

Set

المجموعة Set

المجموعة Set عبارة عن سلسلة من العناصر الفريدة . تسمى العناصر داخل المجموعة : أعضاء . و لا يمكن ان يتكرر نفس العضو داخل المجموعة . للمجموعات فائدة كبيرة و ذلك بسبب العمليات التي يمكن تنفيذها على المجموعة ، مثل الإتحاد ، التقاطع ، الاختلاف ، الخ .

مميزات المجموعة :

- تقبل اي نوع من البيانات ك عضو.
- غير مرتبة .
- تحتوي فقط على أعضاء فريدة القيمة .
- قابلة للتعديل .

لإنشاء مجموعة فارغة

```
my_empty_set = set()
```

لإنشاء مجموعة مع أعضاء مبدئية

```
my_set = {"kiwi", "apple"}
```

لإضافة عضو جديد للمجموعة ، نستخدم الدالة add

```
my_set.add("orange")
```

لحذف عضو من المجموعة ، نستخدم الدالة remove
لاحظ ان هذه الدالة تحذف العضو اذا كان موجود أو سترفع خطأ .

```
my_set.remove("kiwi")
```

لحذف عضو من المجموعة باستخدام `discard`
لاحظ سيتم حذف العضو ان وجد ، و لن يتم رفع خطأ في حال عدم وجوده.

```
my_set.discard("orange")
```

لإختبار وجود عضو داخل المجموعة نستخدم الكلمة المفتاحية `in`
حيث سترجع لنا قيمة منطقية إما بالإيجاب أو النفي `True` أو `False`

```
if 'apple' in my_set:  
    print("found!!")
```

لتكرار عملية على جميع أعضاء المجموعة نستخدم `for...in`

```
for member in my_set:  
    print(member)
```

عمليات المجموعة

كما ذكرنا آنفاً ، ما يجعل المجموعات مميزة هي العمليات التي يمكننا تنفيذها على المجموعة . و التي سنطلع عليها هنا ، و كبدائية سننشئ مجموعتين : مجموعة للمدن التي زارها فايز ، و مجموعة للمدن التي زارها أحمد و سنختبر عليها العمليات .

```
faiz_visited_cities = {"london", "Finland", "Jeddah"}  
ahmed_visited_cities = {"Finland", "New York", "Munich", "Damm  
am"}
```

الإتحاد بين مجموعتين Union

بحيث يتم الجمع بين جميع اعضاء المجموعتين في مجموعة واحدة مع مراعاة عدم تكرار الأعضاء .

```
print(faiz_visited_cities.union(ahmed_visited_cities))  
# using an operator  
print(faiz_visited_cities | ahmed_visited_cities)
```

التقاطع بين مجموعتين Intersection

بحيث يتم فقط انشاء مجموعة تحتوي على العناصر المشتركة (المتقاطعة) بين كلا المجموعتين . مع مراعاة عدم التكرار.

```
print(faiz_visited_cities.intersection(ahmed_visited_cities))  
# using an operator  
print(faiz_visited_cities & ahmed_visited_cities)
```

الاختلاف بين مجموعتين Difference

يتم انشاء مجموعة جديدة تحتوي على العناصر الموجودة في المجموعة الأولى و الغير موجودة في المجموعة الثانية .

```
print(faiz_visited_cities.difference(ahmed_visited_cities))  
# using an operator  
print(ahmed_visited_cities - faiz_visited_cities)
```

الاختلاف المتماثل بين مجموعتين Symmetric Difference

يتم انشاء مجموعة جديدة تحتوي على العناصر المختلفة في كل من المجموعتين . اي العناصر الموجودة في المجموعة الأولى و الغير موجودة في المجموعة الثانية و العكس .

```
print(faiz_visited_cities.symmetric_difference(ahmed_visited_cities))  
# using an operator  
print(faiz_visited_cities ^ ahmed_visited_cities)
```

Dictionary

السجل Dictionary

ما هو السجل Dictionary

السجل عبارة عن مجموعة من العناصر المحفوظة في مكان واحد. لكل عنصر منها : مفتاح و قيمة .
فعلى سبيل المثال ، كنا ندخل على قيمة عناصر القائمة `list` من خلال موقعها بداية من الصفر ، لكن بالنسبة للسجل ، فنحن ندخل على قيمة العناصر من خلال المفتاح .

بالعادة المفتاح يكون من نوع نصي `str` و لكن يمكنك استخدام اي نوع من انواع البيانات الغير قابلة للتعديل . يجب أن يكون المفتاح ذو قيمة فريدة ، فعندما تحاول ان تطيف عنصرين بنفس المفتاح ، فسيتم الكتابة على قيمة العنصر السابق بقيمة العنصر الجديد.

مزايا السجل Dictionary

- قابل للتعديل
- العناصر غير مرتبة (مضمونة الترتيب حسب الإضافة من نسخة 3.7)
- المفتاح يجب ان يكون غير قابل للتعديل ، و فريد .
- القيمة يمكن ان تكون اي شيء.

إنشاء سجل جديد فارغ

```
empty_dictionary = {}
```

إنشاء سجل مع عناصر مبدئية

```
my_dictionary = {"key1" : 124, "key2": 400}
```

الدخول على (استرجاع قيمة) عنصر في السجل Dictionary

يتم استرجاع قيمة عنصر داخل السجل باستخدام المفتاح

```
print(my_dictionary["key2"])
```

الدخول على (استرجاع قيمة) عنصر في السجل Dictionary باستخدام الدالة get
يمكننا ايضا استخدام الدالة الخاصة بالسجل `get` من اجل استرجاع قيمة عنصر عبر المفتاح ، أو توفير قيمة افتراضية في حالة عدم وجود هذا العنصر في السجل .

```
my_dictionary.get("key", "nothing")
```

إضافة عنصر جديد للسجل
لإضافة عنصر جديد في السجل ، بإمكاننا ان نضيف المفتاح الجديد و قيمته للسجل .

```
my_dictionary["key3"] = "another value"
```

إضافة عنصر جديد أو عدة عناصر للسجل باستخدام الدالة update

```
my_dictionary.update(key4 = "value 4", key5 = "value 5")
```

بالإمكان ايضا اضافة العناصر باستخدام الدالة `update` و لكن على شكل قائمة متعددة الأبعاد ، بحيث داخل القائمة ، قائمة أخرى تتكون من عنصرين ، سيتم اعتبار العنصر الأول كمفتاح و العنصر الثاني كقيمة .

```
my_dictionary.update([["key6", "value6"], ["key7", "value7"]])
```

لحذف عنصر من السجل

```
del my_dictionary["key4"]
```

لحذف عنصر من السجل و استرجاع قيمته في نفس الوقت نستخدم الدالة `pop` ، و التي تأخذ المفتاح ك مُعطى .

```
dict_elem = my_dictionary.pop("key5")  
print(dict_elem)
```

لحذف عنصر عشوائي (مع النسخ الجديدة من البايتون ، اخر عنصر) من السجل و استرجاع العنصر ك توبل tuple مكون من المفتاح و القيمة

```
randomElment = my_dictionary.popitem()
```

لإختبار وجود عنصر في السجل نستخدم الكلمة المفتاحية `in` ، بحيث سنسترجع قيمة منطقية اما بالإيجاب او النفي على هيئة `True` او `False`

```
if "key4" in my_dictionary:
    print("key4 is found!")
```

لتنفيذ عملية على جميع عناصر السجل (تكرار) نستخدم `for ... in`

```
for key in my_dictionary:
    print(my_dictionary[key])
```

استرجاع سلسلة من مفاتيح السجل باستخدام الدالة `keys` بالإمكان استرجاع سلسلة مكونة من جميع المفاتيح (بدون القيم) من السجل باستخدام الدالة `keys` . و التي بدورها تعيد لنا سلسلة من المفاتيح القابلة للدوران `dict_keys` ، و بالإمكان تحويلها الى قائمة من خلال الدالة `list` في حال قضت الحاجة لذلك .

```
my_dictionary_keys = my_dictionary.keys()
# convert to list
my_dictionary_keys_list = list(my_dictionary.keys())
```

استرجاع سلسلة من قيم السجل باستخدام الدالة `values` بالإمكان استرجاع سلسلة مكونة من جميع القيم (بدون مفاتيحها) من السجل باستخدام الدالة `values` . و التي بدورها تعيد لنا سلسلة من القيم القابلة للدوران `dict_values` ، و بالإمكان تحويلها الى قائمة من خلال الدالة `list` في حال قضت الحاجة لذلك .

```
my_dictionary_values = my_dictionary.values()
```

```
# convert to list
my_dictionary_values_list = list(my_dictionary.values())
print(my_dictionary_values_list)
```

استرجاع سلسلة من قيم السجل و المفاتيح القابلة للدوران

```
for key, value in my_dictionary.items():
    print(key, value)
```