



EXPLORER

OPEN EDITORS

percipio17\_dict\_type.py P...

PYTHON

Automate-Boring-Stuff

my\_code

Percipio\_Python3-Course

01\_Start

02\_Data-Sequence Types

03\_Collections-Mapping-Lo...

percipio15\_range\_type\_a...

percipio16\_set\_type.py

percipio17\_dict\_type.py

percipio18\_while\_loop.py

percipio19\_forloop.py

percipio20\_if\_statement....

percipio21\_exercise\_nam...

04\_Modules-Functions

05\_Classes

06\_Working-with-Files

07\_Comprehensions

08\_Iterables-and-Generators

09\_Exceptions

Python Projects\_2014

CMD\_Python\_Set-Path.txt

Python\_Basics.txt

Python\_Clear-Window-Comm..

python\_exercises\_00.py

python\_exercises\_01.py

Python\_Tutorial\_Running-Scr...

Python\_Tutorials.md

percipio17\_dict\_type.py x

```
1  # percipio17_dict_type.py
2  # Percipio video: Collections, Mapping, & Looping; The dict type in Python (Note: dict=dictionary)
3  # "One of the workhorses of python"
4  # The dictionary class allows creating an dict-object which is basically mapping or an associative array of
   keys and associated values.
5  # Dict-Keys must be unique, immutable objects, similar to numbers and strings.
6  # Dictionaries are not ordered objects but there is an ordered-dict class available
7  nl = '\n'
8
9  print(nl, 'Working with Dictionaries')
10 empty_dict = dict() # Use the dict-function to create a dict-object
11 print('empty_dict = dict() ->', empty_dict) # a dict-object is respresented as a pair of curly braces
12 empty_dict = {} # a simpler way to create an empty dict-object
13 print('empty_dict = dict{} ->', empty_dict) #
14 dict_syn = {'k1' : 'v1', 'k2' : 'v2'} # typical way to write a dictionary object with comma-seperated key &
   value pairs
15 # curly-brace key-colon-associatedValue comma key-colon-associatedValue comma and-so-on curly-brace (all in
   individual quotes) {'k1' : 'v1', 'k2' : 'v2'}
16 print('dict_syn ->', dict_syn) #
17 dict_syn = dict(k1='v1', k2='v2') # use the dict-function as another way to write a dictionary object with
   comma-seperated key & value pairs
18 print('dict_syn ->', dict_syn) #
19
20 print(nl, 'Working with Keys & Values')
21 print("dict_syn['k2'] ->", dict_syn['k2']) # access the value of one-specific-key in a dict-object
22 dict_syn['k3'] = 'v3' # add a new key & value to an existing dictionary
23 print('dict_syn ->', dict_syn) #
24 dict_syn['k3'] = 'v6' # change the value of an existing key in an existing dictionary
25 print('dict_syn ->', dict_syn) #
26 del(dict_syn['k3']) # delete an element with the key & value
27 print('dict_syn ->', dict_syn) #
28 dict_syn['k1']=1 # change the value of an existing key in an existing dictionary
29 print('dict_syn ->', dict_syn) #
30 dict_syn['k2']=1 # change the value of an existing key showing where keys must be unique but values do not
   need to be
31 print('dict_syn ->', dict_syn) #
32
33 print(nl, 'Clear a dict, create a copy of a dict')
34 dict_ref = dict_syn # create a variable-reference to the original dict_syn object
35 dict_copy = dict_syn.copy() # create a variable-reference to a newly created copy of the dict_syn object
   using the copy-method of the dict-object
36 dict_syn.clear() # clears the original dict_syn object, leaving the copy dict_syn.copy() unchanged
37 print('dict_syn ->', dict_syn) # dict_syn shown cleared
38 print('dict_ref ->', dict_ref) # dict_syn shown cleared
39 print('dict_copy ->', dict_copy) # dict_copy still has all elements, keys, & values
40 print('dict_syn.copy (curiosity-test) ->', dict_syn.copy()) # DOES THIS PRINT OUT? Result = <built-in method
   copy of dict object at 0x000001E1610961F8>
41
```





EXPLORER

OPEN EDITORS

- percipio17\_dict\_type.py P...

PYTHON

- Automate-Boring-Stuff
- my\_code
- Percipio\_Python3-Course
  - 01\_Start
  - 02\_Data-Sequence Types
  - 03\_Collections-Mapping-Lo...
    - percipio15\_range\_type\_a...
    - percipio16\_set\_type.py
    - percipio17\_dict\_type.py
    - percipio18\_while\_loop.py
    - percipio19\_forloop.py
    - percipio20\_if\_statement....
    - percipio21\_exercise\_nam...
  - 04\_Modules-Functions
  - 05\_Classes
  - 06\_Working-with-Files
  - 07\_Comprehensions
  - 08\_Iterables-and-Generators
  - 09\_Exceptions
- Python Projects\_2014
- CMD\_Python\_Set-Path.txt
- Python\_Basics.txt
- Python\_Clear-Window-Comm..
- python\_exercises\_00.py
- python\_exercises\_01.py
- Python\_Tutorial\_Running-Scr...
- Python\_Tutorials.md

percipio17\_dict\_type.py

```
40 print('dict_syn.copy (curiosity-test) ->', dict_syn.copy) # DOES THIS PRINT OUT? Result = <built-in method
   copy of dict object at 0x000001E1610961F8>
41
42 print(nl, 'Get a list of key &/or a list of values')
43 key_list = dict_copy.keys() # display a list of all keys
44 print('key_list ->', key_list) # display a list of all values
45 value_list = dict_copy.values() #
46 print('value_list ->', value_list) #
47
48 print(nl, 'Create a new dictionary using existing keys & values')
49 mapping = zip(key_list, value_list) # create a mapping-object by creating a zip-function and passing in a
   key-list & a value-list
50 dict_new = dict(mapping) # Creates a new dictionary using the dict-function passing in the mapping object
   (zip object) as the argument
51 print('mapping ->', mapping) # shows the mapping and zip objects
52 print('dict_new ->', dict_new) #
53
54 print(nl, 'Dictionary inspection functions')
55 print('Test if keys are in or not-in a dict')
56 print("'k3' in dict_new ->", 'k3' in dict_new) # test for key in dict which is not returns False
57 print("'k3' not in dict_new ->", 'k3' not in dict_new) # test for a key not in dict which is not returns True
58 print("'k2' in dict_new ->", 'k2' in dict_new) # test for a key in dict which is returns True
59 print("'k2' not in dict_new ->", 'k2' not in dict_new) # test for a key not in dict which is returns False
60 ...
61 Working with Dictionaries
62 empty_dict = dict() -> {}
63 empty_dict = dict{} -> {}
64 dict_syn -> {'k1': 'v1', 'k2': 'v2'}
65 dict_syn -> {'k1': 'v1', 'k2': 'v2'}
66
67 Working with Keys & Values
68 dict_syn['k2'] -> v2
69 dict_syn -> {'k1': 'v1', 'k2': 'v2', 'k3': 'v3'}
70 dict_syn -> {'k1': 'v1', 'k2': 'v2', 'k3': 'v6'}
71 dict_syn -> {'k1': 'v1', 'k2': 'v2'}
72 dict_syn -> {'k1': 1, 'k2': 'v2'}
73 dict_syn -> {'k1': 1, 'k2': 1}
74
75 Clear a dict, create a copy of a dict
76 dict_syn -> {}
77 dict_ref -> {}
78 dict_copy -> {'k1': 1, 'k2': 1}
79 dict_syn.copy (curiosity-test) -> <built-in method copy of dict object at 0x000002605F295240>
80
81 Get a list of key &/or a list of values
82 key_list -> dict_keys(['k1', 'k2'])
83 value_list -> dict_values([1, 1])
84
```

```
dict_syn = {'k1': 'v1', 'k2': 'v2'}
dict_syn['k3'] = 'v3'
dict_syn['k3'] = 'v6'
dict_syn['k2'] = 1
dict_syn['k2'] = 1
dict_syn.clear()
dict_ref = dict_syn
dict_copy = dict_syn
dict_syn.copy()
dict_syn.keys()
dict_syn.values()
```





EXPLORER

## OPEN EDITORS

percipio17\_dict\_type.py P..

## PYTHON

Automate-Boring-Stuff

my\_code

Percipio\_Python3-Course

01\_Start

02\_Data-Sequence Types

03\_Collections-Mapping-Lo..

percipio15\_range\_type\_a...

percipio16\_set\_type.py

percipio17\_dict\_type.py

percipio18\_while\_loop.py

percipio19\_forloop.py

percipio20\_if\_statement....

percipio21\_exercise\_nam...

04\_Modules-Functions

05\_Classes

06\_Working-with-Files

07\_Comprehensions

08\_Iterables-and-Generators

09\_Exceptions

Python Projects\_2014

CMD\_Python\_Set-Path.txt

Python\_Basics.txt

Python\_Clear-Window-Comm..

python\_exercises\_00.py

python\_exercises\_01.py

Python\_Tutorial\_Running-Scr...

Python\_Tutorials.md

percipio17\_dict\_type.py x

```
'''
60
61 Working with Dictionaries
62 empty_dict = dict() -> {}
63 empty_dict = dict{} -> {}
64 dict_syn -> {'k1': 'v1', 'k2': 'v2'}
65 dict_syn -> {'k1': 'v1', 'k2': 'v2'}
66
67 Working with Keys & Values
68 dict_syn['k2'] -> v2
69 dict_syn -> {'k1': 'v1', 'k2': 'v2', 'k3': 'v3'}
70 dict_syn -> {'k1': 'v1', 'k2': 'v2', 'k3': 'v6'}
71 dict_syn -> {'k1': 'v1', 'k2': 'v2'}
72 dict_syn -> {'k1': 1, 'k2': 'v2'}
73 dict_syn -> {'k1': 1, 'k2': 1}
74
75 Clear a dict, create a copy of a dict
76 dict_syn -> {}
77 dict_ref -> {}
78 dict_copy -> {'k1': 1, 'k2': 1}
79 dict_syn.copy (curiosity-test) -> <built-in method copy of dict object at 0x000002605F295240>
80
81 Get a list of key &/or a list of values
82 key_list -> dict_keys(['k1', 'k2'])
83 value_list -> dict_values([1, 1])
84
85 Create a new dictionary using existing keys & values
86 mapping -> <zip object at 0x000002605F296488>
87 dict_new -> {'k1': 1, 'k2': 1}
88
89 Dictionary inspection functions
90 Test if keys are in or not-in a dict
91 'k3' in dict_new -> False
92 'k3' not in dict_new -> True
93 'k2' in dict_new -> True
94 'k2' not in dict_new -> False
95 '''
```