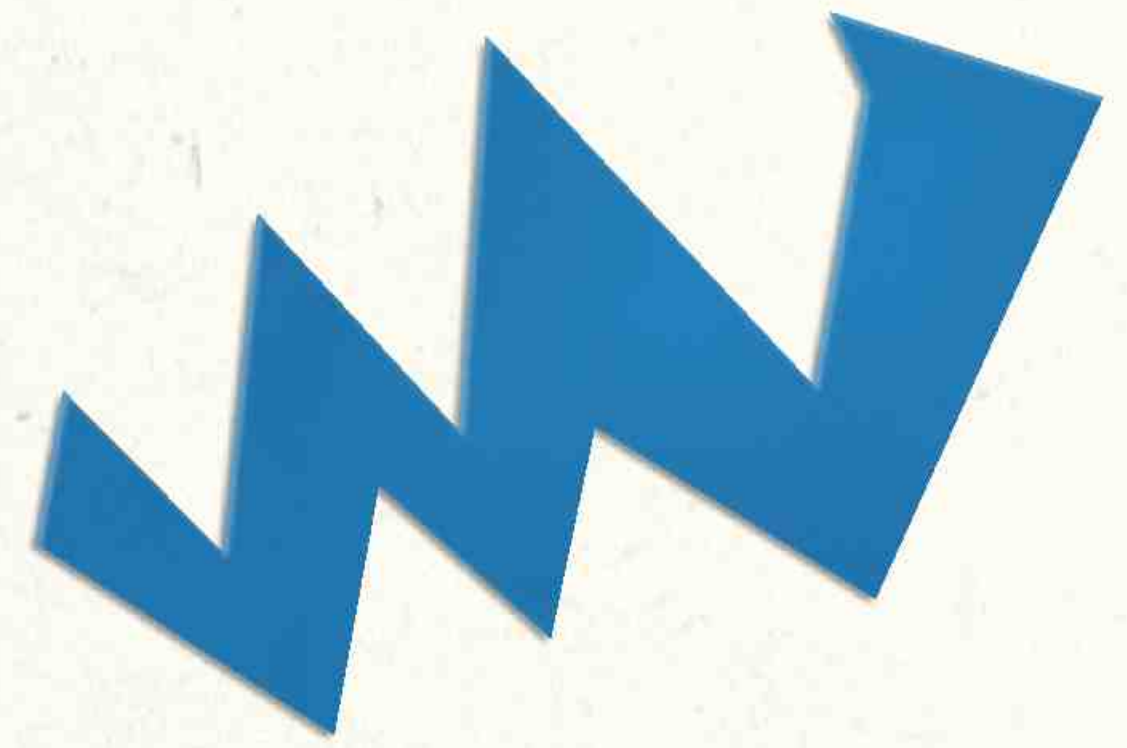
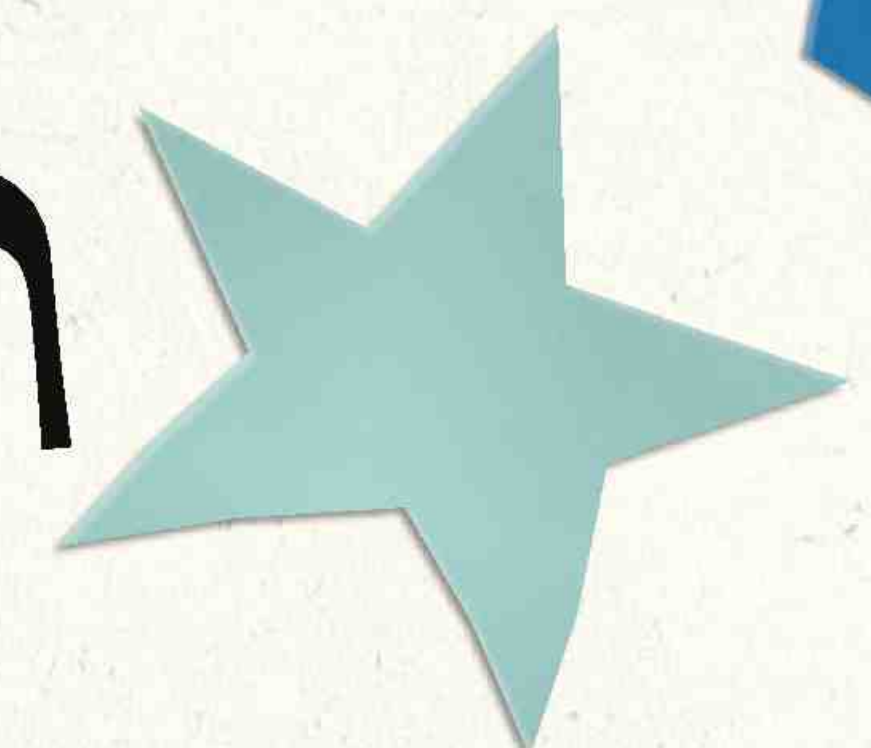
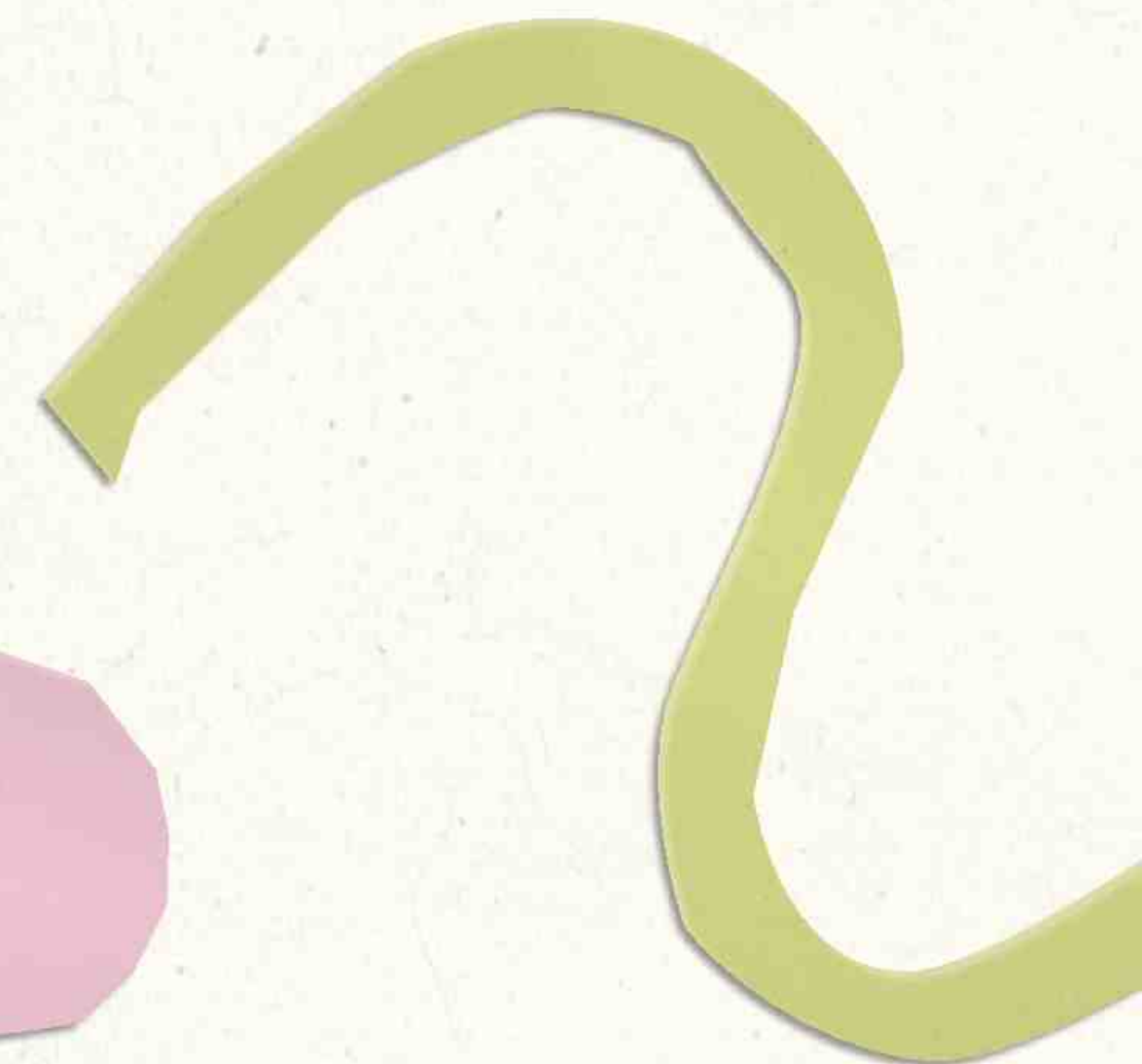
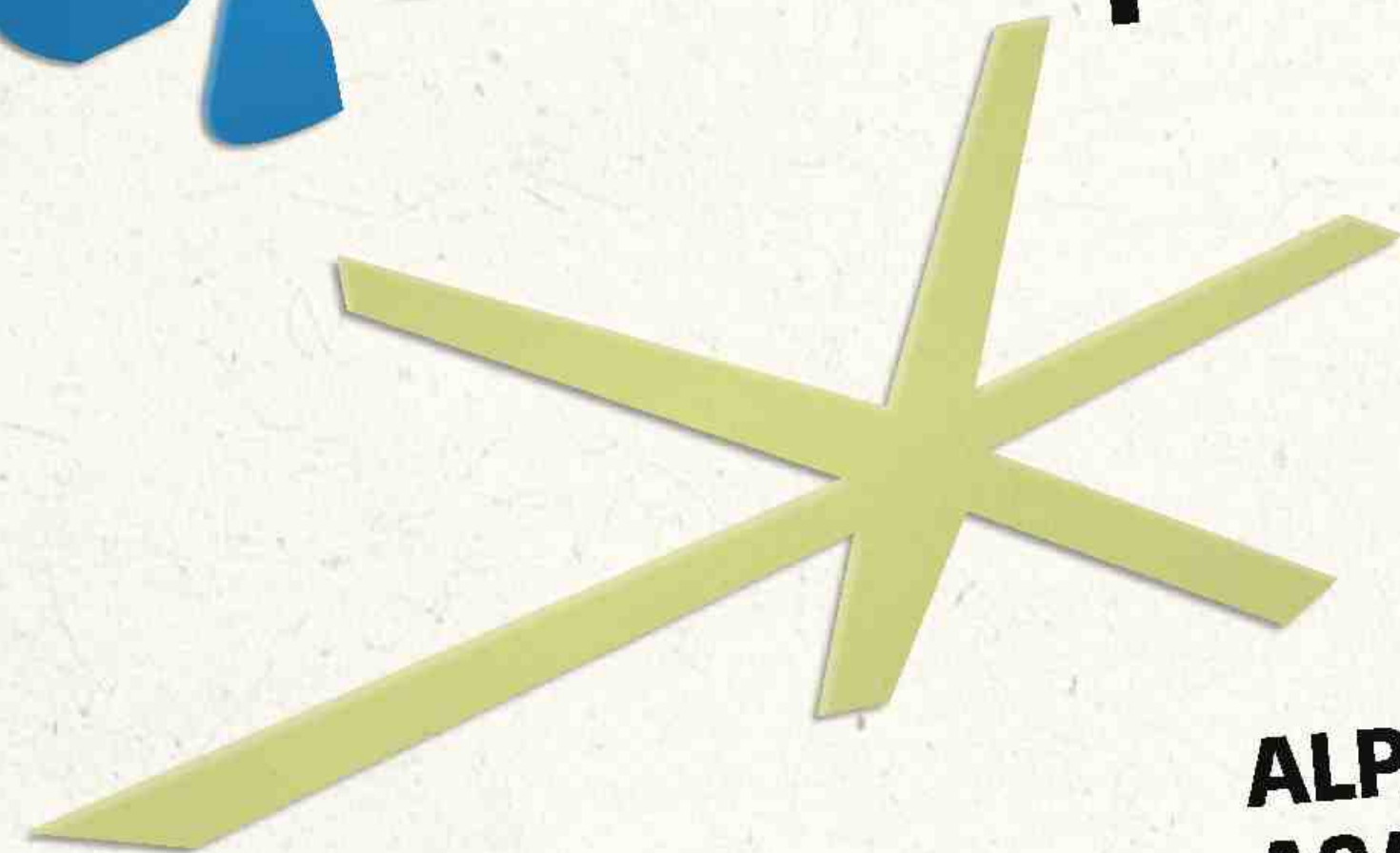
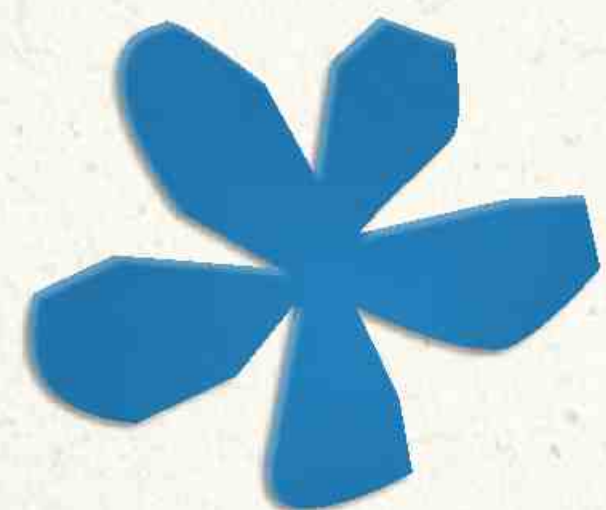




python

class

ALPAN ACADEMY
ASAL RASOULI





مجموعه (Set) چیست؟

مجموعه (Set) یکی از انواع داده‌های مرکب در پایتون است که برای نگهداری مجموعه‌ای از عناصر یکتا و بدون ترتیب مشخص استفاده می‌شود.

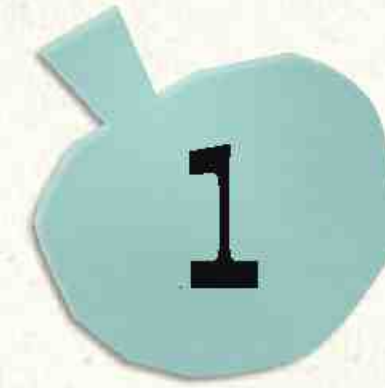
مهم‌ترین ویژگی مجموعه این است که عناصر تکراری را نگه نمی‌دارد.

مجموعه را معمولاً با {} یا تابع set() ایجاد می‌کنیم.

```
numbers = {1, 2, 2, 3, 3, 4}
```

```
print(numbers)
```

ویژگی‌های دیکشنری



قابل تغییر
(Mutable)



بدون تکرار



بدون ترتیب
(Unordered)



عناصر Set باید
Hashable باشند

مثلاً رشته، عدد و تاپل می‌توانند عضو Set باشند:

```
my_set = {"Ali", 20, (10, 20)}
```

```
print(my_set)
```

اما نمی‌توان یک list یا set را مستقیماً داخل Set قرار داد:

```
my_set = {[1, 2], [3, 4]}
```

این کد خطا ایجاد می‌کند، زیرا list قابل Hash شدن نیست.



ساخت Set

روش اول: استفاده از {}

روش دوم: استفاده از set()

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

```
print(numbers)
```

```
fruits = {"apple", "banana", "orange"}
```

```
print(fruits)
```

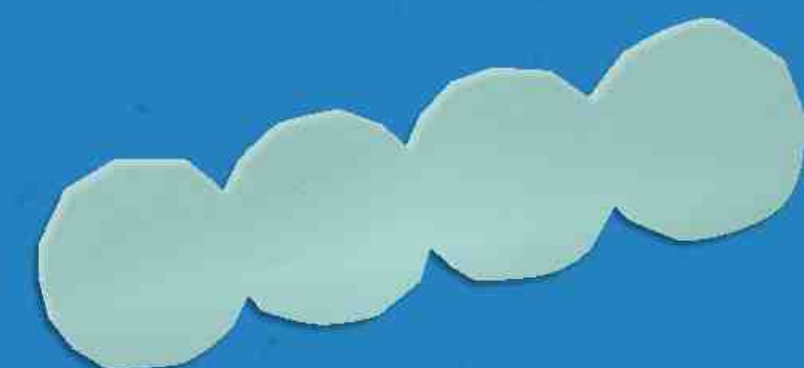
نکته مهم: Set خالی

برای ساخت یک Set خالی نباید فقط از {} استفاده کنیم.

زیرا {} در پایتون یک Dictionary خالی ایجاد می‌کند.

برای ساخت Set خالی باید از set() استفاده کنیم:

```
empty_set = set()  
print(type(empty_set))
```





عملیات پایه‌ای روی Set

اضافه کردن عنصر با add

برای اضافه کردن یک عنصر به Set از متد add استفاده می‌کنیم.

```
fruits = {"سیب", "موز"}
```

```
fruits.add("پرتقال")
```

```
print(fruits)
```

اضافه کردن چند عنصر با update

اگر بخواهیم چند عنصر را همزمان به Set اضافه کنیم، می‌توانیم از update استفاده کنیم.

```
numbers = {1, 2, 3}
```

```
numbers.update([4, 5, 6])
```

```
print(numbers)
```

حذف عنصر با remove

متد remove یک عنصر مشخص را از Set حذف می‌کند.
اگر عنصر موردنظر در Set وجود نداشته باشد، remove خطا ایجاد می‌کند.

```
numbers = {1, 2, 3, 4}
```

```
numbers.remove(3)
```

```
print(numbers)
```

حذف عنصر با discard

متد discard نیز برای حذف یک عنصر استفاده می‌شود، اما تفاوت آن با remove این است که اگر عنصر وجود نداشته باشد، خطا ایجاد نمی‌کند.

```
numbers = {1, 2, 3}
```

```
numbers.discard(5)
```

```
print(numbers)
```

حذف یک عنصر با pop

متد pop یک عنصر از Set را حذف کرده و همان عنصر را برمی گرداند.
در Set نمی توان مشخص کرد کدام عنصر با pop حذف شود؛ چون Set ترتیب مشخصی ندارد.

```
numbers = {10, 20, 30}
```

```
item = numbers.pop()
```

```
print("عنصر حذف شده:", item)
```

```
print("Set:", numbers)
```

حذف تمام عناصر با clear

برای خالی کردن کامل Set از clear استفاده می‌کنیم.

```
numbers = {1, 2, 3, 4}
```

```
numbers.clear()
```

```
print(numbers)
```



بررسی وجود عنصر در Set

وقتی **in** را روی یک Dictionary استفاده می‌کنیم، **کلیدها** بررسی می‌شوند، نه مقدا رها.

not in

```
fruits = {"سیب", "موز", "پرتقال"}
```

```
print("انگور" not in fruits)
```

in

```
fruits = {"سیب", "موز", "پرتقال"}
```

```
print("سیب" in fruits)
```

```
print("انگور" in fruits)
```



عملیات ریاضی روی Set ها

اجتماع (Union)

اجتماع دو مجموعه شامل تمام عناصر موجود در هر دو مجموعه است.
با union یا عملگر | انجام می شود.

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

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

```
print(set1.union(set2))
```

```
print(set1 | set2)
```

اشتراک (Intersection)

اشتراک دو مجموعه شامل عناصری است که در هر دو مجموعه وجود دارند.
با intersection یا & انجام می شود.

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

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

```
print(set1.intersection(set2))
```

```
print(set1 & set2)
```

تفاوت (Difference)

تفاوت set1 و set2 یعنی عناصری که در set1 وجود دارند اما در set2 وجود ندارند.

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

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

```
print(set1.difference(set2))
```

```
print(set1 - set2)
```

تفاوت متقارن (Symmetric Difference)

تفاوت متقارن شامل عناصری است که فقط در یکی از دو مجموعه وجود دارند، نه در هر دو.

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

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

```
print(set1.symmetric_difference(set2))
```

```
print(set1 ^ set2)
```



توابع کاربردی برای Set

توابع کاربردی



len

تعداد عناصر



sum

مجموع



max

بزرگ‌ترین مقدار



min

کوچک‌ترین مقدار

```
numbers = {10, 20, 30, 40}
```

```
print(len(numbers))
```

```
print(sum(numbers))
```

```
print(max(numbers))
```

```
print(min(numbers))
```



تبدیل Set به List یا Tuple

```
numbers = {10, 20, 30}
```

```
numbers_list = list(numbers)  
print(numbers_list)
```

```
numbers_tuple = tuple(numbers)  
print(numbers_tuple)
```

چون Set از Indexing پشتیبانی نمی‌کند،
اگر بخواهیم به عناصر با اندیس دسترسی
داشته باشیم، می‌توانیم آن را به List تبدیل
کنیم.



Frozen Set

تا اینجا با set آشنا شدیم. اما نوع دیگری به نام Frozen Set نیز در پایتون وجود دارد.

تقریباً شبیه set است، با این تفاوت که قابل تغییر نیست.

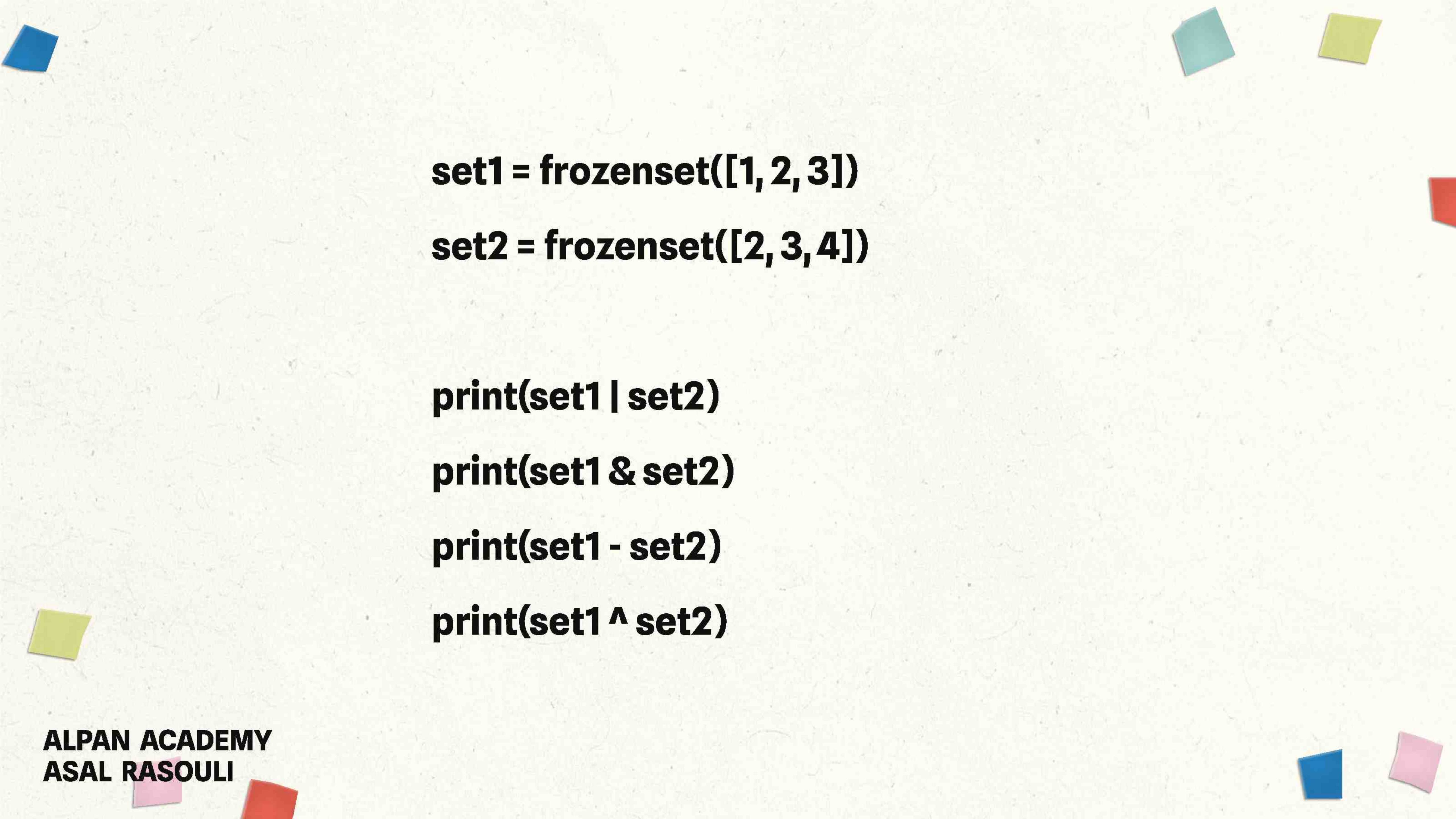
یعنی بعد از ساخت یک frozenset نمی‌توانیم عناصر آن را اضافه، حذف یا تغییر دهیم.

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

```
print(numbers)
```



عملیات روی Frozen Set



```
set1 = frozenset([1, 2, 3])
```

```
set2 = frozenset([2, 3, 4])
```

```
print(set1 | set2)
```

```
print(set1 & set2)
```

```
print(set1 - set2)
```

```
print(set1 ^ set2)
```

چرا Frozen Set داریم؟

گاهی می‌خواهیم مجموعه‌ای از داده‌ها داشته باشیم که نباید تغییر کنند.

در این شرایط frozenset انتخاب مناسبی است.

یکی دیگر از ویژگی‌های مهم frozenset این است که چون Hashable است، می‌تواند به‌عنوان:

- کلید Dictionary
 - عنصر یک Set دیگر
- استفاده شود.



تمرین کلاسی

66

یک برنامه بنویسید که:

- یک Set شامل ۳ نام دانش آموز ایجاد کند.
- یک دانش آموز جدید به آن اضافه کند.
- تعداد دانش آموزان را با len() نمایش دهد.

67



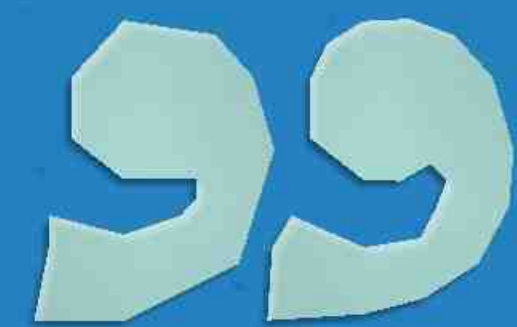
دو Set زیر را ایجاد کنید:

```
class1 = {"Ali", "Sara", "Maryam"}
```

```
class2 = {"Sara", "Nima", "Ali"}
```

سپس موارد زیر را نمایش دهید:

1. دانش‌آموزان مشترک
2. همه دانش‌آموزان
3. دانش‌آموزانی که فقط در کلاس اول هستند.



66

یک Set شامل ۴ عدد ایجاد کنید.
برنامه‌ای بنویسید که:

1. بررسی کند آیا عدد 5 در مجموعه وجود دارد یا
خیر.

2. عدد 2 را حذف کند.

3. مجموعه جدید را چاپ کند.

66

66

یک frozenset شامل ۴ عدد ایجاد کنید.
سپس:

1. تعداد عناصر آن را نمایش دهید.
2. بررسی کنید آیا عدد 3 در آن وجود دارد.
3. تلاش کنید یک عنصر به آن اضافه کنید و نتیجه را بررسی کنید.

66



تمرین‌های خانگی



یک Set شامل ۵ عدد ایجاد کنید و موارد زیر را
نمایش دهید:

- تعداد عناصر
- بزرگ‌ترین عدد
- کوچک‌ترین عدد
- مجموع اعداد

از توابع `min()`، `max()`، `len()` و `sum()` استفاده
کنید.



66

برنامه‌ای بنویسید که سه وسیله از کاربر دریافت کند و آن‌ها را در یک Set قرار دهد.
سپس:

- یک وسیله جدید اضافه کند.
- تعداد کل وسایل را نمایش دهد.
- بررسی کند آیا وسیله "کتاب" در مجموعه وجود دارد یا خیر.

67

66

دو Set شامل نام چند دانش آموز ایجاد کنید.
برنامه‌ای بنویسید که:

- دانش آموزان مشترک را پیدا کند.
- دانش آموزانی که فقط در مجموعه اول هستند را پیدا کند.
- دانش آموزانی که فقط در یکی از دو مجموعه هستند را پیدا کند.

66

66

یک Set شامل ۵ عدد ایجاد کنید.
سپس:

1. آن را به List تبدیل کنید.
2. را مرتب کنید List
3. مرتب شده را چاپ کنید List
4. تعداد عناصر را نمایش دهید.

66

66

یک frozenset شامل نام ۳ کشور ایجاد کنید.
سپس:

1. آن را چاپ کنید.
2. بررسی کنید آیا "Iran" در آن وجود دارد.
3. آن را به یک Set تبدیل کنید.
4. یک کشور جدید به Set اضافه کنید.
5. جدید را چاپ کنید Set

67

THANKS!

ALPAN ACADEMY
ASAL RASOULI