



EXPLORER

OPEN EDITORS

- percipio13_tuple_type...

PYTHON

- Automate-Boring-Stuff
- my_code
- Percipio_Python3-Course
 - 01_Start
 - 02_Data-Sequence Types
 - percipio04_int_types.py
 - percipio05_float_type....
 - percipio06_math_func...
 - percipio07_boolean_t...
 - percipio08_Strings.py
 - percipio09_float_type....
 - percipio10_bytes_type..
 - percipio11_bytearray_...
 - percipio12_list_type.py
 - percipio13_tuple_type...
 - percipio14_slice_type....
 - percipio14a_list_copy_...
 - 03_Collections-Mapping..
 - 04_Modules-Functions
 - 05_Classes
 - 06_Working-with-Files
 - 07_Comprehensions
 - 08_Iterables-and-Gener...
 - 09_Exceptions
 - 10_Automation Program..
- Python Projects_2014
- CMD_Python_Set-Path.txt
- Python_Clear-Window-Co...
- python_debug_logging_co...
- python_exercises_00.py
- python_exercises_01.py
- Python_Tutorial_Running-...
- Python_Tutorials.md
- start_code_for_python_ma...

percipio13_tuple_type.py x

```
1  '''
2  percipio13_tuple_type.py
3  Percipio video: Data & Sequence Types; The tuple Type in Python
4
5  * The tuple class provides an immutable sequence of elements (immutable = unchanging)
6  * Tuple class is similar to list class but have no methods to change the tuple elements
7  '''
8  nl = '\n'
9  empty_tuple = tuple() # creates an empty tuple
10 print('empty_tuple ->', empty_tuple)
11 print('type(empty_tuple) ->', type(empty_tuple))
12 tuple_str = tuple('hello') # converts sequences of characters in a string into individual elements
   in a tuple
13 print('tuple_str ->', tuple_str)
14 tuple_list = tuple([1, 2, [3, 5, 7]]) # convert a list into a tuple; Notice the inner list remains
   intact as one element of the tuple
15 print('tuple_list ->', tuple_list)
16 empty_tuple = () # creates an empty tuple similar in result to above line, 'empty_tuple = tuple()'
17 print('empty_tuple ->', empty_tuple)
18 print(nl)
19 # Creating a tuple with 1 element requires a comma.
20 # An single element inside a parathesis uses the parathesis to evaluate only that 1 object.
21 # Need the parathesis, a comma, & 1 element to create a single-element-tuple which is called a
   singleton.
22 single_object_tuple = (1) #
23 print('single_object_tuple = (1) ->', single_object_tuple)
24 singleton_tuple = (1,) #
25 print('singleton_tuple = (1,) ->', singleton_tuple)
26 print(nl, 'Next')
27 tuple_syn = (3, 4, 'a', 'b') #
28 print('tuple_syn = (3, 4, "a", "b") ->', tuple_syn)
29 print("'a' in tuple_syn ->", 'a' in tuple_syn) # boolean test for tuple membership with the 'in'
   operator
30 print("1 in tuple_syn ->", 1 in tuple_syn)
31 print("1 not in tuple_syn ->", 1 not in tuple_syn) # boolean test for tuple membership with the
   'not in' operator
32 print(nl, 'Next')
```




- EXPLORER
- percipio13_tuple_type.py x
- OPEN EDITORS
- percipio13_tuple_type...
- PYTHON
- Automate-Boring-Stuff
 - my_code
 - Percipio_Python3-Course
 - 01_Start
 - 02_Data-Sequence Types
 - percipio04_int_types.py
 - percipio05_float_type....
 - percipio06_math_func...
 - percipio07_boolean_t...
 - percipio08_Strings.py
 - percipio09_float_type....
 - percipio10_bytes_type..
 - percipio11_bytearray_...
 - percipio12_list_type.py
 - percipio13_tuple_type...
 - percipio14_slice_type....
 - percipio14a_list_copy_...
 - 03_Collections-Mapping..
 - 04_Modules-Functions
 - 05_Classes
 - 06_Working-with-Files
 - 07_Comprehensions
 - 08_Iterables-and-Gener...
 - 09_Exceptions
 - 10_Automation Program..
 - Python Projects_2014
 - CMD_Python_Set-Path.txt
 - Python_Clear-Window-Co...
 - python_debug_logging_co..
 - python_exercises_00.py
 - python_exercises_01.py
 - Python_Tutorial_Running-...
 - Python_Tutorials.md
 - start_code_for_python_ma...

```
32 print(nl, 'Next')
33 print('tuple_str ->', tuple_str)
34 print('min(tuple_str) ->', min(tuple_str)) # finds the minimum value which is lowest in
    alphabetical order
35 print('max(tuple_str) ->', max(tuple_str)) # finds the minimum value which is largest in
    alphabetical order
36 print('sorted(tuple_str) ->', sorted(tuple_str)) # creates a COPY of the original tuple sorting a
    new list alphabetically
37 print('tuple_str ->', tuple_str)
38 print('tuple_str.count("o") ->', tuple_str.count("o")) # how many elements are within a list
39 print('tuple_str.index("o") ->', tuple_str.index("o")) # which position within a list an element
    appears (if not element, than error)
40 print('len(tuple_str) ->', len(tuple_str)) # displays total number sof elements within the list
41 '''
42 RESULT:
43 empty_tuple -> ()
44 type(empty_tuple) -> <class 'tuple'>
45 tuple_str -> ('h', 'e', 'l', 'l', 'o')
46 tuple_list -> (1, 2, [3, 5, 7])
47 empty_tuple -> ()
48
49 single_object_tuple = (1) -> 1
50 singleton_tuple = (1,) -> (1,)
51
52 Next
53 tuple_syn = (3, 4, "a", "b") -> (3, 4, 'a', 'b')
54 'a' in tuple_syn -> True
55 1 in tuple_syn -> False
56 1 not in tuple_syn -> True
57
58 Next
59 tuple_str -> ('h', 'e', 'l', 'l', 'o')
60 min(tuple_str) -> e
61 max(tuple_str) -> o
62 sorted(tuple_str) -> ['e', 'h', 'l', 'l', 'o']
63 tuple_str -> ('h', 'e', 'l', 'l', 'o')
64 tuple_str.count("o") -> 1
65 tuple_str.index("o") -> 4
66 len(tuple_str) -> 5
67 '''
```

