values()** بترتيب مصفوفة فيها جميع القيم الموجودة في القاموس.

Example

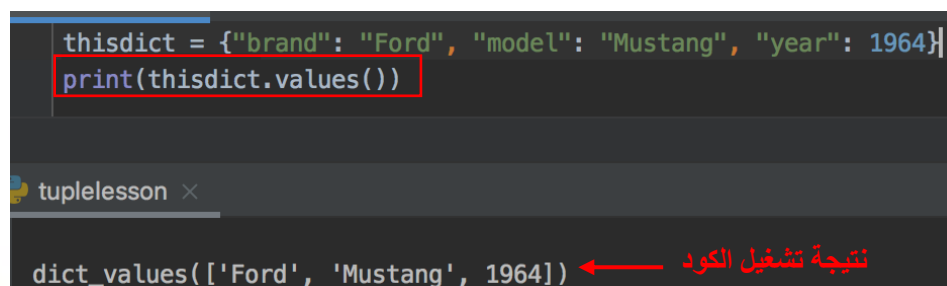
```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
for x in thisdict.values():
    print(x)
```



يمكنك استخدام الدالة **values()** لعرض جميع القيم الموجودة في القاموس بدون استخدام الحلقة **for**

Example

```
thisdict = {"brand": "Ford", "model": "Mustang", "year": 1964}
print(thisdict.values())
```

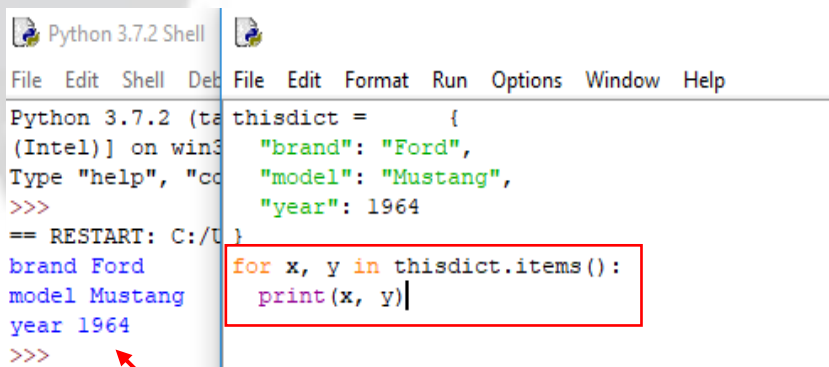


تقوم الدالة **items()** بترجيع مصفوفة فيها جميع العناصر (المفاتيح والقيم) الموجودة في القاموس.

Example

Loop through both keys and values, by using the **items()** function.

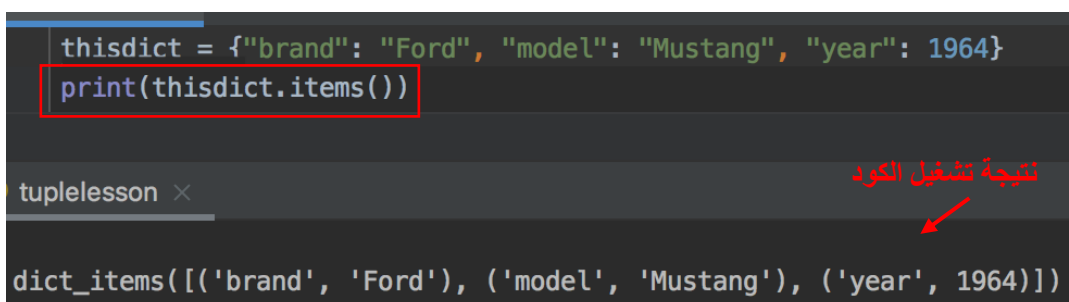
```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
for x, y in thisdict.items():
    print(x, y)
```



نتيجة تشغيل الكود

يمكنك استخدام الدالة **items()** لعرض جميع العناصر الموجودة بدون استخدام الحلقة **for**

```
thisdict = {"brand": "Ford", "model": "Mustang", "year": 1964}
print(thisdict.items())
```



نتيجة تشغيل الكود

مبرمج الغد!
أتممت درسك

طبّق ما تعلمته في هذا الدرس
ولا تنسى مشاركتنا أكوادك

اليوم الثالث والعشرون

القواميس (المعاجم) في لغة البايثون

Python Dictionaries 2

➤ Check if Key Exists

التحقق من وجود مفتاح معين في القاموس

To determine if a specified key is present in a **dictionary** use the **in** keyword.

يُستخدم العامل **in** للبحث عن مفتاح محدد في القاموس، و للمرور على قيم المفاتيح عند استخدامه مع الحلقة **for** كما رأينا في امثلة الدرس السابق.

Example

Check if "model" is present in the **dictionary**.

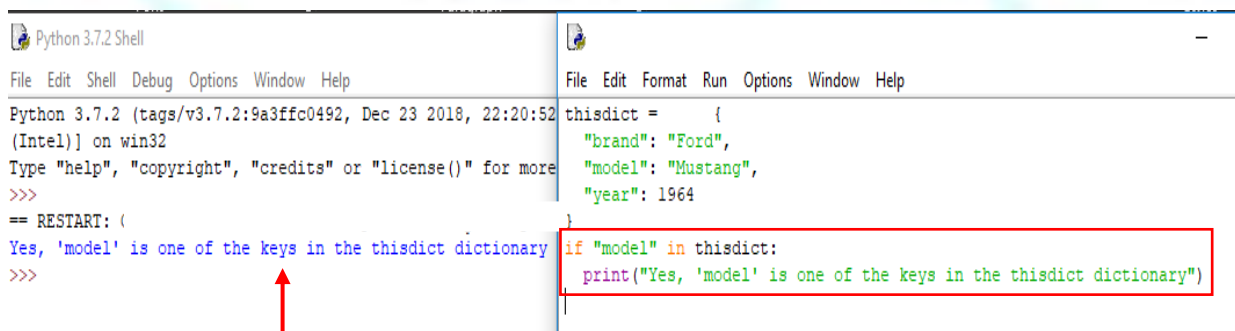
```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}

if "model" in thisdict:
    print("Yes, 'model' is one of the keys in the thisdict dictionary")
```

في هذا المثال سنتحقق بواسطة العامل **in**

ما إذا كان هناك عنصر يملك المفتاح **model** في القاموس **thisdict**

ثم سنقوم بطباعة جملة تفيد ذلك إذا كان موجوداً.



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52)
(Intel) on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
== RESTART: (
Yes, 'model' is one of the keys in the thisdict dictionary
>>>
```

نتيجة تشغيل الكود

➤ Dictionary Length

عدد عناصر القاموس

To determine how many items (key-value pairs) a **dictionary** has, use the **len()** method.

استخدم الدالة **len()** إذا أردت معرفة عدد عناصر القاموس.

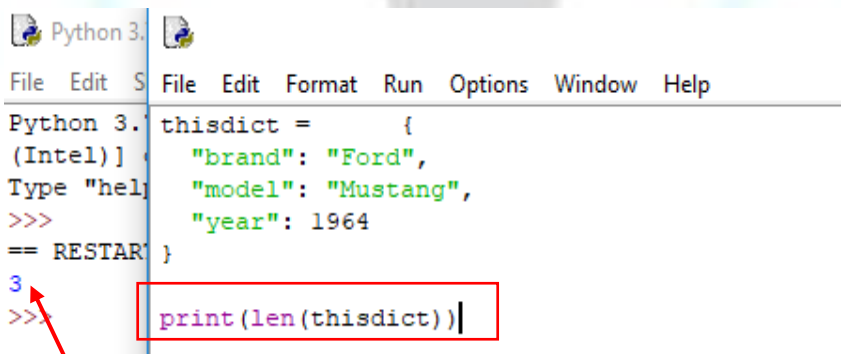
Example

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

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

في هذا المثال قمنا بطباعة عدد عناصر القاموس **thisdict** بواسطة الدالة **len()**

```
print(len(thisdict))
```



```
Python 3.6  
File Edit Shell View Help  
Python 3.6 thisdict = {  
(Intel)] "brand": "Ford",  
Type "help" "model": "Mustang",  
>>> "year": 1964  
== RESTART ==>  
3  
>>> print(len(thisdict))
```

نتيجة تشغيل الكود

إضافة عناصر إلى القاموس

➤ Adding Items

Adding an item to the **dictionary** is done by using a new index key and assigning a value to it.

لإضافة عنصر/عناصر جديدة على القاموس، أولاً نضيف مفتاح جديد ثم نضع القيمة التي يساويها هذا المفتاح.

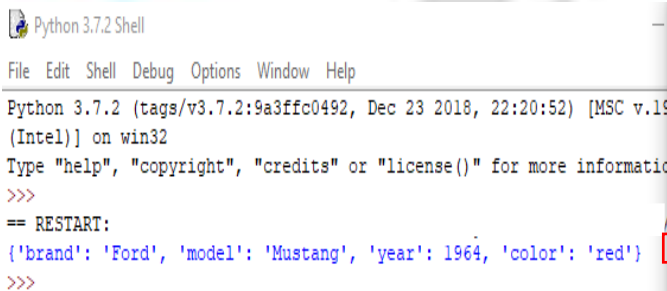
Example

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}
```

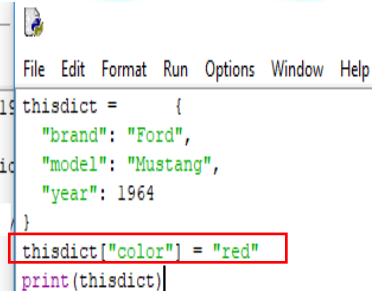
هنا في هذا المثال

أضفنا المفتاح الجديد **color** وأسندنا إليه قيمته وهي **red**

```
thisdict["color"] = "red" ←  
print(thisdict)
```



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 64-bit  
(Intel)] on win32  
Type "help", "copyright", "credits" or "license()" for more  
>>>  
== RESTART:  
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'red'}  
>>>
```



```
File Edit Format Run Options Window Help  
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
thisdict["color"] = "red"  
print(thisdict)
```

نتيجة تشغيل الكود

➤ Removing Items

حذف العناصر في القاموس

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

هناك عدة طرق للحذف في القواميس.

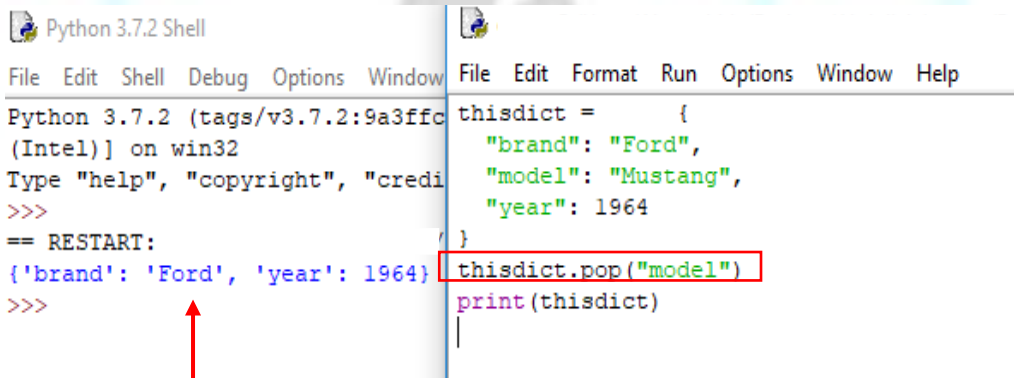
Example

The **pop()** method removes the item with the specified key name.

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
```

تقوم الدالة **pop()** بحذف العنصر عن طريق المفتاح الذي يُمرر لها

```
thisdict.pop("model") ←
print(thisdict)
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc) on win32
Type "help", "copyright", "credits()" or "quit()" for more
>>>
== RESTART:
{'brand': 'Ford', 'year': 1964}
>>>
```

نتيجة تشغيل الكود

ملاحظة:

في حال لم يتم العثور على المفتاح للعنصر المراد حذفه، أو لم تُمرر قيمة للباراميتر () في الدالة فإن البرنامج سيظهر لك خطأ أثناء التشغيل **KeyError**

تقوم الدالة **popitem()** بحذف آخر عنصر تمت إضافته للقاموس.

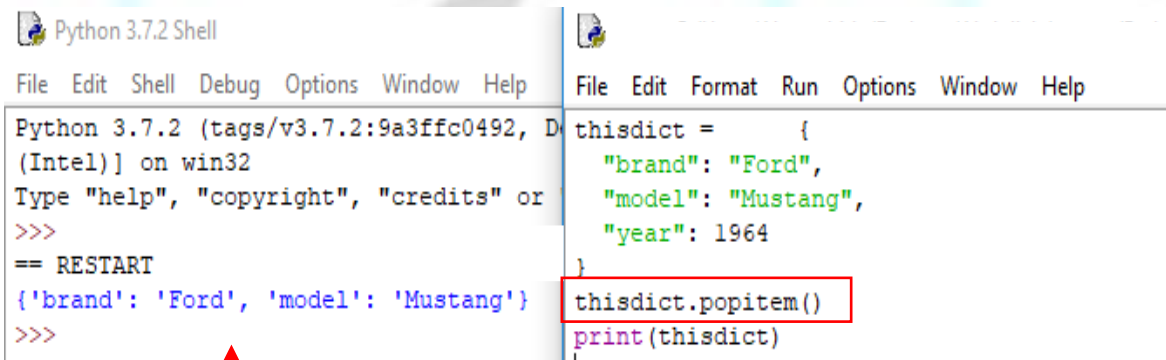
في الإصدارات الأقدم من ٣,٧ سيتم حذف عنصر عشوائي عند استخدام هذه الدالة.

The **popitem()** method removes the last inserted item.

(in versions before 3.7, a random item is removed instead).

Example

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
thisdict.popitem() ←  
print(thisdict)
```



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, D thisdict = {  
(Intel)] on win32 "brand": "Ford",  
Type "help", "copyright", "credits" or "model": "Mustang",  
>>> "year": 1964  
== RESTART {  
{'brand': 'Ford', 'model': 'Mustang'} thisdict.popitem()  
>>> print(thisdict)
```

نتيجة تشغيل الكود

في حال كان القاموس فارغاً، سيُظهر لك البرنامج خطأ عند التشغيل **KeyError**

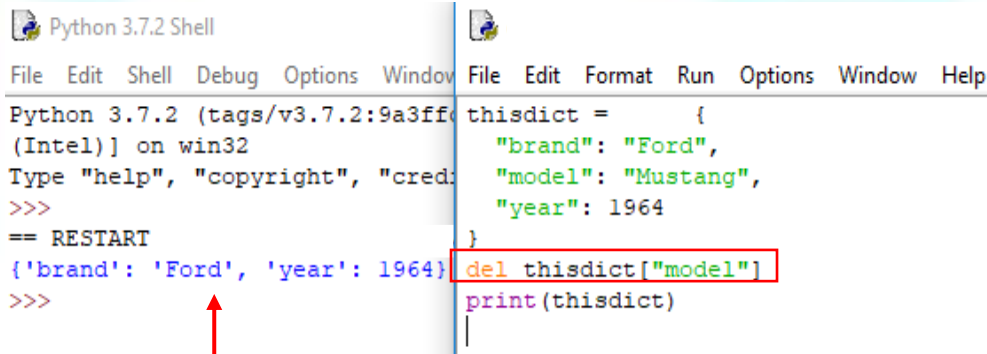
جملة **del** ستقوم بحذف القاموس كاملاً أو عناصر محددة منه.

Example

The **del** keyword removes the item with the specified key name.

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
del thisdict["model"]  
print(thisdict)
```

هنا قمنا بحذف عنصر محدد بواسطة **del** عن طريق مفتاح هذا العنصر



```
Python 3.7.2 Shell  
File Edit Shell Debug Options Window File Edit Format Run Options Window Help  
Python 3.7.2 (tags/v3.7.2:9a3ffe, Dec 12 2019, [AMD64]) on win32  
Type "help", "copyright", "credits()" or "quit()" for more  
>>>  
== RESTART ==>>>  
{'brand': 'Ford', 'year': 1964}  
>>> del thisdict["model"]  
>>> print(thisdict)  
{'brand': 'Ford', 'year': 1964}
```

نتيجة تشغيل الكود

Example

هنا تم حذف القاموس كاملاً من الذاكرة بواسطة **del**

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

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
del thisdict ←
print(thisdict) #this will cause an error because "thisdict" no longer exists.
```

بالتالي سيظهر لك خطأ **error** عند التشغيل **#this will cause an error because "thisdict" no longer exists.**

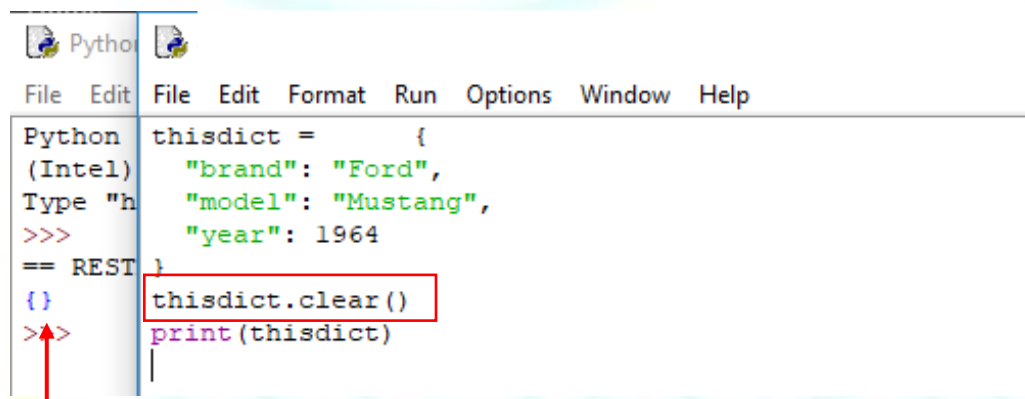
NameError: name 'thisdict' is not defined ← نتيجة تشغيل الكود

Example

The **clear()** keyword empties the **dictionary**.

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
thisdict.clear() ←
print(thisdict)
```

الدالة **clear()** تقوم بحذف جميع عناصر القاموس ويصبح القاموس فارغاً.



نتيجة تشغيل الكود

أتممت درسك بنجاح!
واصل التعلم

طبّق ما تعلمته في هذا الدرس
ولا تنسى مشاركتنا أكوادك

اليوم الرابع والعشرون

القواميس (المعاجم) في لغة البايثون **Python** Dictionaries 3

نسخ القاموس

➤ Copy a Dictionary

You cannot copy a **dictionary** simply by typing **dict2 = dict1**, because: **dict2** will only be a *reference* to **dict1**, and changes made in **dict1** will automatically also be made in **dict2**.

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

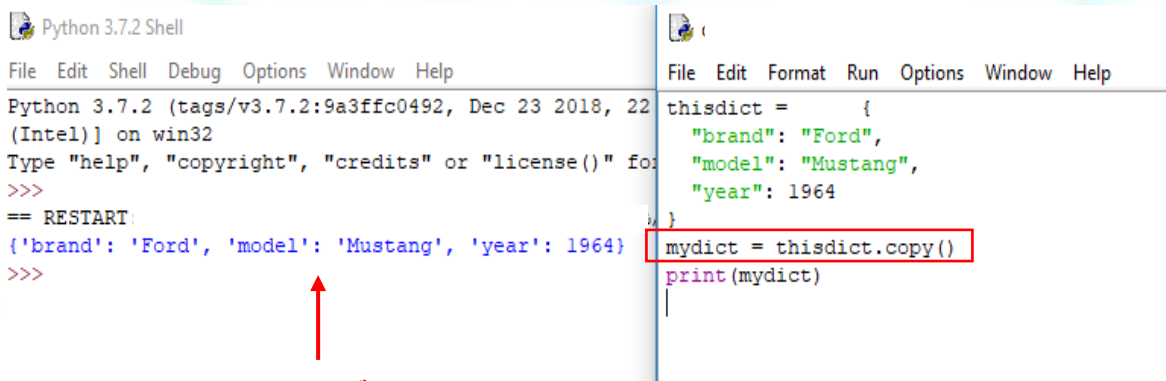
لا يمكنك نسخ القاموس بطريقة الإسناد **dict2 = dict1** فهناك طرق لنسخ القاموس منها استخدام الدالة الجاهزة **copy()**

Example

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

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
mydict = thisdict.copy()  
print(mydict)
```

تقوم الدالة **copy()** بترجيع نسخة من القاموس الذي استدعاه



The screenshot shows two windows. The left window is a 'Python 3.7.2 Shell' with a menu bar (File, Edit, Shell, Debug, Options, Window, Help). It displays the output of the code: '== RESTART: {'brand': 'Ford', 'model': 'Mustang', 'year': 1964} >>>'. The right window is a code editor with a menu bar (File, Edit, Format, Run, Options, Window, Help) showing the code:

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
mydict = thisdict.copy()  
print(mydict)
```

 A red box highlights the line **mydict = thisdict.copy()** in the code editor. A red arrow points from this line down to the shell output.

نتيجة تشغيل الكود

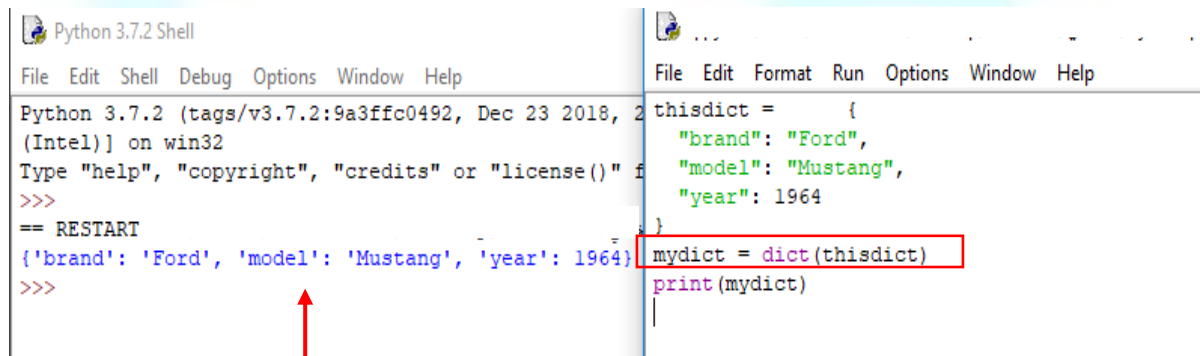
طريقة أخرى لنسخ القاموس وذلك باستخدام الدالة الجاهزة **dict()**

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

Example

Make a copy of a **dictionary** with the **dict()** method.

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
mydict = dict(thisdict) ←  
print(mydict)
```



نتيجة تشغيل الكود

➤ Nested Dictionaries

القواميس المتداخلة

A **dictionary** can also contain many dictionaries, this is called **nested dictionaries**.

القاموس يمكن أن يحتوي على عدة قواميس.

Example

Create a **dictionary** that contain three dictionaries.

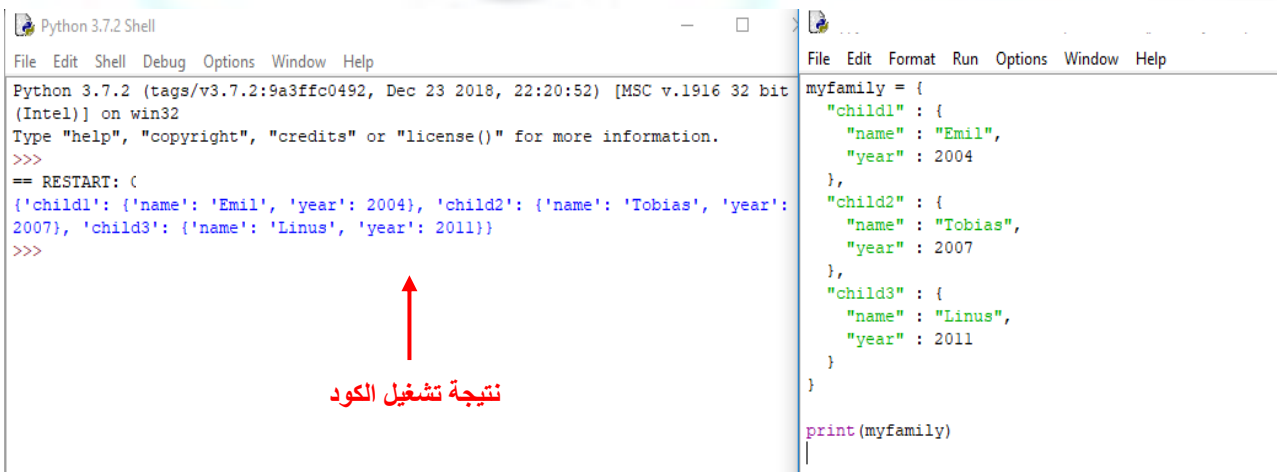
```
myfamily = {
    "child1" : {
        "name" : "Emil",
        "year" : 2004
    },
    "child2" : {
        "name" : "Tobias",
        "year" : 2007
    },
    "child3" : {
        "name" : "Linus",
        "year" : 2011
    }
}
```

```
print(myfamily)
```

هنا في هذا المثال

قمنا بإنشاء القاموس myfamily الذي يحتوي على ٣ قواميس.

“child1” “child2” “child3”



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: (
{'child1': {'name': 'Emil', 'year': 2004}, 'child2': {'name': 'Tobias', 'year':
2007}, 'child3': {'name': 'Linus', 'year': 2011}}
>>>
```

نتيجة تشغيل الكود

```
myfamily = {
    "child1" : {
        "name" : "Emil",
        "year" : 2004
    },
    "child2" : {
        "name" : "Tobias",
        "year" : 2007
    },
    "child3" : {
        "name" : "Linus",
        "year" : 2011
    }
}

print(myfamily)
```

Or, if you want to nest three **dictionaries** that already exists as **dictionaries**.

Example

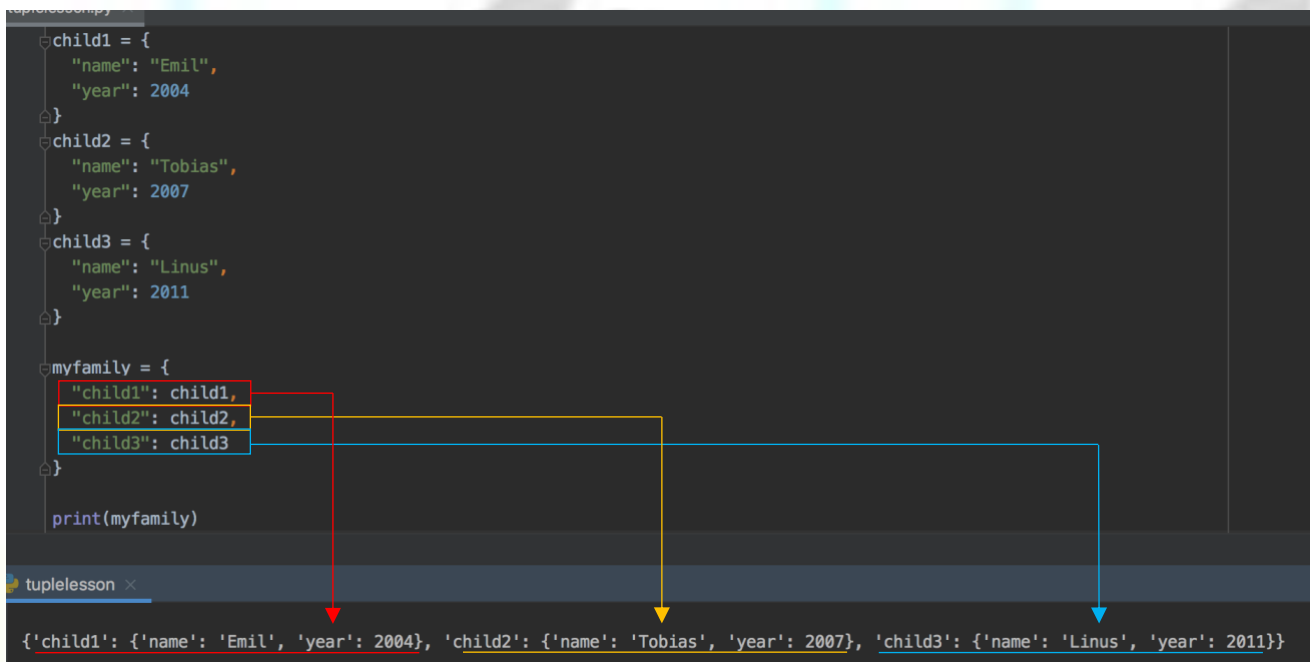
Create three **dictionaries**, then create one **dictionary** that will contain the other three **dictionaries**.

```
child1 = {  
    "name" : "Emil",  
    "year" : 2004  
}  
child2 = {  
    "name" : "Tobias",  
    "year" : 2007  
}  
child3 = {  
    "name" : "Linus",  
    "year" : 2011  
}  
  
myfamily = {  
    "child1" : child1,  
    "child2" : child2,  
    "child3" : child3  
}  
  
print(myfamily)
```

هنا في هذا المثال

قمنا بإنشاء ثلاثة قواميس child1 – child2 – child3
ثم أنشأنا قاموس واحد myfamily يضم بداخله الثلاثة قواميس

نتيجة تشغيل الكود



```
{'child1': {'name': 'Emil', 'year': 2004}, 'child2': {'name': 'Tobias', 'year': 2007}, 'child3': {'name': 'Linus', 'year': 2011}}
```


➤ The `dict()` Constructor

بناء/تكوين القاموس باستخدام الدالة `dict()`

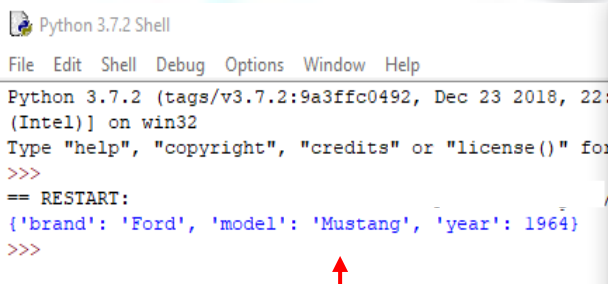
It is also possible to use the `dict()` constructor to make a new dictionary.

Example

```
thisdict = dict(brand="Ford", model="Mustang", year=1964)
# note that keywords are not string literals
# note the use of equals rather than colon for the assignment
print(thisdict)
```

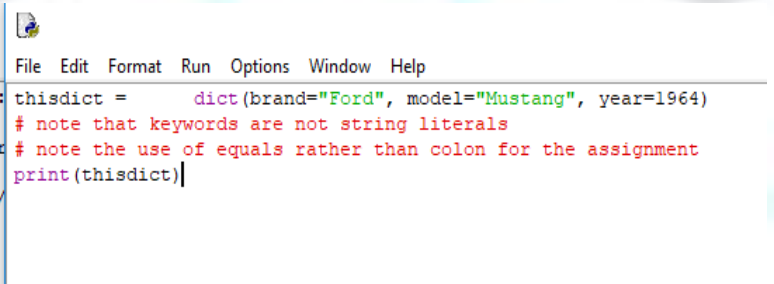
لاحظ بداخل الدالة `dict()`

- المفاتيح هنا لا تكون سلاسل نصية
- قمنا باستبدال الرمز : الذي يكون بين المفتاح والقيمة بالرمز =



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for
>>>
== RESTART:
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
>>>
```

نتيجة تشغيل الكود



```
File Edit Format Run Options Window Help
thisdict = dict(brand="Ford", model="Mustang", year=1964)
# note that keywords are not string literals
# note the use of equals rather than colon for the assignment
print(thisdict)
```

➤ Dictionary Methods

دوال القاموس في لغة بايثون

بايثون تحتوي على دوال جاهزة يمكنك استخدامها للتعامل مع القواميس وهي ما ناقشنا أغلبها في هذه الدروس.

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

Method	Description
<code>clear()</code>	Removes all the elements from the dictionary
<code>copy()</code>	Returns a copy of the dictionary
<code>fromkeys()</code>	Returns a dictionary with the specified keys and values
<code>get()</code>	Returns the value of the specified key
<code>items()</code>	Returns a list containing the a tuple for each key value pair
<code>keys()</code>	Returns a list containing the dictionary's keys
<code>pop()</code>	Removes the element with the specified key
<code>popitem()</code>	Removes the last inserted key-value pair
<code>setdefault()</code>	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
<code>update()</code>	Updates the dictionary with the specified key-value pairs
<code>values()</code>	Returns a list of all the values in the dictionary

رائع!
أتممت درسك الأخير لهذا الأسبوع

روابط قد تساعدك

Check the links below

- [في بايثون dict القواميس](#)
- [Python Programming #6 - Dictionaries](#)
- [\[Dictionary\] دورة لغة بايثون 7 : تعلم البرمجة بطريقة سهلة القواميس](#)
- [09 - Python - Beginners Tutorial - Dictionaries](#)
- [متغير قاموس Python in Arabic شرح بايثون بالعربي 16 - 16 Learn Python in Arabic #16](#)
- [Dictionary Python](#)
- [special loop on dictionary Pyth تكرر احترافي للمتغير القاموس - 31 Learn Python in Arabic #31](#)
- [Python Dictionary , Nested Dictionary, Update/modify dictionary & It's method - Python programming](#)
- [Python nested dictionaries](#)

طبق ما تعلمته في هذا الدرس
ولا تنسى مشاركتنا أكوادك

اليوم الخامس والعشرون & اليوم السادس والعشرون

تحدي الأسبوع (يتم حله ورفعته على Github)

أولاً:

قم بإضافة الأعداد التالية 4,8,12 إلى `set= {1,3,5,7,8}` ثم احذف العنصر 8 وأخيراً قم بإفراغ المجموعة تمامًا.

ثانياً:

أنشئ قاموس وأضف إليه البيانات التالية:

name	pigeon
type	bird
skin cover	feathers

ثم قم بالآتي:

- 1 - اكتب كود يطبع قيمة **type**
- 2 - قم بتغيير قيمة **skin cover**

انتظرونا في دروس الأسبوع القادم