

Set

Set 1 = $\{4, 5, 7, 2, 7, 0\}$
`bool(set1) → True`

Set 1 = `set({ })`
class set → False

dir(set1)
"add", "clear", "copy", "discard", "pop",
"difference", "difference_update",
"intersection", "intersection_update",
"symmetric_difference", "symmetric_difference_update",
"union", "update"

Set properties

sets are unordered, immutables/unchangeable
and no dups

lists are indexable, changeable, curs
multiple datatypes in one set is allowed

multiple items in set
and no dups

Set 1 = $\{1, 2, 3\}$ pf(set1) → 1, 2, 3
unordered

unchangeable
set1[0] = 6 X

No dups
 $\{1, 2, 3, 2\} \rightarrow \underline{\underline{\{1, 2, 3\}}}$

↳ $\text{set}4 = \{4, 2, "a", "python", \text{True}, 3.5, \text{False}\}$
len(set4) $\rightarrow 7$

↳ Accessible

```
for i in set1:  
    print(i, end=" ")
```

↳ $\text{set}1 = \{1, 2, 1, 3, 2, 5, 6, 7, 9\}$
 $\text{set}1.add(10)$
 $\text{set}1 \rightarrow \text{set}()$

↳ $\text{set}1.clear() \rightarrow \text{set}()$

↳ $\text{set}1 \rightarrow \text{set}1.\text{copy}()$

↳ $\text{set}2 = \text{set}1$
 $\text{id}(\text{set}2) \neq \text{id}(\text{set}1)$ } different

$\text{set}1 = \{1, 2, 1, 3, 2, 5, 6, 7, 9\}$
 $\text{set}2 = \{1, 2, 1, 4, 5, 5, 5, 6, 7, 9, 10\}$

$\text{set}3 = \text{set}1 - \text{difference}(\text{set}2)$

↳ $\text{set}3 \rightarrow \text{set}1 - \text{difference}(\text{set}2)$

↳ $\text{set}1 - \text{difference} - \text{update}(\text{set}2)$

↳ $\{3\}$

↳ difference

$$S_1 = \{1, 2, 1, 3, 2, 5, 6, 7, 9\}$$
$$S_2 = \{1, 2, 1, 4, 5, 6, 7, 9, 10\}$$

Set 2

$$S_3 = S_1 \cdot \text{difference}(S_2)$$

↳ 3rd intersection

intersection (set 2)

$$S_3 = S_1 \cdot \text{intersection}(S_2)$$

$$\{1, 2, 5, 6, 7, 9\}$$

S1 · intersection · update

$$\{1, 2, 5, 6, 7, 9\}$$

union / update

$$S_1 = \{1, 2, 1, 3, 2, 5, 6, 7, 9\}$$

$$S_2 = \{1, 2, 1, 4, 5, 6, 7, 9, 10\}$$

S1.pop() # 1

S1.remove(9) # {2, 3, 5, 6, 7}

S1.remove(1)

$$S_1 = \{1, 2, 3, 5, 6, 7, 9, 11\}$$

$$S_2 = \{1, 2, 4, 5, 6, 7, 9, 10, 12\}$$

S1.union(S2)

$$\{1, 2, 1, 4, 5, 5, 6, 7, 9, 10, 12\}$$

↳ union / update

Update

S1. update (S2)

S1 # {1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12} ✓

S2 # {1, 2, 4, 5, 6, 7, 9, 10, 12}

S1 # {1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12} ✓

S2 # {1, 2, 4, 5, 6, 7, 9, 10, 12} ✓

~~Add()~~ !
~~Index~~

S2 = {1, 2, 3, 2, 4, 5, 5, 6}

S3 = S1, 2, 5, 'jack' time, 3 + 4j

Collection of heterogeneous, distinct elements

↳ Indexing/slicing are not allowed in sets

S3[2] X Type Error

S3[1: 43] X Type Error

set (iterable)

Creation of Sets

S1 = {1, 2, 3, 4, 5, 6} } set (Type)

S2 = set([1, 2, 3, 4, 5, 6])

S3 = set('python')

S4 = set('xyz') } set

S5 = set() } set

S6 = {x, y} dict

Set 3

↳ $s1 = [10, 20, 30, 40, 50]$

$0/p = \{40, 10, 50, 20, 30\}$

for i in s1:
 print(i, end="")

40, 10, 50, 20, 30

unordered

↳ mutable

$s1 = [10, 20, 30, 40, 50]$

$s1.append(60)$

$\{40, 10, 50, 20, 60, 30\}$

$s1.append("hi")$

$\{40, 10, 50, 20, "hi", 60, 30\}$

$s1.append([1, 2, 3])$

$\{40, "hi", 10, [1, 2, 3], 50, 20, 60, 30\}$

$s1.append([1, 2, 3])$

List is mutable

TypeError :-

unstable type: list

↳ Methods

remove

$s1 = [10, 20, 30, 40, 50]$

$s1.remove(30)$

$\{40, 10, 50, 20\}$ ✓

mutable and
not hashable.

pop $s1.pop()$ # 40

== don't know which element
 will be removed.

union / Intersection

$S_1 = \{1, 2, 3, 5, 7\}$
 $S_2 = \{5, 7, 9, 10, 11\}$
 $S_3 = S_1 \cup S_2$
 $\{1, 2, 3, 5, 7, 9, 10, 11\}$
 $\{1, 2, 3, 5, 7, 9, 10, 11\}$

$S_1 \text{.update}(S_2)$
 $\{1, 2, 3, 5, 7, 9, 10, 11\}$

Intersection / Intersection_update

$S_1 = \{1, 2, 3, 5, 7\}$
 $S_2 = \{5, 7, 9, 10, 11\}$
 $S_3 = S_1 \text{.intersection}(S_2)$
 $\{5, 7\}$

$S_1 \text{.intersection_update}(S_2)$
 $\{5, 7\}$

difference / Difference_update

$S_1 = \{1, 2, 3, 5, 7\}$
 $S_2 = \{5, 7, 9, 10, 11\}$
 $S_3 = S_1 \text{.difference}(S_2)$
 $\{1, 2, 3\}$

$S_1 \text{.difference_update}(S_2)$
 $\{1, 2, 3\}$

Symmetric difference

$$S_1 = \{1, 2, 3, 5, 7\}$$

$$S_2 = \{5, 7, 9, 10, 11\}$$

$$S_3 = S_1 \text{ symmetric-difference } (S_2)$$

$$\{1, 2, 3, 9, 10, 11\}$$

Set 4

Operations

$$| \rightarrow \text{union}$$

$$|= \rightarrow \text{update}$$

$$- \rightarrow \text{difference}$$

$$-- \rightarrow \text{difference update}$$

$$\$ \rightarrow \text{Intersection}$$

$$\$ = \rightarrow \text{Intersection update}$$

$$1 \rightarrow \text{symmetric diff}$$

$$1 = \rightarrow \text{symmetric list}$$

$$\text{update}$$

Union

$$S_1 = \{1, 2, 3, 5, 7\}$$

$$S_2 = \{5, 7, 9, 10, 11\}$$

$$S_3 = S_1 \cup S_2$$

$$\{1, 2, 3, 5, 7, 9, 10, 11\}$$

$$\text{Update } S_1 := S_2 \rightarrow S_1$$

Intersection

$$S_1 = \{1, 2, 3, 5, 7\}$$

$$S_2 = \{5, 7, 9, 10, 11\}$$

$$S_3 = S_1 \cap S_2$$

$$= \{5, 7\}$$

$$S_1 \cap S_2$$

$$\{5, 7\}$$

Difference

$$S_1 = \{1, 2, 3, 5, 7\}$$

$$S_2 = \{5, 7, 9, 10, 11\}$$

$$S_3 = S_1 - S_2$$

$$= S_1 - S_2$$

$$\{1, 2, 3\}$$

$$\{1, 2, 3\} // \text{diff update}$$

symmetric

Difference

$$S_1 = \{1, 2, 3, 5, 7\}$$

$$S_2 = \{5, 7, 9, 10, 11\}$$

$$S_3 = S_1 \Delta S_2$$

$$\{1, 2, 3, 9, 10, 11\}$$

Update

$$S_1 \Delta S_2$$

$$\{1, 2, 3, 5, 9, 10, 11\}$$

Set S

#add(element)
#update(Iterable)

#copy()

#pop() → Remove top most element

#remove(element)

#discard(element)

#clear()

#add {10, 20, 30, 40, 50}

S1 = {10, 20, 30, 40, 50} ✓

S1.add(60) ✓

S1.add(60, 70) ✗

S1.add((60, 70)) ✓

S1.add([60, 70]) ✗

S1.add({10, 20, 30, 40, 50, 60, 70}) ✗

S1.add([60, 70]) // unhashable type

[60, 70] ✗

tuple work

#update

S1 = {10, 20, 30, 40, 50}

S1.update((60, 70))

S1.update({60, 70})

⇒ S1.update('busy')

S1.update({'r', 's', 't', 'u', 'v', 'w'})

$\hookrightarrow \text{copy}$

$s_1 = S(0, 20, 30, 40, 50)$

$s_2 = s_1.\text{copy}()$

$\text{id}(s_1), \text{id}(s_2)$

$\text{fx } 16^9$

$\hookrightarrow \text{pop}()$

$s_1 = S(0, 20, 30, 40, 50)$

$s_1.\text{pop}() \# 40$

$s_1.\text{pop}() \# 10$

$s_1 = S(0, 20, 30, 40, 50)$

$\hookrightarrow \text{Discard}$

$s_1 = S(0, 20, 30, 40, 50)$

$s_1.\text{discard}(30)$

$s_1.\text{discard}(60) \# \text{no error}$

$s_1.\text{discard}(60) \# \text{no error}$

$\Rightarrow \text{remove}$

$s_1 = S(0, 20, 30, 40, 50)$

$s_1.\text{remove}(30)$

$\# \text{none}$

$s_1.\text{remove}(100)$

$s_1.\text{remove}(100) \# \text{keyError}$

remove

$s_1 = \{10, 20, 30, 40, 50\}$

$s_1.remove(20, 30)$

$s_1 = \{40, 50\}$

Set 6

clear

$s_1 = \{10, 20, 30, 40, 50\}$

$s_1.clear()$

$s_1 = \{\}$

del

$s_1 = \{10, 20, 30, 40, 50\}$

~~del s_1~~

~~del s_1~~

comprehensions

$s = \{x \text{ for } x \in \text{iterable}\}$

$s = \{x \text{ for } x \in \text{iterable} \text{ if } \text{condition}\}$

$s_1 = \{x \text{ for } x \in \text{range}(1, 5)\}$

$s_2 = \{x \text{ for } x \in \text{range}(0, 4)\}$

$\{0, 1, 2, 3\}$

$s_3 = \{x.lower() \text{ for } x \text{ in } "apple"\}$

$\{'a', 'p', 'l', 'e'\}$

set comprehension

$\{ \text{set} \}$ set $\{ \text{list} \}$ list $\{ \text{allow dups} \}$ allow dups
 $\{ \text{no dups} \}$ no dups

Set comprehension

Set

~~Set exercise:~~

word_set = ['plea', 'medical', 'listens', 'sleep',
 'silent', 'pole', 'enlist',
 'deciml', 't']

print(sorted('plea'))
print(sorted('sleep'))

['a', 'e', 'l', 'p', 'r']
['g', 'e', 'l', 's']

result = set(c)

for word1 in word_set:
 for word2 in word_set:
 if word1 != word2 and sorted(word1) == sorted(word2):
 pair = tuple(sorted([word1, word2]))

result.add(pair)

for pair in result:
 print(pair)

JACCARD

Set 7

import re

str1 = ('Time is the most valuable we have', 'and once lost, it never returns')

str2 = ('we have set time back once it's gone - it's truly the most valuable resource')

words1 = re.findall(r'\w+', str1.lower())

words2 = re.findall(r'\w+', str2.lower())

wset1 = set(words1)

['time', 'is', 'the', 'most', ...]

wset2 = set(words2)

['we', 'have', ...]

common = wset1 & wset2

univ = wset1 | wset2

ratio = len(common) / len(univ) $\rightarrow 0.40$

ratio = len(common) / len(univ) $\rightarrow 0.40$