

Content Covered

- Set
- Set Methods
- Frozen Set
- Dictionary
- Range

Set

- Multiple data separated by comma & enclosed in {}
- Often used to lookup in bigger data
- We can perform set operations
- Can store only immutable data type
- Doesn't store duplicate
- No concept of positional index.
- Doesn't supporting methods related to indexing

Set Methods

add

- Used to add element to existing set
- Example:
 - `set_a = {1, 4, 3}`
 - `set_a.add(5)`
 - `print(set_a)`

Set Methods

update

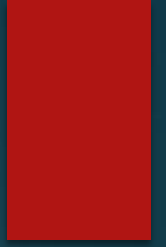
- Add element from iterable to set
- Equivalent to extend in set
- Example:
 - `set_a = {1, 4, 3}`
 - `set_a.update((7, 8))`
 - `print(set_a)`
 - `set_a.update([15, 20])`
 - `print(set_a)`

Set Methods

remove

- remove element from set
- If element is not found it gives key error
- Example:
 - `set_a = {1, 4, 3}`
 - `set_a.remove(3)`
 - `print(set_a)`

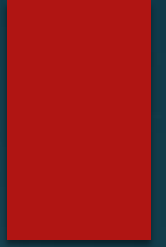
Set Methods



discard

- remove element from set
- Doesn't give error if key not found
- Example:
 - `set_a = {1, 4, 3}`
 - `set_a.discard(5)`
 - `print(set_a)`

Set Methods



containment

- Check if element is present in set
- Example:
 - `set_a = {1, 4, 3}`
 - `print(1 in set_a)`
 - `print(5 in set_a)`

Set Methods

pop

- Removes & return element randomly from the set
- Example:
 - `set_a = {1, 4, 3}`
 - `popped_data = set_a.pop()`
 - `print(popped_data)`

Set Methods

copy

- Like list assignment doesn't copy the set
- `.copy` is used to copy set
- Example:
 - `set_a = {1, 4, 3}`
 - `set_b = set_a.copy()`
 - `set_b.add(7)`
 - `print(set_a)`
 - `print(set_b)`

Set Methods

clear

- Flushes the values in the set
- Example:
 - `set_a = {1, 4, 3}`
 - `print(set_a)`
 - `set_a.clear()`
 - `print(set_a)`

Set Methods

union

- Calculates union of sets
- Example:
 - `set_a = {1, 2, 3}`
 - `set_b = {2, 4, 6}`
 - `union_set = set_a.union(set_b) # a | b`
 - `print(union_set)`

Set Methods

intersection

- Calculates common elements of sets
- Example:
 - `set_a = {1, 2, 3}`
 - `set_b = {2, 4, 6}`
 - `intr_set = set_a.intersection(set_b) # a & b`
 - `print(intr_set)`

Set Methods

difference

- Calculates difference of sets
- Example:
 - `set_a = {1, 2, 3}`
 - `set_b = {2, 4, 6}`
 - `diff = set_a.difference(set_b) # a - b`
 - `print(diff)`

Set Methods

isdisjoint

- Check if sets are disjoint
- Example:
 - `set_a = {1, 2, 3}`
 - `set_b = {2, 4, 6}`
 - `print(set_a.isdisjoint(set_b))`

Set Methods

issubset

- Check if set is subset of another set
- Example:
 - `set_a = {1, 2}`
 - `set_b = {1, 2, 4, 6}`
 - `print(set_a.issubset(set_b))`

Frozen Set

- Can't perform update & remove operation after set is frozen.
- We can perform membership operation or iterate
- Example:
 - `set_a = {1, 2}`
 - `frozen_set = frozenset(set_a)`

Dictionary

- Enclosed inside {} & separated by comma but we have key and value pairs.
- Key must be unique and immutable
- Value can be any Python data type
- Used for lookups

Dictionary

Example:

- `my_dict = { 'name' : 'Rabindra', 'age': 25 }`
 - `my_dict = dict(name='Rabindra', age=25)`
 - `nes_dict = {'d1': {'id': 1, 'nm': 'Doc1'}, 'd2': {'id': 2, 'nm': 'Doc2'}}`
 - `value_list = ['Rabi', 25]`
 - `key_list = ['Name', 'Age']`
 - `key_value_pairs = list(zip(key_list, value_list))`
- # List of tuples of key and values
- `me_dict = dict(key_value_pairs)`

Dictionary Methods

Access and Add element

- Big bracket is used to access / add item
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `print(course_info['name'])`
 - `course_info['name'] = 'Data Science' # Add / Update`
 - `print(course_info)`

Dictionary Methods

get

- Used to find value of given key
- Unlike access, doesn't give error if key not found
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `print(course_info.get('name'))`
 - `print(course_info.get('description'))`
 - `print(course_info.get('difficulty', 'beginner'))`

Dictionary Methods

keys

- Returns list of keys from the dictionary
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `print(course_info.keys())`

Dictionary Methods

values

- Returns list of values from the dictionary
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `print(course_info.values())`

Dictionary Methods

items

- Returns list of tuples with key & values from the dictionary
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `print(course_info.items())`

Dictionary Methods

clear

- Flush / Empty the dictionary
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `print(course_info)`
 - `course_info.clear()`
 - `print(course_info)`

Dictionary Methods

pop

- Removes the value of key from dictionary & returns it.
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `course_name = course_info.pop('name')`
 - `print(course_name)`
 - `print(course_info)`

Dictionary Methods

len

- Calculate number of items in dictionary
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `print(len(course_name))`

Dictionary Methods

copy

- Copies dictionary to another dictionary
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `course_copy = course_info.copy()`
 - `print(course_copy)`

Dictionary Methods

update

- Add items of second dictionary to first
- Second dictionary is unaffected
- If key already exist in first dictionary, value is updated
- Example:
 - `dict_1 = { 'one': 1, 'two': 2 }`
 - `dict_2 = { 'three': 3, 'four': 4 }`
 - `dict_1.update(dict_2)`
 - `print(dict_1)`
 - `print(dict_2)`

Dictionary Methods

containment

- Used to check if key exist in dictionary
- Example:
 - `course_info = { 'name': 'Python', 'month_duration': 3 }`
 - `print('name' in course_info)`
 - `print('Python' in course_info.values())`

Range

- Gives integer in provided range
- Beginning, end and jump can be specified
- End position is exclusive
- Example:
 - `range_a = range(2)`
 - `range_b = range(1, 10)`
 - `range_c = range(2, 11, 2)`

Collection Comparison

property	List	Tuple	Set	Dictionary
ordered	yes	yes	no	no
mutable	yes	no	yes	yes
unique	no	no	yes	yes



Exercise