

# مجموعه‌ها (Sets) در پایتون

مجموعه (Set) یک ساختار داده‌ای بدون ترتیب و بدون مقدار تکراری است که امکان انجام عملیات ریاضی مانند اشتراک، اتحاد و تفاوت را فراهم می‌کند.

## ۱. ایجاد مجموعه

✓ تعریف مجموعه با `{ }`

```
my_set = {1, 2, 3, 4, 5}
print(my_set) # خروجی: {5, 4, 3, 2, 1}
```

📌 نکته:

- مقادیر تکراری به صورت خودکار حذف می‌شوند!

```
my_set = {1, 2, 2, 3, 4, 4}
print(my_set) # خروجی: {4, 3, 2, 1}
```

✓ استفاده از `set()` برای تبدیل لیست به مجموعه

```
my_list = [1, 2, 3, 4, 4, 5, 5, 6]
my_set = set(my_list)
print(my_set) # خروجی: {6, 5, 4, 3, 2, 1}
```

✓ مجموعه خالی (`empty set`)

```
empty_set = set() # ✗ استفاده کنید چون دیکشنری ایجاد می‌کند
```

## ۲. دسترسی به عناصر مجموعه

📌 چون مجموعه‌ها \*ترتیب ندارند\*، نمی‌توان با ایندکس (`[]`) به عناصر آن‌ها دسترسی داشت! اما می‌توان روی آن‌ها حلقه زد:

```
my_set = {10, 20, 30}
for item in my_set:
    print(item)
```

### ۳. عملیات روی مجموعه‌ها

افزودن مقدار جدید → `() add`

```
my_set = {1, 2, 3}
my_set.add(4)
print(my_set) # خروجی: {4, 3, 2, 1}
```

نکته:

- اگر مقدار موجود باشد، تغییری ایجاد نمی‌شود.

حذف مقدار → `() discard` و `() remove`

```
my_set.remove(3) # مقدار 3 را حذف می‌کند
print(my_set) # خروجی: {4, 2, 1}

# my_set.remove(10) # خطا می‌دهد اگر مقدار وجود نداشته باشد ❌
```

`() discard` بدون خطا اجرا می‌شود:

```
my_set.discard(10) # اگر مقدار نباشد، خطا نمی‌دهد
```

حذف و دریافت یک مقدار تصادفی → `() pop`

```
removed_item = my_set.pop()
print(removed_item) # یک مقدار تصادفی را حذف و برمی‌گرداند
```

حذف همه مقادیر مجموعه → `() clear`

```
my_set.clear()
print(my_set) # خروجی: set()
```

### ۴. بررسی عضویت (`in`)

```
my_set = {1, 2, 3}
print(2 in my_set) # خروجی: True
print(5 in my_set) # خروجی: False
```

## ۵. عملیات ریاضی روی مجموعه‌ها

اتحاد (  | ) نیز قابل استفاده است → `union()` 

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}

union_set = set1.union(set2)
print(union_set) # خروجی: {5, 4, 3, 2, 1}

# روش کوتاه‌تر:
print(set1 | set2) # خروجی: {5, 4, 3, 2, 1}
```

اشتراک (  & ) نیز قابل استفاده است → `intersection()` 



```
intersect_set = set1.intersection(set2)
print(intersect_set) # خروجی: {3}

# روش کوتاه‌تر:
print(set1 & set2) # خروجی: {3}
```

تفاوت (  - ) نیز قابل استفاده است → `difference()` 

```
diff_set = set1.difference(set2)
print(diff_set) # خروجی: {2, 1} (مقادیر موجود در set1 ولی نه در set2)


# روش کوتاه‌تر:
print(set1 - set2) # خروجی: {2, 1}
```

تفاوت متقارن (  ^ ) نیز قابل استفاده است → `symmetric_difference()` 

```
sym_diff_set = set1.symmetric_difference(set2)
print(sym_diff_set) # خروجی: {5, 4, 2, 1} (همه مقادیری که در یکی از مجموعه‌ها هستند ولی نه در هر دو)

# روش کوتاه‌تر:
print(set1 ^ set2) # خروجی: {5, 4, 2, 1}
```

## ۶. روابط بین مجموعه‌ها

→ `issubset()` بررسی زیرمجموعه بودن 

```
set1 = {1, 2}
set2 = {1, 2, 3, 4}

print(set1.issubset(set2)) # خروجی: True (وجود دارد set1 در set2 چون)
print(set2.issubset(set1)) # خروجی: False
```

→ `issuperset()` بررسی شامل بودن 

```
print(set2.issuperset(set1)) # خروجی: True
print(set1.issuperset(set2)) # خروجی: False
```

→ `isdisjoint()` بررسی عدم اشتراک 

```
set3 = {10, 20, 30}
print(set1.isdisjoint(set3)) # خروجی: True (چون اشتراکی ندارند)
```

## ۷. تبدیل انواع داده به مجموعه

✓ تبدیل `list` به `set` برای حذف مقادیر تکراری

```
my_list = [1, 2, 2, 3, 4, 4, 5]
unique_values = set(my_list)
print(unique_values) # خروجی: {5, 4, 3, 2, 1}
```

۸. تبدیل `set` به `list`

```
my_set = {10, 20, 30}
my_list = list(my_set)
print(my_list) # خروجی: [30, 20, 10]
```

## ۹. کپی کردن مجموعه

→ `copy()` کپی سطحی 

```
set1 = {1, 2, 3}
set2 = set1.copy()

print(set2) # خروجی: {3, 2, 1}
```

## ۱۰. حذف مقدار تصادفی از مجموعه

🚀 چون مجموعه‌ها \*بدون ترتیب\* هستند، `pop()` مقدار \*تصادفی\* را حذف می‌کند!

```
my_set = {10, 20, 30, 40}
removed_item = my_set.pop()
print(removed_item) # مقدار تصادفی را حذف می‌کند
print(my_set)
```

## جمع‌بندی

- ✅ مجموعه‌ها (Sets) در پایتون بدون ترتیب و بدون مقدار تکراری هستند.
  - ✅ از عملیات ریاضی مانند اتحاد، اشتراک، تفاوت و تفاوت متقارن پشتیبانی می‌کنند.
  - ✅ برای حذف مقادیر تکراری از لیست‌ها بسیار مفید هستند.
  - ✅ امکان انجام عملیات افزودن، حذف، بررسی عضویت و کپی کردن دارند.
  - ✅ چون مجموعه‌ها ایندکس ندارند، نمی‌توان مقدار خاصی را مستقیماً دسترسی داد.
- 🔥 مجموعه‌ها برای داده‌های منحصربه‌فرد و پردازش سریع روی مجموعه‌های بزرگ بسیار کارآمد هستند! 🚀