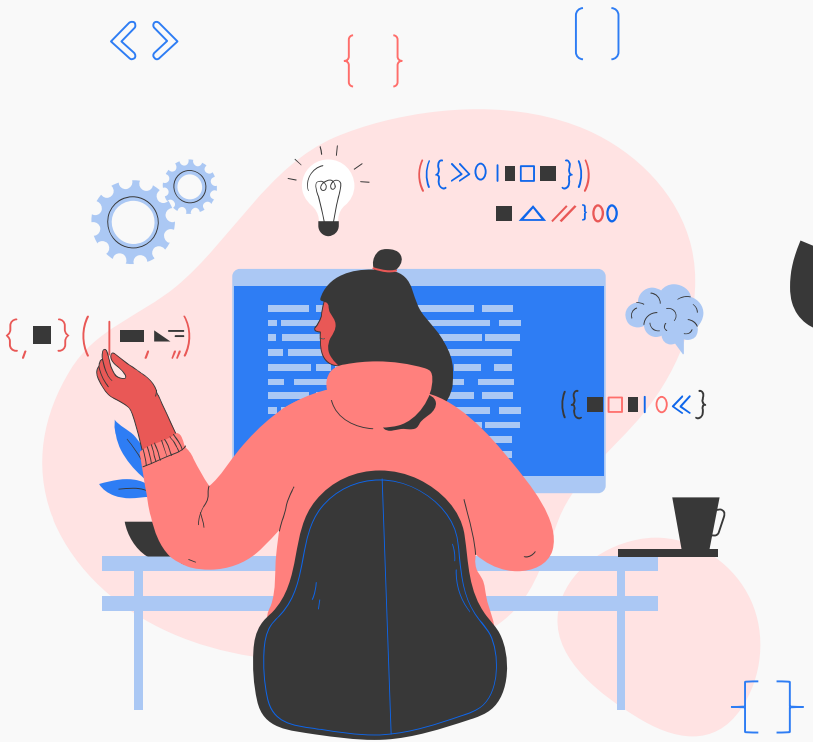


پایتون مقدماتی

دوره برنامه نویسی

محمد حسین شفیع آبادی

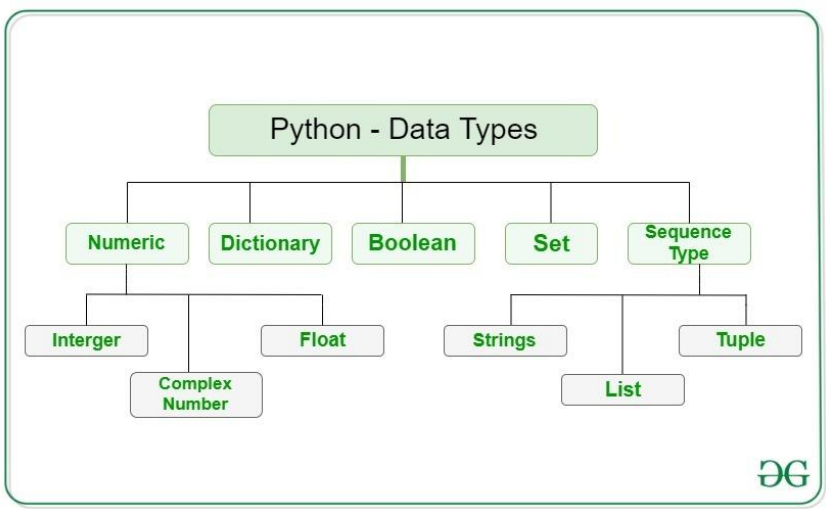
محمد حسین شفیع آبادی

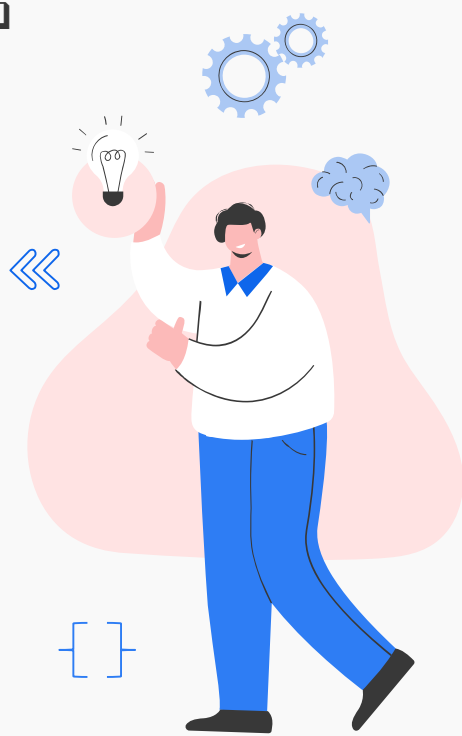
- دکتری کامپیوتر
- عضو هیات علمی دانشگاه آزاد
- مشاور و منتور توسعه کسب و کار
- مشاور و منتور راه اندازی و توسعه استارتاپ
- مدرس دوره های تخصصی کوچینگ فردی /سازمانی
- کوچ بین المللی چابکی سازمانی (Agile Coach)
- کوچ بین المللی رهبران سازمان (Leadership Coach)

+98 912 511 0460
Shafiabadi.ir
mhshafiaabadi@gmail.com

انواع داده در پایتون

- داده ها در پایتون انواع مختلف داریم.
- مجموعه ها Set
- داده های ترتیبی Strings, List, Tuple
- دیکشنری Dictionary





داده Tuple

- تاپل مجموعه ای از اشیاء برنامه نویسی پایتون است که بسیار شبیه به یک لیست است.
- توالی مقادیر ذخیره شده در یک تاپل می تواند از هر نوع باشد و با اعداد صحیح ایندکس می شوند.
- مقادیر یک تاپل از نظر نحوی با "کاما" از هم جدا می شوند.
- اگرچه لازم نیست، اما معمولاً تعریف یک تاپل با بستن دنباله مقادیر داخل پرانتز رایج است.

```
Tuple1 = ('Geeks', 'For')
```

Tuple داده

مثال

```
Tuple1 = ()
print("Initial empty Tuple: ")
print(Tuple1)

Tuple1 = ('Geeks', 'For')
print("\nTuple with the use of String: ")
print(Tuple1)

list1 = [1, 2, 4, 5, 6]
print("\nTuple using List: ")
print(tuple(list1))

Tuple1 = tuple('Geeks')
print("\nTuple with the use of function: ")
print(Tuple1)
```

```
Initial empty Tuple:
()

Tuple with the use of String:
('Geeks', 'For')

Tuple using List:
(1, 2, 4, 5, 6)

Tuple with the use of function:
('G', 'e', 'e', 'k', 's')
```

داده Tuple

مثال

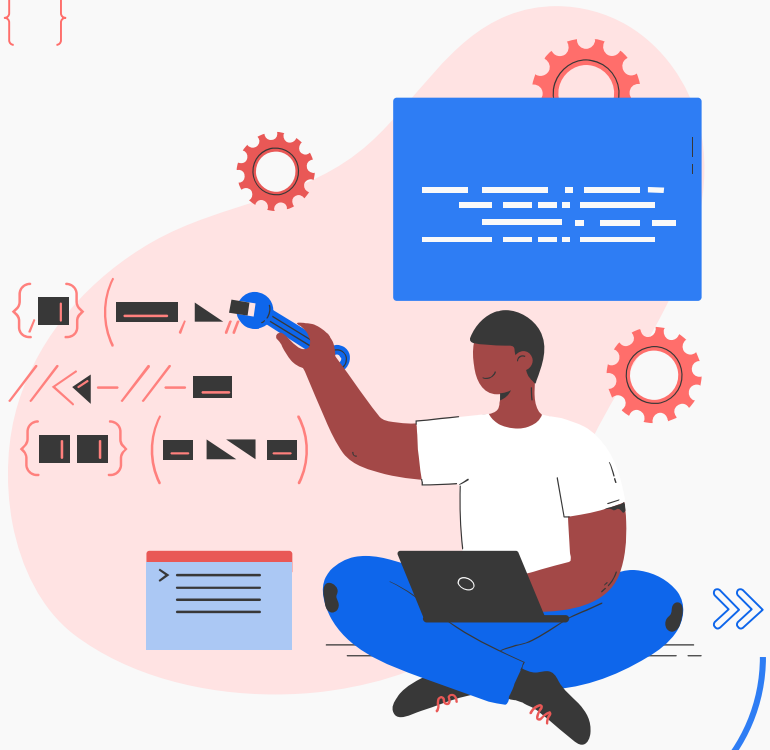
```
Tuple1 = (5, 'Welcome', 7, 'Geeks')
print("\nTuple with Mixed Datatypes: ")
print(Tuple1)

Tuple1 = (0, 1, 2, 3)
Tuple2 = ('python', 'geek')
Tuple3 = (Tuple1, Tuple2)
print("\nTuple with nested tuples: ")
print(Tuple3)

Tuple1 = ('Geeks',) * 3
print("\nTuple with repetition: ")
print(Tuple1)

Tuple1 = ('Geeks')
n = 5
print("\nTuple with a loop")
for i in range(int(n)):
    Tuple1 = (Tuple1,)
    print(Tuple1)
```

```
Tuple with Mixed Datatypes:
(5, 'Welcome', 7, 'Geeks')

Tuple with nested tuples:
((0, 1, 2, 3), ('python', 'geek'))

Tuple with repetition:
('Geeks', 'Geeks', 'Geeks')

Tuple with a loop
('Geeks',)
(('Geeks',),)
((('Geeks',),),)
(((('Geeks',),),),)
((((('Geeks',),),),),)
```

دسترسی به تاپل ها

- در پایتون، **تاپل‌ها تغییرناپذیر** هستند و معمولاً شامل دنباله‌ای از **عناصر ناهمگن** هستند که از طریق باز کردن بسته‌بندی یا نمایه‌سازی به آن‌ها دسترسی پیدا می‌کنند.
- **لیست** ها **قابل تغییر** هستند و عناصر آنها معمولاً **همگن** هستند و با تکرار روی لیست قابل دسترسی هستند.

نکته: در باز کردن بسته بندی تاپل، تعداد متغیرهای سمت چپ باید با تعدادی از مقادیر در تاپل داده شده برابر باشد.



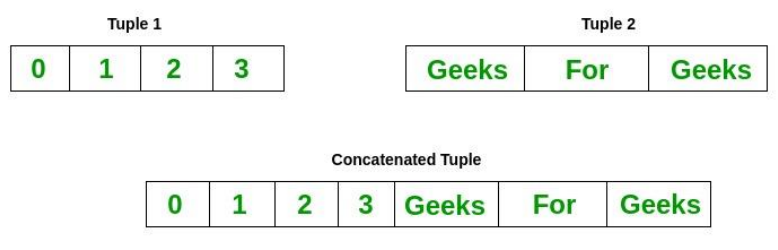
```
Tuple1 = tuple("Geeks")
print("\nFirst element of Tuple: ")
print(Tuple1[0])

Tuple1 = ("Geeks", "For", "Geeks")

a, b, c = Tuple1
print("\nValues after unpacking: ")
print(a)
print(b)
print(c)
```

```
First element of Tuple:
G

Values after unpacking:
Geeks
For
Geeks
```

الحاق تاپل ها

- الحاق تاپل فرآیند به هم پیوستن دو یا چند تاپل است.
- الحاق با استفاده از **عملگر "+"** انجام می شود.
- الحاق تاپل ها همیشه از انتهای تاپل اصلی انجام می شود.
- سایر عملیات های حسابی روی تاپل ها اعمال نمی شود.

نکته- فقط انواع داده های مشابه را می توان با الحاق ترکیب کرد، اگر یک لیست و یک تاپل با هم ترکیب شوند، خطا ایجاد می شود.



الحاق تاپل ها

مثال

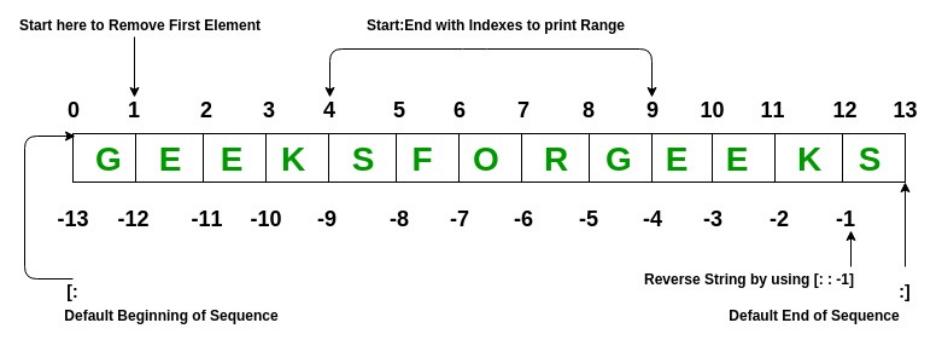
```
Tuple1 = (0, 1, 2, 3)
Tuple2 = ('Geeks', 'For', 'Geeks')

Tuple3 = Tuple1 + Tuple2

print("Tuple 1: ")
print(Tuple1)

print("\nTuple2: ")
print(Tuple2)

print("\nTuples after Concatenation: ")
print(Tuple3)
```

```
Tuple 1:
(0, 1, 2, 3)

Tuple2:
('Geeks', 'For', 'Geeks')

Tuples after Concatenation:
(0, 1, 2, 3, 'Geeks', 'For', 'Geeks')
```

برش تاپل

- برش یک تاپل برای واکشی محدوده یا تکه ای از عناصر فرعی خاص از یک تاپل انجام می شود.
- برش را می توان روی لیست ها و آرایه ها نیز انجام داد.
- نمایه سازی در یک لیست منجر به واکشی یک عنصر می شود در حالی که برش دادن امکان واکشی مجموعه ای از عناصر را فراهم می کند.

نکته- مقادیر منفی همچنین می تواند برای معکوس کردن دنباله تاپل ها استفاده شود.



برش تاپل

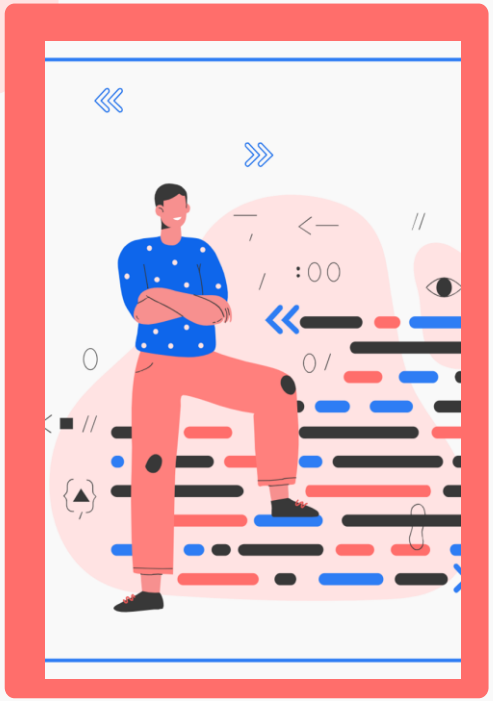
```
Tuple1 = tuple('GEEKSFORGEEKS')

# Removing First element
print("Removal of First Element: ")
print(Tuple1[1:])

# Reversing the Tuple
print("\nTuple after sequence of Element is reversed: ")
print(Tuple1[::-1])

# Printing elements of a Range
print("\nPrinting elements between Range 4-9: ")
print(Tuple1[4:9])
```

```
Removal of First Element:
('E', 'E', 'K', 'S', 'F', 'O', 'R', 'G', 'E', 'E', 'K', 'S')

Tuple after sequence of Element is reversed:
('S', 'K', 'E', 'E', 'G', 'R', 'O', 'F', 'S', 'K', 'E', 'E', 'G')

Printing elements between Range 4-9:
('S', 'F', 'O', 'R', 'G')
```

توابع و متد تاپل

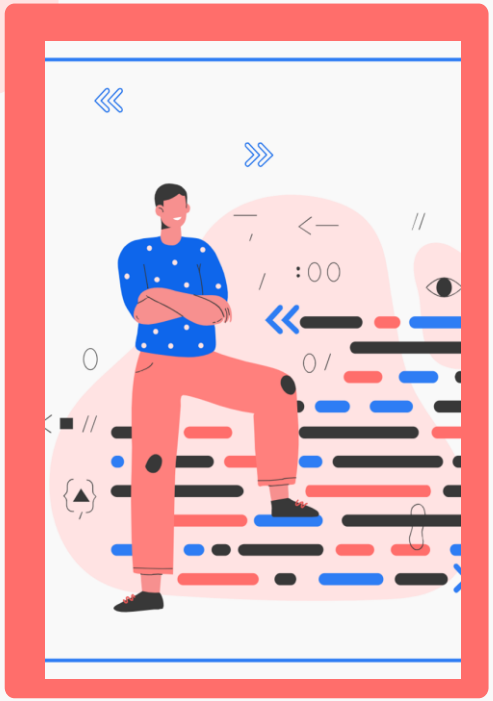
متد یا تاپل	توضیحات
del	برای پاک کردن یک تاپل استفاده می شود
index()	مقداری را در تاپل جستجو میکند و در صورت وجود شماره خانه آنرا بر میگرداند
count()	تعداد تکرار یک مقدار مشخص را برمیگرداند
all()	اگر همه مقادیر تاپل درست باشد و یا خالی باشد مقدار True برمیگرداند
len()	اندازه یا طول تاپل را برمیگرداند
min()	کوچکترین مقدار داخل تاپل را برمیگرداند
max()	بزرگترین مقدار داخل تاپل را برمیگرداند
sum()	مجموع اعداد داخل تاپل را برمیگرداند
sorted()	مقادیر داخل تاپل را مرتب و در یک لیست جدید بر میگرداند

مجموعه set



- مجموعه ها در پایتون، مجموعه ای نامرتب از انواع داده است که قابل تکرار، تغییرپذیر است و هیچ عنصر تکراری ندارد.
- ترتیب عناصر در یک مجموعه تعریف نشده است اگرچه ممکن است از عناصر مختلفی تشکیل شده باشد.
- مزیت اصلی استفاده از یک مجموعه، بر خلاف لیست، این است که یک روش بسیار بهینه برای بررسی اینکه آیا یک عنصر خاص در مجموعه موجود است یا خیر.

مجموعه set

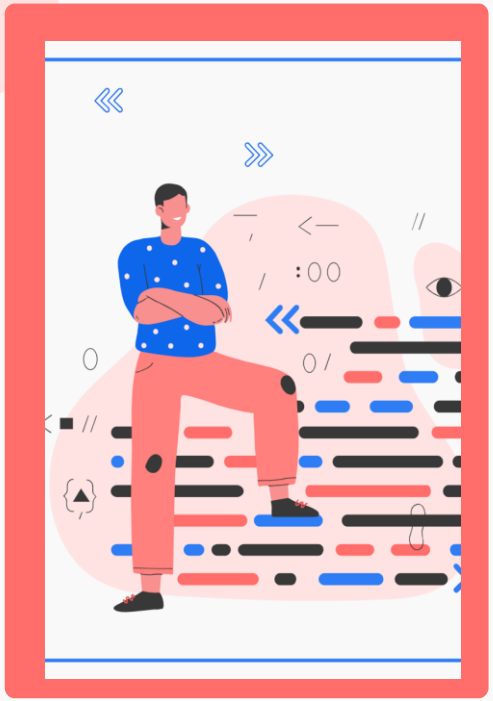


```
set1 = set()
print("Initial blank Set: ")
print(set1)

set1 = set("GeeksForGeeks")
print("\nSet with the use of String: ")
print(set1)

String = 'GeeksForGeeks'
set1 = set(String)
print("\nSet with the use of an Object: ")
print(set1)

set1 = set(["Geeks", "For", "Geeks"])
print("\nSet with the use of List: ")
print(set1)

t = ("Geeks", "for", "Geeks")
print("\nSet with the use of Tuple: ")
print(set(t))

d = {"Geeks": 1, "for": 2, "Geeks": 3}
print("\nSet with the use of Dictionary: ")
print(set(d))
```

مثال

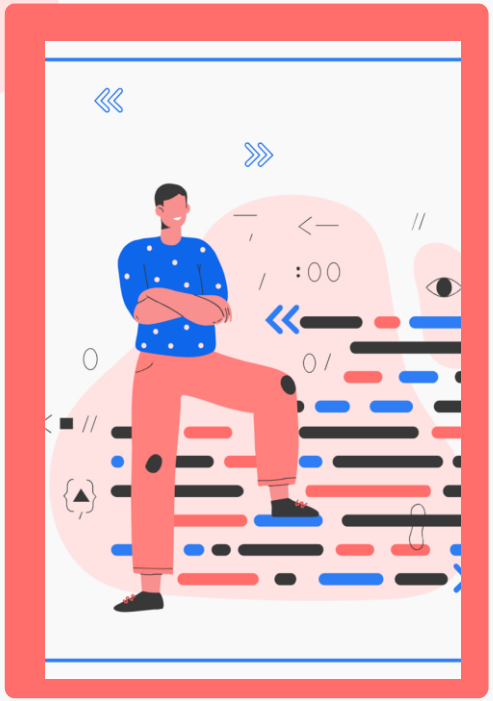
```
Initial blank Set:
set()

Set with the use of String:
{'e', 'G', 's', 'F', 'o', 'r', 'k'}

Set with the use of an Object:
{'e', 'G', 's', 'F', 'o', 'r', 'k'}

Set with the use of List:
{'For', 'Geeks'}

Set with the use of Tuple:
{'for', 'Geeks'}

Set with the use of Dictionary:
{'for', 'Geeks'}
```

مجموعه set



- یک مجموعه پایتون فقط شامل عناصر منحصربه‌فرد است، اما در زمان ایجاد مجموعه، چندین مقدار تکراری نیز می‌تواند ارسال شود.
- ترتیب عناصر در مجموعه پایتون تعریف نشده و تغییر ناپذیر است.
- نیازی نیست که نوع عناصر در یک مجموعه یکسان باشد، مقادیر مختلف نوع داده های مختلط نیز می توانند به مجموعه ارسال شوند.

مجموعه set

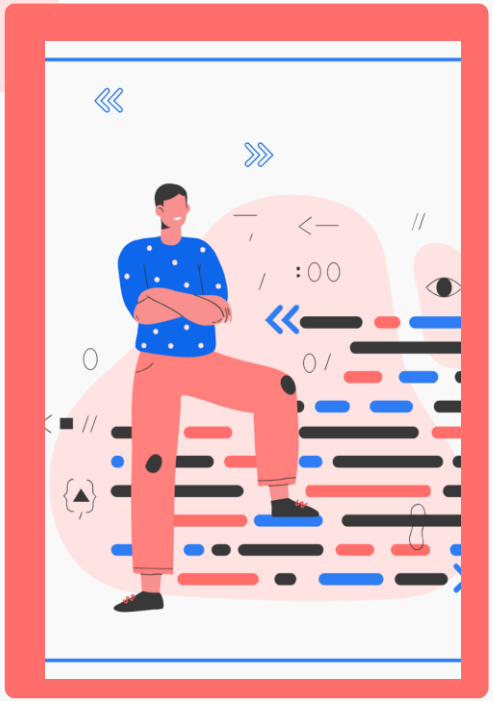


مثال

```
set1 = set([1, 2, 4, 4, 3, 3, 3, 6, 5])
print("\nSet with the use of Numbers: ")
print(set1)

set1 = set([1, 2, 'Geeks', 4, 'For', 6, 'Geeks'])
print("\nSet with the use of Mixed Values")
print(set1)
```

```
Set with the use of Numbers:
{1, 2, 3, 4, 5, 6}

Set with the use of Mixed Values
{1, 2, 4, 6, 'Geeks', 'For'}
```

مجموعه set

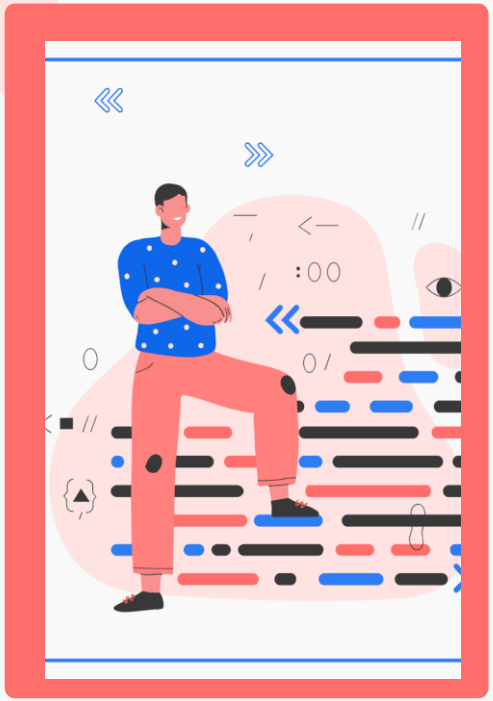


افزودن عناصر به مجموعه در پایتون

برای اضافه کردن مقداری به مجموعه در پایتون از دو متد می توان استفاده کرد

- ❑ با استفاده از متد add()
- ❑ استفاده از متد update()

مجموعه set

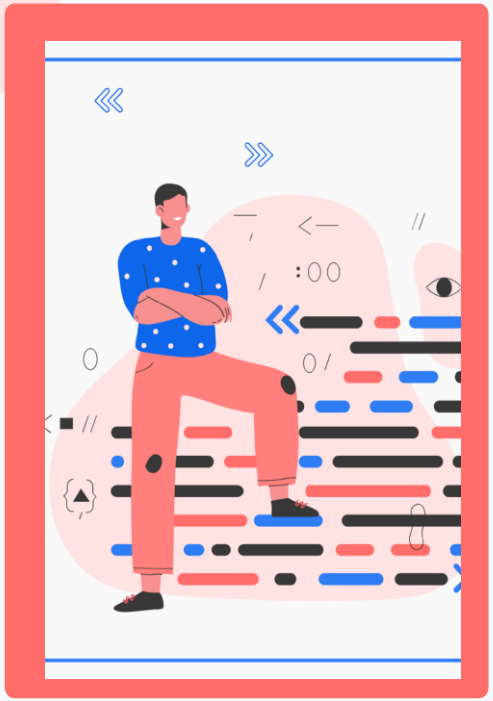


متد add()

```
set1 = set()
print("Initial blank Set: ")
print(set1)

set1.add(8)
set1.add(9)
set1.add((6, 7))
print("\nSet after Addition of Three elements: ")
print(set1)

for i in range(1, 6):
    set1.add(i)
print("\nSet after Addition of elements from 1-5: ")
print(set1)
```

```
Initial blank Set:
set()

Set after Addition of Three elements:
{8, 9, (6, 7)}

Set after Addition of elements from 1-5:
{1, 2, 3, (6, 7), 4, 5, 8, 9}
```

مجموعه set

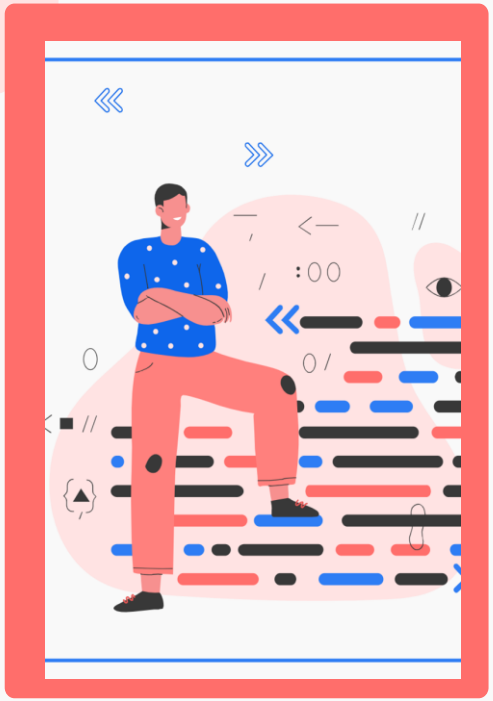


متد update()

```
set1 = set([4, 5, (6, 7)])
set1.update([10, 11])
print("\nSet after Addition of elements using Update: ")
print(set1)
```

```
Set after Addition of elements using Update:
{4, 5, (6, 7), 10, 11}
```

مجموعه set

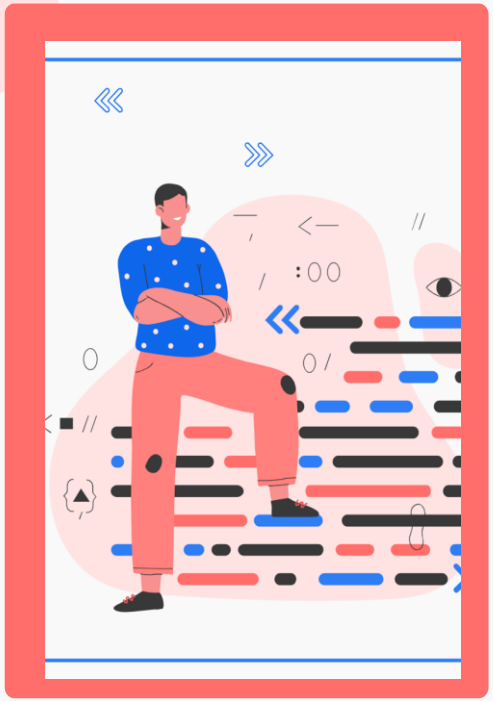


حذف عناصر از مجموعه در پایتون

برای حذف عضوی از مجموعه از متد های زیر می توان استفاده کرد:

- ❑ با استفاده از روش ()remove یا ()Discard
- ❑ استفاده از روش ()pop
- ❑ با استفاده از روش ()clear

مجموعه set

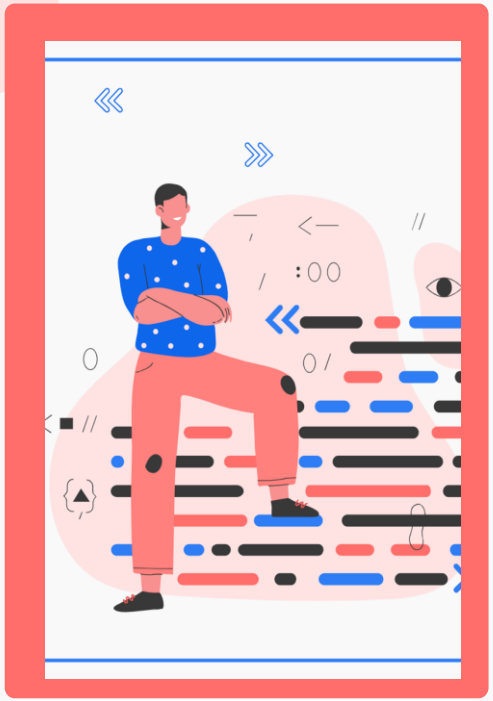


```
set1 = set([1, 2, 3, 4, 5, 6,
            7, 8, 9, 10, 11, 12])
print("Initial Set: ")
print(set1)

set1.remove(5)
set1.remove(6)
print("\nSet after Removal of two elements: ")
print(set1)

set1.discard(8)
set1.discard(9)
print("\nSet after Discarding two elements: ")
print(set1)

for i in range(1, 5):
    set1.remove(i)
print("\nSet after Removing a range of elements: ")
print(set1)
```

با استفاده از روش ()remove یا ()Discard

```
Initial Set:
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

Set after Removal of two elements:
{1, 2, 3, 4, 7, 8, 9, 10, 11, 12}

Set after Discarding two elements:
{1, 2, 3, 4, 7, 10, 11, 12}

Set after Removing a range of elements:
{7, 10, 11, 12}
```

مجموعه set

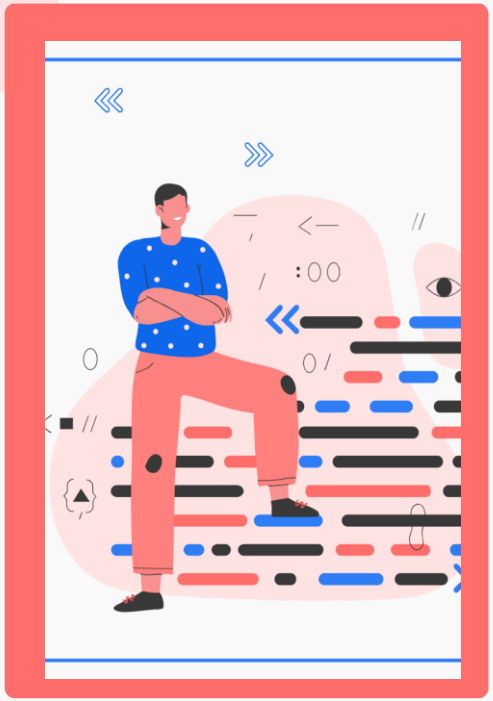


با استفاده از روش ()pop

```
set1 = set([1, 2, 3, 4, 5, 6,
            7, 8, 9, 10, 11, 12])
print("Initial Set: ")
print(set1)

set1.pop()
print("\nSet after popping an element: ")
print(set1)
```

```
Initial Set:
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

Set after popping an element:
{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}
```

مجموعه set



با استفاده از روش clear()

```
set1 = set([1,2,3,4,5])
print("\n Initial set: ")
print(set1)


set1.clear()
print("\nSet after clearing all the elements: ")
print(set1)
```

```
 Initial set:
{1, 2, 3, 4, 5}

Set after clearing all the elements:
set()
```

دیکشنری dictionary

دیکشنری در پایتون ساختمان داده ای است که اطلاعات در آن به شکل key:value ذخیره می شود.

```
Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}
print(Dict)
```

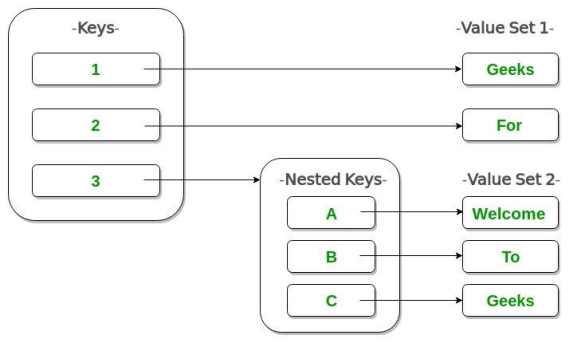
```
{1: 'Geeks', 2: 'For', 3: 'Geeks'}
```

dict_var = {key1 : value1, key2 : value2, …..}

دیکشنری dictionary

دیکشنری در پایتون ساختمان داده ای است که اطلاعات در آن به شکل key:value ذخیره می شود.

```
Dict = {1: 'Geeks', 2: 'For',
        3: {'A': 'Welcome', 'B': 'To', 'C': 'Geeks'}}

print(Dict)
```



دیکشنری dictionary

```
Dict = {}
print("Empty Dictionary: ")
print(Dict)
Dict[0] = 'Geeks'
Dict[2] = 'For'
Dict[3] = 1
print("\nDictionary after adding 3 elements: ")
print(Dict)

Dict['Value_set'] = 2, 3, 4
print("\nDictionary after adding 3 elements: ")
print(Dict)

Dict[2] = 'Welcome'
print("\nUpdated key value: ")
print(Dict)
Dict[5] = {'Nested': {'1': 'Life', '2': 'Geeks'}}
print("\nAdding a Nested Key: ")
print(Dict)
```

```
Empty Dictionary:
{}

Dictionary after adding 3 elements:
{0: 'Geeks', 2: 'For', 3: 1}

Dictionary after adding 3 elements:
{0: 'Geeks', 2: 'For', 3: 1, 'Value_set': (2, 3, 4)}

Updated key value:
{0: 'Geeks', 2: 'Welcome', 3: 1, 'Value_set': (2, 3, 4)}

Adding a Nested Key:
{0: 'Geeks', 2: 'Welcome', 3: 1, 'Value_set': (2, 3, 4), 5: {'Nested': {'1': 'Life', '2': 'Geeks'}}}
```