5/2/24, 1:02 PM about:blank

Cheat Sheet: Python Data Structures Part-2

Dictionaries

Semple S	Package/Method	Description	Code Example
Dictionary A dictionary is a built-in data type that represents calcilation of key-value pair. Dictionaries are inclosed in cuty brace (i. dictionary early 2, person - { "name"; "3ohn", "age"; 36, "city"; "Mew York") 2. person - { "lame; age york"} 2. person - { "coped"} 2. person - { "coped"} 2. person - { "coty of age york"} 3. person - { "coty of age york"} 4.			Example:
Accessing Value Copied Copied Copied Copied Copied	Creating a	a collection of key-value pairs. Dictionaries are	1. 1 2. 2
Accessing Values Accessing Value Accessing Values Accessing Val			<pre>1. dict_name = {} #Creates an empty dictionary 2. person = { "name": "John", "age": 30, "city": "New York"}</pre>
Accessing Values You can access the values in a dictionary using their corresponding keys. 1. 1 1. Value = dict_name[*key_name*]			Copied!
Accessing Values Vou can access the values in a dictionary using the corresponding keys. 1. value = dict_name["key_name"] Copeed			Syntax:
Accessing Values Pound access the values in a dictionary using their corresponding keys. Accessing Values Pound access the values in a dictionary using their corresponding keys. Add or modify Pound access the value will be updated otherwise, a new key-value pair into the dictionary. If the key already exists, the value will be updated otherwise, a new entry is created. Add or modify the key already exists, the value will be updated otherwise, a new entry is created. Add or modify the key already exists, the value will be updated otherwise, a new entry is created. Accepted Syntax: 1. 1. 1. person ["country"] = "USA" # A new entry will be created. 2. person ["city"] = "Thicago" # Update the existing value for the same key copied Syntax: 1. 1. 1. ded id:name[key] Copied! Syntax: 1. 1. 1. ded. person ["country"] Copied! Syntax: 1. 1. 1. dict_name.update({key: value}) Copied! Syntax: 1. 1. 1. dict_name.update({*Profession*: *Doctor**})			
Accessing Values the values in a dictionary using their corresponding keys. Add or modify the key already exist, the value will be updated of the key already exist, the value will be updated of the key already exist, the value will be updated of the key already exist, the value will be updated of the key already exist, the value will be updated of the key already exist, the value will be updated of the key already exist, the value will be updated to therevise, a new entry is created. Add or modify the key already exist, the value will be updated to the key already exist, the value will be updated to the key already exist, the value will be updated to the key already exist, the value will be updated to the key already exist, the value will be updated to the key already exist, the value will be updated to the key already exist, the value will be updated to the same key lead to the key already exist, the value will be updated to the same key lead to the key already exist, the value pair from the dictionary. Raises a Key£rror if the key does not exist. 1. 1			
The cuter() method empties the dictionary, adding or update() method empties the dictionary, adding leaves are unwaignal and several pairs within it. After this cuter() and the cuter() method empties the dictionary, adding or update() and the cuter() method empties the dictionary, adding or update() and the cuter() method empties the dictionary, emonying all key-value pairs within it. After this cuter() and the cuter() method empties the dictionary, emonying all key-value pairs within it. After this cuter() and the cuter() method empties the dictionary, emonying all key-value pairs within it. After this cuter() and the cuter() method empties the dictionary, emonying all key-value pairs within it. After this cuter() copedil cuter() method empties the dictionary, emonying all key-value pairs within it. After this cuter() copedil cuter() method empties the dictionary, emonying all key-value pairs within it. After this cuter() copedil cuter() copedil cuter() copedil cuter() method empties the dictionary, emonying all key-value pairs within it. After this cuter() copedil cuter() copedil cuter() copedil cuter() method empties the dictionary, emonying all key-value pairs within it. After this cuter() copedil cuter() cuter() cuter() cuter() cuter() cuter() cuter() copedil cuter()	Accessing Values		
1. name = person["name"] 2. age = person["age"] Copied Syntax: 1. 1 1. dict_name[key] = value Copied Syntax: 1. 1 1. dict_name[key] = "USA" # A new entry will be created. 1. 1 2. 2 2. person["cuntry"] = "USA" # A new entry will be created. 2. person["cuntry"] = "USA" # A new entry will be created. 2. person["cuntry"] = "USA" # A new entry will be created. 2. person["cuntry"] = "USA" # A new entry will be created. 2. person["cuntry"] = "USA" # A new entry will be created. 3. person["cuntry"] = "USA" # A new entry will be created. 4. person["cuntry"] = "USA" # A new entry will be created. 5. person["cuntry"] = "USA" # A new entry will be created. 6. person["cuntry"] = "USA" # A new entry will be created. 7. person["cuntry"] = "USA" # A new entry will be created. 8. person["cuntry"] = "USA" # A new entry will be created. 9. person["cuntry"] = "USA" # A new entry will be created. 1. 1	J		1. 1
Copied! Syntax: 1.1 1. dict_name[key] = value Copied! Syntax: 1.2 1.2 Copied! Syntax: 1.3 1.4 Copied! Syntax: 1.4 1.5 Copied! Syntax: 1.5 1.5 Copied! Syntax:			<pre>1. name = person["name"]</pre>
Add or modify Inserts a new key-value pair into the dictionary. If the key already exists, the value will be updated; otherwise, a new entry is created. Inserts a new key-value pair into the dictionary. If the key already exists, the value will be updated; otherwise, a new entry is created. Inserts a new key-value pair into the dictionary. If the key already exists, the value will be updated; and the will be created. In person ["Country"] = "USA" # A new entry will be created. In person ["Country"] = "Chicago" # Update the existing value for the same key Copiedl Syntax: In ded dict_name[key] Copiedl Syntax: In ded dict_name[key] Copiedl Syntax: In ded person ["Country"] Copiedl Syntax: In ded intername. update({key: value}) Copiedl Syntax: In ded intername. update({key: value}) Copiedl Syntax: In the update() method merges the provided dictionary, adding or updating key-value pairs. In the update() method empties the dictionary, adding or updating key-value pairs. In the update() method empties the dictionary, adding or updating key-value pairs. In the update() method empties the dictionary, adding or updating key-value pairs within it. After this person. update({"Profession": "Doctor"}) Copiedl Syntax: In the update() method empties the dictionary, adding or update({"Profession": "Doctor"}) Copiedl Syntax:			
Add or modify Inserts a new key-value pair into the dictionary, if the key already exists, the value will be updated; otherwise, a new entry is created. 1. 1			
Add or modify in the key already exists, the value will be updated; otherwise, a new entry is created. Copied Example:			1. 1
Add or modify Inserts a new key-value pair into the dictionary, if the key already exists, the value will be updated; otherwise, a new entry is created.			<pre>1. dict_name[key] = value</pre>
Add or modify otherwise, a new entry is created. 1. 1		T	Copied!
1. 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2	Add or modify	the key already exists, the value will be updated;	Example:
1. person["Country"] = "USA" # A new entry will be created. 2. person["city"] = "Chicago" # Update the existing value for the same key Copied Syntax: 1. 1			
Copied! Syntax: 1. 1 1. del dict_name[key] Copied! Example:			 person["Country"] = "USA" # A new entry will be created.
Removes the specified key-value pair from the dictionary, Raises a KeyError if the key does not exist. Removes the specified key-value pair from the dictionary, Raises a KeyError if the key does not exist. Copied! Example: 1. 1 1. del person["Country"] Copied! Syntax: 1. 1 1. dict_name.update({key: value}) Copied! Example: 1. 1 1. dict_name.update({key: value}) Copied! Example: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Syntax: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Syntax:			
1. del dict_name[key] Copiedl Copiedl Example:			Syntax:
Removes the specified key-value pair from the dictionary. Raises a KeyError if the key does not exist. Example: 1. 1 1. del person["Country"] Copied! Syntax: 1. 1 1. dict_name.update({key: value}) Copied! Syntax: 1. 1 1. dict_name.update({key: value}) Copied! Syntax: 1. 1 1. dict_name.update({key: value}) Copied! Lample: 1. 1 1. dict_name.update({key: value}) Copied! In the update() method merges the provided dictionary, adding or updating key-value pairs. Copied! Example: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Syntax: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Syntax: Copied! Copied! Syntax: Copied!			1. 1
del dictionary. Raises a KeyError if the key does not exist. Example: 1. 1 1. del person["Country"] Copied! Syntax: 1. 1 1. dict_name.update({key: value}) Copied! Example: 1. 1 1. dict_name.update({key: value}) Copied! Example: 1. 1 1. dict_name.update({key: value}) Copied! Example: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Syntax: Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this Copied! Syntax:			<pre>1. del dict_name[key]</pre>
exist. Example: 1. 1 1. del person["Country"] Copied! Syntax: 1. 1 1. dict_name.update({key: value}) Copied! Line dict_name.update({key: value}) Copied! Example: 1. 1 1. dict_name.update({key: value}) Copied! Example: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this	dol	dictionary. Raises a KeyError if the key does not	Copied!
1. del person["Country"] Copied! Syntax: 1. 1 1. dict_name.update({key: value}) The update() method merges the provided dictionary into the existing dictionary, adding or updating key-value pairs. Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this The clear() method empties the dictionary, removing all key-value pairs within it. After this	uci		Example:
Copied! Syntax: 1. 1 1. dict_name.update({key: value}) Copied! Copied! Copied! Copied! Lample: Update() method merges the provided dictionary into the existing dictionary, adding or updating key-value pairs. Copied! Example: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Copied! Syntax:			1. 1
Syntax: 1. 1 1. dict_name.update({key: value}) The update() method merges the provided dictionary into the existing dictionary, adding or updating key-value pairs. Copied! Example: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this			1. del person["Country"]
1. 1 1. dict_name.update({key: value}) The update() method merges the provided dictionary into the existing dictionary, adding or updating key-value pairs. Example: 1. 1 1. dict_name.update({key: value}) Copied! Example: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this			
The update() method merges the provided dictionary into the existing dictionary, adding or updating key-value pairs. 1. dict_name.update({key: value}) Copied! Example: 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this			
The update() method merges the provided dictionary into the existing dictionary, adding or updating key-value pairs. 1. 1 1. person.update({"Profession": "Doctor"}) Copied! Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this			
update() dictionary into the existing dictionary, adding or updating key-value pairs. 1. 1 1. person.update({"Profession": "Doctor"}) Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this	update()	dictionary into the existing dictionary, adding or	
1. 1 1. person.update({"Profession": "Doctor"}) Clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this			
1. person.update({"Profession": "Doctor"}) Copied! clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this			-
clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this			
clear() The clear() method empties the dictionary, removing all key-value pairs within it. After this	clear()	removing all key-value pairs within it. After this	
			1. 1
Copied!			Copied!
Example:			Example:
1. 1			1. 1
<pre>1. grades.clear()</pre>			1. grades.clear()

```
Copied!
                                                                     Example:
                                                                       1. 1
                   You can check for the existence of a key in a
key existence

    if "name" in person:
    print("Name exists in the dictionary.")

                   dictionary using the in keyword
                                                                     Copied!
                                                                     Syntax:
                                                                       1. 1
                                                                       1. new_dict = dict_name.copy()
                                                                      Copied!
                   Creates a shallow copy of the dictionary. The new
                   dictionary contains the same key-value pairs as
copy()
                                                                     Example:
                   the original, but they remain distinct objects in
                   memory.

    new_person = person.copy()
    new_person = dict(person) # another way to create a copy of dictionary

                                                                     Copied!
                                                                     Syntax:
                                                                       1. keys_list = list(dict_name.keys())
                                                                     Copied!
                   Retrieves all keys from the dictionary and
keys()
                   converts them into a list. Useful for iterating or
                   processing keys using list methods.
                                                                     Example:
                                                                       1. 1
                                                                       1. person_keys = list(person.keys())
                                                                     Copied!
                                                                     Syntax:
                                                                       1. 1
                                                                       1. values_list = list(dict_name.values())
                   Extracts all values from the dictionary and
                                                                     Copied!
values()
                   converts them into a list. This list can be used for
                   further processing or analysis.
                                                                     Example:
                                                                       1. 1
                                                                       1. person_values = list(person.values())
                                                                     Copied!
                                                                     Syntax:
                                                                       1. 1
                                                                       1. items_list = list(dict_name.items())
                   Retrieves all key-value pairs as tuples and
                                                                     Copied!
items()
                   converts them into a list of tuples. Each tuple
                                                                     Example:
                   consists of a key and its corresponding value.
                                                                       1. 1
                                                                       1. info = list(person.items())
                                                                     Copied!
```

Sets

Package/Metho	d Description	Code Example
		Syntax:
		1. 1
add()	Elements can be added to a set using the `add()` method. Duplicates are automatically removed, as sets only store unique values.	<pre>1. set_name.add(element)</pre>
		Copied!
		Example:
		1. 1
		 fruits.add("mango")
clear()		Copied!
	The `clear()` method removes all elements from the set, resulting in an empty set. It updates the set in-place.	. Syntax:
		1. 1
		<pre>1. set_name.clear()</pre>

5/2/24, 1:02 PM about:blank

```
Copied!
                                                                                                   Example:
                                                                                                      1. 1
                                                                                                      1. fruits.clear()
                                                                                                    Copied!
                                                                                                   Syntax:
                                                                                                      1. 1
                                                                                                      1. new_set = set_name.copy()
                                                                                                   Copied!
                  The `copy()` method creates a shallow copy of the set. Any modifications to the
copy()
                  copy won't affect the original set.
                                                                                                   Example:
                                                                                                      1. 1
                                                                                                      1. new_fruits = fruits.copy()
                                                                                                    Copied!
                                                                                                   Example:
                                                                                                      1. 1
2. 2
                  A set is an unordered collection of unique elements. Sets are enclosed in curly
Defining Sets
                  braces `{}`. They are useful for storing distinct values and performing set
                                                                                                      1. empty_set = set() #Creating an Empty
2. Set fruits = {"apple", "banana", "orange"}
                  operations.
                                                                                                    Copied!
                                                                                                   Syntax:
                                                                                                      1. 1

    set_name.discard(element)

                                                                                                    Copied!
                  Use the 'discard()' method to remove a specific element from the set. Ignores if
discard()
                  the element is not found.
                                                                                                   Example:
                                                                                                      1. 1

    fruits.discard("apple")

                                                                                                    Copied!
                                                                                                   Syntax:
                                                                                                      1. 1
                                                                                                      1. is subset = set1.issubset(set2)
                                                                                                    Copied!
                  The `issubset()` method checks if the current set is a subset of another set. It
issubset()
                  returns True i\ddot{f} all elements of the current set are present in the other set,
                                                                                                   Example:
                  otherwise False.
                                                                                                      1. is_subset = fruits.issubset(colors)
                                                                                                    Copied!
                                                                                                   Syntax:
                                                                                                   is_superset = set1.issuperset(set2)
                  The `issuperset()` method checks if the current set is a superset of another set. It
                                                                                                   Example:
                  returns True if all elements of the other set are present in the current set,
issuperset()
                  otherwise False.
                                                                                                      1. 1
                                                                                                      1. is_superset = colors.issuperset(fruits)
                                                                                                    Copied!
                                                                                                   Syntax:
                                                                                                      1. removed_element = set_name.pop()
                                                                                                    Copied!
                  The 'pop()' method removes and returns an arbitrary element from the set. It
pop()
                  raises a 'KeyError' if the set is empty. Use this method to remove elements when
                                                                                                   Example:
                  the order doesn't matter.
                                                                                                      1. removed_fruit = fruits.pop()
                                                                                                   Copied!
remove()
                  Use the 'remove()' method to remove a specific element from the set. Raises a
                                                                                                   Syntax:
                  `KeyError` if the element is not found.
                                                                                                      1. 1

    set_name.remove(element)

                                                                                                    Copied!
```

5/2/24, 1:02 PM about:blank

Perform various operations on sets: `union`, `intersection`, `difference`,

The 'update()' method adds elements from another iterable into the set. It

Example:

fruits.remove("banana")

Copied!

Syntax:

1. 1 2. 2 3. 3 4. 4

1. union set = set1.union(set2)

3. difference_set = set1.intersection(set2)
3. difference_set = set1.difference(set2)
4. sym_diff_set = set1.symmetric_difference(set2)

Copied!

Example:

1. 1 2. 2 3. 3 4. 4

1. combined = fruits.union(colors)

common = fruits.intersection(colors)
 unique_to_fruits = fruits.difference(colors)
 sym_diff = fruits.symmetric_difference(colors)

Copied!

Syntax:

set_name.update(iterable)

Copied!

Example:

1. 1

1. fruits.update(["kiwi", "grape"])

Copied!

maintains the uniqueness of elements.

`symmetric difference`.



© IBM Corporation. All rights reserved.

Set Operations

update()

about:blank