SET_VHA

November 1, 2025

1 Sets

A set is an unordered collection of items. Every set element is unique (no duplicates) and must be immutable (cannot be changed).

However, a set itself is mutable. We can add or remove items from it.

Sets can also be used to perform mathematical set operations like union, intersection, symmetric difference, etc.

Characterstics:

Unordered

Mutable

No Duplicates

Can't contain mutable data types

2 creating set

[6]:
```python
# empty
s = set()
print(s)
print(type(s))
s={}
print(s)
print(type(s))
# 1D and 2D
s1 = {1,2,3}
print(s1)
#s2 = {1,2,3,{4,5}}
#print(s2)
# homo and hetro
s3 = {1,'hello',4.5,(1,2,3)}
print(s3)
# using type conversion

s4 = set([1,2,3])
```

```
print(s4)
# duplicates not allowed
s5 = {1,1,2,2,3,3}
print(s5)
# set can't have mutable items
s6 = {1,2,[3,4]}
print(s6)
```

```
set()
<class 'set'>
{}
<class 'dict'>
{1, 2, 3}
{1, (1, 2, 3), 4.5, 'hello'}
{1, 2, 3}
{1, 2, 3}
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_25024\4001107056.py in <module>
     22 print(s5)
     23 # set can't have mutable items
---> 24 s6 = {1,2,[3,4]}
     25 print(s6)

TypeError: unhashable type: 'list'
```

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

[4]:
```
True
```

3 Accessing Items

[7]:
```
s1 = {1,2,3,4}
s1[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_25024\1334365164.py in <module>
      1 s1 = {1,2,3,4}
----> 2 s1[0]

TypeError: 'set' object is not subscriptable
```

```
[8]: s1 = {1,2,3,4}
     s1[-1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_25024\2343478921.py in <module>
      1 s1 = {1,2,3,4}
----> 2 s1[-1]

TypeError: 'set' object is not subscriptable
```

```
[9]: s1 = {1,2,3,4}
     s1[0:2]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_25024\3613277626.py in <module>
      1 s1 = {1,2,3,4}
----> 2 s1[0:2]

TypeError: 'set' object is not subscriptable
```

4 Editing Items

```
[10]: s1= {1,2,3,4}
      s1[0] = 100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_25024\3149387887.py in <module>
      1 s1= {1,2,3,4}
----> 2 s1[0] = 100

TypeError: 'set' object does not support item assignment
```

5 Adding Items

add

update

```
[11]: S = {1,2,3,4}
      # add
```

```
S.add(5)
print(S)
```

```
{1, 2, 3, 4, 5}
```

[12]:
```
S = {1,2,3,4}
# update
S.update([5,6,7])
print(S)
```

```
{1, 2, 3, 4, 5, 6, 7}
```

6 Deleting Items

7 del

8 discard

9 remove

10 pop

11 clear

[14]:
```
# del
s = {1,2,3,4,5}
print(s)
del s
print(s)
```

```
{1, 2, 3, 4, 5}
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_25024\1866925707.py in <module>
      3 print(s)
      4 del s
----> 5 print(s)

NameError: name 's' is not defined
```

[15]:
```
# del
s = {1,2,3,4,5}
print(s)
del s[0]
print(s)
```

```
{1, 2, 3, 4, 5}
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_25024\415133016.py in <module>
      2 s = {1,2,3,4,5}
      3 print(s)
----> 4 del s[0]
      5 print(s)

TypeError: 'set' object doesn't support item deletion
```

[17]:
```
#discard
s = {1,2,3,4,5}
print(s)
s.discard(3)
print(s)
```

```
{1, 2, 3, 4, 5}
{1, 2, 4, 5}
```

[16]:
```
#discard
s = {1,2,3,4,5}
print(s)
s.discard(50)
print(s)
```

```
{1, 2, 3, 4, 5}
{1, 2, 3, 4, 5}
```

[18]:
```
#remove
s = {1,2,3,4,5}
print(s)
s.remove(5)
print(s)
```

```
{1, 2, 3, 4, 5}
{1, 2, 3, 4}
```

[19]:
```
#remove
s = {1,2,3,4,5}
print(s)
s.remove(50)
print(s)
```

```
{1, 2, 3, 4, 5}
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
```

```
~\AppData\Local\Temp\ipykernel_25024\335069362.py in <module>
      2 s = {1,2,3,4,5}
      3 print(s)
----> 4 s.remove(50)
      5 print(s)

KeyError: 50
```

[24]:
```
#pop
s = {7,2,3,4,5}
print(s)
s.pop()
print(s)
```

```
{2, 3, 4, 5, 7}
{3, 4, 5, 7}
```

[25]:
```
#clear
s = {7,2,3,4,5}
print(s)
s.clear()
print(s)
```

```
{2, 3, 4, 5, 7}
set()
```

[65]:
```
s={1,2,3,4}
c={5,6,7,8}
print(s>c)
p
```

```
False
```

12 Set Operation

union(|)

intersection(&)

Difference(-)

Symmetric Difference(^)

Membership Test

Iteration

[28]:
```
s1 = {1,2,3,4,5}
s2 = {4,5,6,7,8}
```

```
# Union(|)
print(s1 | s2,"Union(|)")
# Intersection(&)
print(s1 & s2,"Intersection(&)")
# Difference(-)
print(s1 - s2,"s1-s2 Difference(-)")
print(s2 - s1,"s2-s1 Difference(-)")
# Symmetric Difference(^)
print(s1 ^ s2,"Symmetric Difference(^)")
# Membership Test
print(1 not in s1)
# Iteration
for i in s1:
  print(i)
```

```
{1, 2, 3, 4, 5, 6, 7, 8} Union(|)
{4, 5} Intersection(&)
{1, 2, 3} s1-s2 Difference(-)
{8, 6, 7} s2-s1 Difference(-)
{1, 2, 3, 6, 7, 8} Symmetric Difference(^)
False
1
2
3
4
5
```

13 Set Functions

len

sum

min

max

sorted

[30]:
```
# len/sum/min/max/sorted
s = {3,1,4,5,2,7}
print(len(s))
print(sum(s))
print(min(s))
print(max(s))
print(sorted(s,reverse=True))
```

```
6
22
1
```

```
7
[7, 5, 4, 3, 2, 1]
```

14 Set Functions

union

update

intersection

intersection_update

difference

difference_update

symmetric_difference

symmetric_difference_update

isdisjoint

issubset

issuperset

copy

[31]:
```
# union/update
s1 = {1,2,3,4,5}
s2 = {4,5,6,7,8}

# s1 | s2
s1.union(s1)

s1.update(s2)
print(s1)
print(s2)
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
{4, 5, 6, 7, 8}
```

[47]:
```
# intersection/intersection_update
s1 = {1,2,3,4,5}
s2 = {4,5,6,7,8}

print(s1.intersection(s2))

s1.intersection_update(s2)
print(s1)
print(s2)
```

```
{4, 5}
{4, 5}
{4, 5, 6, 7, 8}
```

[46]:
```
# difference/difference_update
s1 = {1,2,3,4,5}
s2 = {4,5,6,7,8}

print(s1.difference(s2))

s1.difference_update(s2)

print(s1)
print(s2)
```

```
{1, 2, 3}
{1, 2, 3}
{4, 5, 6, 7, 8}
```

[49]:
```
# symmetric_difference/symmetric_difference_update
s1 = {1,2,3,4,5}
s2 = {4,5,6,7,8}

s1.symmetric_difference(s2)

s1.symmetric_difference_update(s2)
print(s1)
print(s2)
```

```
{1, 2, 3, 4, 5}
{4, 5, 6, 7, 8}
```

[34]:
```
s1 = {1,2,3,4}
s2 = {7,8,5,6}

s1.isdisjoint(s2)
```

[34]: True

[51]:
```
s1 = {1,2,3,4}
s2 = {1,8,5,6}

s1.isdisjoint(s2)
```

[51]: False

[35]:
```
s1 = {1,2,3,4,5}
s2 = {3,4,5}
```

```
s1.issuperset(s2)
```

[35]:
```
True
```

[36]:
```
s1 = {1,2,3,4,5}
s2 = {3,4,5}

s2.issubset(s1)
```

[36]:
```
True
```

[38]:
```
# copy
s1 = {1,2,3}
s2 = s1.copy()

print(s1)
print(s2)
```

```
{1, 2, 3}
{1, 2, 3}
```

15 Frozenset

Frozen set is just an immutable version of a Python set object

[39]:
```
# create frozenset
fs1 = frozenset([1,2,3])
fs2 = frozenset([3,4,5])

fs1 | fs2
```

[39]:
```
frozenset({1, 2, 3, 4, 5})
```

[ ]:
```
# what works and what does not
# works -> all read functions
# does't work -> write operations
```

[53]:
```
# 2D sets
fs = frozenset([1,2,frozenset([3,4])])
fs
```

[53]:
```
frozenset({1, 2, frozenset({3, 4})})
```

16 Set Comprehension

[54]:
```
# examples

{i**2 for i in range(1,11) if i>5}
```

[54]:
```
{36, 49, 64, 81, 100}
```

17 Write a program to find set of common elements in three lists using sets.

Input : ar1 = [1, 5, 10, 20, 40, 80]

```
ar2 = [6, 7, 20, 80, 100]

ar3 = [3, 4, 15, 20, 30, 70, 80, 120]
```

Output : [80, 20]

[55]:
```
# write your code here
ar1 = [1, 5, 10, 20, 40, 80]
ar2 = [6, 7, 20, 80, 100]
ar3 = [3, 4, 15, 20, 30, 70, 80, 120]

s1 = set(ar1)
s2 = set(ar2)
s3 = set(ar3)

result = list((s1 & s2) & s3)
print(result)
```

```
[80, 20]
```

18 Write a program to count unique number of vowels using sets in a given string. Lowercase and upercase vowels will be taken as different.

Input:

Str1 = “hands-on data science mentorship progrAm with live classes at affordable fee only on CampusX”

Output:

No of unique vowels-6

[56]:
```
# write your code here
vowels = set('aeiouAEIOU')
```

```
s = set("hands-on data science mentorship progrAm with live classes at␣
↪affordable fee only on CampusX")

print('No of unique vowels-',len(s & vowels))
```

```
No of unique vowels- 6
```

19 Write a program to Check if a given string is binary string of or not.

A string is said to be binary if it's consists of only two unique characters.

Take string input from user.

```
Input: str = "01010101010"
Output: Yes

Input: str = "1222211"
Output: Yes

Input: str = "Campusx"
Output: No
```

[58]:
```
# write your code here
s = "010101010103"

if len(set(s)) == 2:
  print('binary')
else:
  print('not binary')
```

```
not binary
```

20 find union of n arrays.

Example 1:

Input:

[[1, 2, 2, 4, 3, 6], [5, 1, 3, 4], [9, 5, 7, 1], [2, 4, 1, 3]] Output:

[1, 2, 3, 4, 5, 6, 7, 9]

[61]:
```
# write your code here
L = [[1, 2, 2, 4, 3, 6],
 [5, 1, 3, 4],
 [9, 5, 7, 1],
 [2, 4, 1, 3]]

s = set()
```

```python
for i in L:
  s.update(i)

print(s)
```

```
{1, 2, 3, 4, 5, 6, 7, 9}
```