percipio16_set_type.py X **EXPLORER A OPEN EDITORS** # Percipio video: Collections, Mapping, & Looping; The Set type in Python 🕏 percipio16_set_type.py Pe... # Set class provides mapping of unique immutable elements **▲ PYTHON** $n1 = ' \ n'$ ▶ Automate-Boring-Stuff ▶ my_code empty set = set() # use the set function to create an empty set ▲ Percipio_Python3-Course print('empty set ->', empty set) (%) alpha = set(('a','b','c','d')) # use a set-constructor-function on a tuple-sequence creating a set with ▶ 01 Start curly-braces around an unordered list ▶ 02_Data-Sequence Types print('alpha ->', alpha, '(use a set-constructor-function on a tuple-sequence creating a set with ■ 03_Collections-Mapping-Lo.. curly-braces around an unordered list)') percipio15_range_type_a... percipio16_set_type.py print(nl) percipio17_dict_type.py # a useful feature of sets is to eliminate duplicates dup list = ['c','d','c','d', 'e','f'] # a list with duplicates percipio18_while_loop.py print('dup_list ->', dup_list, '(a list with duplicates)') percipio19_forloop.py beta = set(dup list) # This set-beta will have no duplicates in it percipio20_if_statement.... print('beta ->', beta, '(The dup list-set with duplicates removed)') percipio21_exercise_nam... uniq list = list(beta) # Now convert that list without duplicates back into a list with unique set of elements ▶ 04_Modules-Functions print('uniq list ->', uniq list, '(The dup list set with duplicates removed recreated as a list)') # all ▶ 05 Classes duplicates removed from dup list ▶ 06_Working-with-Files print(nl) print('Set Operation Methods:' + nl + 'Union combines one set with another set also removing any duplicates' ▶ 07_Comprehensions + nl + 'Intersection finds where 2 sets have overlapping elements' + nl + 'Difference compares 2 sets ▶ 08 Iterables-and-Generators removing duplicate elements from the set in parenthesis' + nl + 'Symmetric difference is the Union method ▶ 09_Exceptions minus the intersection method') Python Projects_2014 print(nl) # Set Operations ■ Python_Basics.txt print(nl, 'Set Operations below use these 2 sets, alpha & beta') print('alpha ->', alpha) ■ Python_Clear-Window-Comm... print('beta ->', beta) python_exercises_00.py print(nl) python_exercises_01.py # perform a combination of 2 sets through a union Python_Tutorial_Running-Scr... print(nl, 'alpha.union(beta)') Python_Tutorials.md gamma = alpha.union(beta) # union combines one set with another set also removing any duplicates (??WHY DOES IT REMOVE DUPS??) print('union or | ->', gamma, '(union combines 2 set-methods removing any duplicates)') gamma = alpha | beta # anthor way to do a union print('| or union ->', gamma, '("|" is another way to write union)') print(nl, 'alpha.intersection(beta)') delta = alpha.intersection(beta) # find where 2 sets overlap or intersect print('intersection or & ->', delta, '(intersection finds where 2 sets have overlapping elements)') delta = alpha & beta # another way to do a delta print('& or intersection ->', delta, '("&" is another way to write intersection)') print(nl, 'alpha.difference(beta)')

ensilon - alpha difference(heta) # Do a set difference using the difference method removing duplicate

Ln 29, Col 82 Spaces: 4 UTF-8 CRLF Python 🙂

Ⅲ …

percipio16_set_type.py X **EXPLORER** print('& or intersection ->', delta, '("&" is another way to write intersection)') **A OPEN EDITORS** percipio16_set_type.py Pe... print(nl, 'alpha.difference(beta)') **▲ PYTHON** epsilon = alpha.difference(beta) # Do a set-difference using the difference method removing duplicate ▶ Automate-Boring-Stuff 89 elements from the set in parenthesis ▶ my_code print('difference or - ->', epsilon, '(difference compares 2 sets removing duplicate elements from the set in ▲ Percipio_Python3-Course parenthesis)') (%) epsilon = alpha - beta # another way to do a epsilon ▶ 01 Start print('- or difference ->', epsilon, '("-" is another way to write difference)') ▶ 02_Data-Sequence Types ■ 03_Collections-Mapping-Lo.. print(nl, 'alpha.symmetric difference(beta)') percipio15_range_type_a... eta = alpha.symmetric difference(beta) # Union method minus the intersection method checking for overlapping percipio16_set_type.py percipio17_dict_type.py print('symmetric difference or ^ ->', eta, '(symmetric difference is the Union method minus the intersection percipio18_while_loop.py method)') eta = alpha ^ beta # another way to do a eta percipio19_forloop.py print('^ or symmetric difference ->', eta, '("^" is another way to write symmetric difference)') percipio20_if_statement.... percipio21_exercise_nam... ▶ 04_Modules-Functions print(nl, 'Set Comparisons') ▶ 05 Classes # Set Comparisons ▶ 06_Working-with-Files # isdisjoint - tests two sets for any shared elements ▶ 07_Comprehensions ▶ 08 Iterables-and-Generators print('- isdisjoint - tests two sets for any shared elements; boolean True with no elements in common' + nl + ▶ 09 Exceptions '- issubset - tests if all elements of set(left) are within all elements of set(right), boolean True when Python Projects_2014 subset exists' + nl + '- issuperset - tests if all elements of set(right) are within all elements of set (left), boolean True when subset exists') print(nl, '') ■ Python_Basics.txt print('epsilon.isdisjoint(delta) ->', epsilon.isdisjoint(delta), '()') # ■ Python_Clear-Window-Comm... print('epsilon.isdisjoint(eta) ->', epsilon.isdisjoint(eta), '()') # boolean True with no elements in common python_exercises_00.py print(nl, '') python_exercises_01.py print('epsilon.issubset(eta) ->', epsilon.issubset(eta), '()') # Python_Tutorial_Running-Scr... print('epsilon.issubset(beta) ->', epsilon.issubset(beta), '()') # boolean True when subset exists Python_Tutorials.md print(nl, '') print('eta.issuperset(epsilon) ->', eta.issuperset(epsilon), '()') # print('beta.issuperset(epsilon) ->', beta.issuperset(epsilon), '()') # boolean True when superset exists print(nl, '') feta = frozenset(eta) # frozensets are immutable (unchanging), set operations are allowd such as comparisons, but no modifying nor updating. print('feta ->', feta, '()') # frozenset({'a', 'b', 'e', 'f'}) () # Below applies to sets (mutable), not frozensets (immutable) print(nl, 'zeta') zeta = set() # nrint('zeta - set() ->' zeta '()') #

田 …

percipio16_set_type.py X **EXPLORER A OPEN EDITORS** zeta = set() # print('zeta = set() ->', zeta, '()') # 🕏 percipio16_set_type.py Pe... zeta.add(3) # **▲ PYTHON** print('zeta.add(3) ->', zeta, '()') # ▶ Automate-Boring-Stuff 89 zeta.add(3) # ▶ my_code print('zeta.add(3) ->', zeta, '()') # adding an element which already exists in a set will be ignored, no ▲ Percipio_Python3-Course error (%) ▶ 01 Start zeta.add(4) # print('zeta.add(4) ->', zeta, '()') # {3, 4} () ▶ 02_Data-Sequence Types ■ 03_Collections-Mapping-Lo.. print(nl, 'gamma') percipio15_range_type_a... print('gamma ->', gamma, '()') # {'e', 'f', 'a', 'c', 'b', 'd'} () percipio16_set_type.py gamma.discard('a') # percipio17_dict_type.py print('gamma.discard("a") ->', gamma, '()') # will remove the element if present in the set percipio18_while_loop.py gamma.discard('z') # removes an element within a set, if element is not in set, results will be ignored without an error percipio19_forloop.py print('gamma.discard("z") ->', gamma, '()') # percipio20_if_statement.... gamma.remove('b') # removes an element within a set, if element is not in set, results in an ERROR percipio21_exercise_nam... print('gamma.remove("b") ->', gamma, '()') # ▶ 04_Modules-Functions random element = gamma.pop() # pop method removes a random element ▶ 05 Classes print('random element returned ->', random element, '(pop method removes a random element)') ▶ 06_Working-with-Files print('gamma ->', gamma, '()') # {'f', 'c', 'd'} () ▶ 07_Comprehensions print(nl, 'zeta') 08 Iterables-and-Generators zeta ref = zeta # creates a variable that references an existing set. If set if changed, reference is ▶ 09_Exceptions Python Projects_2014 zeta copy = zeta.copy() # copies a set creating a duplicate zeta.clear() # CAUTION: removes ALL elements from a set ■ Python_Basics.txt print('zeta ->', zeta, '()') # shows set is now clear ■ Python_Clear-Window-Comm... print('zeta_ref ->', zeta_ref, '()') # shows the variable-reference to that set is also clear print('zeta copy ->', zeta copy, '()') # shows the copy of the set made before the clear is still populated python_exercises_00.py with elements python_exercises_01.py Python_Tutorial_Running-Scr... print(nl, 'alpha diff') Python_Tutorials.md print('alpha ->', alpha, '()') # alpha diff = alpha.copy() # copies a set creating a duplicate alpha diff.difference update(beta) # update method using difference print('alpha diff ->', alpha diff, '()') # Difference compares 2 sets removing duplicate elements from the set in parenthesis print(nl, 'alpha intersect') alpha intersect = alpha.copy() # copies a set creating a duplicate alpha intersect.intersection update(beta) # update method using intersection print('alpha intersect ->', alpha intersect, '()') # finds where 2 sets overlap or intersect print(nl, 'alpha sym diff') alpha sym diff = alpha.copy() # copies a set creating a duplicate

Ⅲ …

```
percipio16_set_type.py X
       EXPLORER
      ▲ OPEN EDITORS
                                          alpha sym diff = alpha.copy() # copies a set creating a duplicate
                                           alpha sym diff.symmetric difference update(beta) # update method using symmetric difference
          🕏 percipio16_set_type.py Pe...
Q
                                          print('alpha sym diff ->', alpha sym diff, '()') # Union method minus the intersection method ((where 2 sets
      ▲ PYTHON
                                           have overlapping elements)
       ▶ Automate-Boring-Stuff
89
       ▶ my_code
                                          print(nl, 'alpha union')

▲ Percipio_Python3-Course

                                          alpha union = alpha.copy() # copies a set creating a duplicate
▶ 01 Start
                                          alpha union.update(beta) # update method using union
         ▶ 02_Data-Sequence Types
                                          print('alpha union ->', alpha union, '()') # union combines alpha-set with beta-set also removing any
                                           duplicates
Ċ
         ■ 03_Collections-Mapping-Lo..
          percipio15_range_type_a...
                                          print(nl)
          percipio16_set_type.py
                                          print(nl)
          percipio17_dict_type.py
                                          print('Other examples I made')
          percipio18_while_loop.py
                                          left = set((1, 2, 3, 4, 5, 6, 7, 8, 9, 10))
          percipio19_forloop.py
                                          right = set((2, 4, 6, 8, 10))
                                          print('left ->', left)
          percipio20_if_statement....
                                          print('right ->', right)
          percipio21_exercise_nam...
                                          print(nl, '')
         ▶ 04 Modules-Functions
                                          print('isdisjoint - tests two sets for any shared elements; boolean True with no elements in common' + nl +
         ▶ 05 Classes
                                           'issubset - tests if all elements of set(left) are within all elements of set(right), boolean True when
         ▶ 06_Working-with-Files
                                           subset exists' + nl + 'issuperset - tests if all elements of set(right) are within all elements of set(left),
         ▶ 07_Comprehensions
                                           boolean True when subset exists')
                                          print(nl, '')
         ▶ 08 Iterables-and-Generators
                                          print('left.isdisjoint(right) ->', left.isdisjoint(right), '(are any elements shared between two sets?)')
         ▶ 09 Exceptions
                                          print('right.isdisjoint(left) ->', right.isdisjoint(left))
       Python Projects_2014
                                          print(nl, '')
       ≡ CMD Python Set-Path.txt
                                          print('left.issubset(right) ->', left.issubset(right), '(are left-set-elements all within right-set-elements?)
       Python_Basics.txt
       ₹ Python Clear-Window-Comm..
                                          print('right.issubset(left) ->', right.issubset(left))
                                          print(nl, '')
       python_exercises_00.py
                                          print('left.issuperset(right) ->', left.issuperset(right), '(are right-set-elements all within
       python_exercises_01.py
                                           left-set-elements?)')
       Python_Tutorial_Running-Scr...
                                          print('right.issuperset(left) ->', right.issuperset(left))
       Python_Tutorials.md
```