|  |
| --- |
| **Python Sets** |

**Name : Arnob Reduan**

* **Introduction** : Sets are used to store multiple items in a single variable. A set is a collection which is unordered, unchangeable\*, and unindexed. \* Note: Set items are unchangeable, but you can remove items and add new items. [1]

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.[2]

[1]References : <https://www.w3schools.com/python/python_sets.asp>

[2]References : <https://www.programiz.com/python-programming/set>

* **Creating a set :** [3] The set can be created by enclosing the comma-separated immutable items with the curly braces {}. Python also provides the set() method, which can be used to create the set by the passed sequence.

Days = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"}

print(Days)

print(type(Days))

print("looping through the set elements ... ")

for i in Days:

print(i)

**#output:**

{'Friday', 'Tuesday', 'Monday', 'Saturday', 'Thursday', 'Sunday', 'Wednesday'}

<class 'set'>

looping through the set elements ...

Friday

Tuesday

Monday

Saturday

Thursday

Sunday

Wednesday

[3]References : <https://www.javatpoint.com/python-set>

**Python sets :** [4]

|  |  |
| --- | --- |
| myset = {"apple", "banana", "cherry"} | |
| set | thisset = {"apple", "banana", "cherry"} print(thisset)   * Output: {'cherry', 'banana', 'apple'} |
| Python - Access Set Items | thisset = {"apple", "banana", "cherry"}  print("banana" in thisset)   * Output: true |
| Python - Add Set Items | thisset = {"apple", "banana", "cherry"}  tropical = {"pineapple", "mango", "papaya"}  thisset.update(tropical)  print(thisset)   * Output: {'apple', 'mango', 'cherry', 'pineapple', 'banana', 'papaya'} |
| Python - Remove Set Items | thisset = {"apple", "banana", "cherry"}  thisset.remove("banana")  print(thisset)   * Output: {'apple', 'cherry'} |
| Python - Loop Sets | thisset = {"apple", "banana", "cherry"}  for x in thisset:  print(x)   * Output: banana   cherry  apple |
| Python - Join Sets | set1 = {"a", "b" , "c"}  set2 = {1, 2, 3}  set3 = set1.union(set2)  print(set3)   * Output: {3, 'b', 2, 1, 'c', 'a'} |

[4]References: <https://www.w3schools.com/python/python_sets_join.asp>

**Python - Set Methods :** [5]

|  |
| --- |
| **Method Description**  add() Adds an element to the set  clear() Removes all the elements from the set  copy() Returns a copy of the set  difference() Returns a set containing the difference between two or more sets  difference\_update() Removes the items in this set that are also included in another, specified set  discard() Remove the specified item  intersection() Returns a set, that is the intersection of two other sets  intersection\_update() Removes the items in this set that are not present in other, specified set(s)  isdisjoint() Returns whether two sets have a intersection or not  issubset() Returns whether another set contains this set or not  issuperset() Returns whether this set contains another set or not  pop() Removes an element from the set  remove() Removes the specified element  symmetric\_difference() Returns a set with the symmetric differences of two sets  symmetric\_difference\_update() inserts the symmetric differences from this set and another  union() Return a set containing the union of sets  update() Update the set with the union of this set and others |

[5]References: <https://www.w3schools.com/python/python_sets_methods.asp>