

# 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)
```