

ch_11_assignment

April 6, 2023

Copyright (C) 2023 201800294_DongilKim All rights reserved (<https://KimTein.github.io>)

Ch_11_assignment

```
[ ]: from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'
```

1 Storing Data Using Other Collection type

1.1 Storing Data Using Sets

```
[ ]: vowels = {'a','e','i','o','u'}
vowels

vowels = {'a','e','a','a','i','o','u','u'}
vowels

{'a','e','i','o','u'} == {'a','e','a','a','i','o','u','u'}
```

```
[ ]: {'a', 'e', 'i', 'o', 'u'}
```

```
[ ]: {'a', 'e', 'i', 'o', 'u'}
```

```
[ ]: True
```

```
[ ]: set([2, 3, 2, 5])
```

```
[ ]: {2, 3, 5}
```

```
[ ]: set(2,3,5)
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[6], line 1
----> 1 set(2,3,5)

TypeError: set expected at most 1 argument, got 3
```

```
[ ]: set(range(5))
```

```
[ ]: {0, 1, 2, 3, 4}
```

1.1.1 Set Operations

```
[ ]: vowels = {'a','e','i','o','u'}  
vowels  
  
vowels.add('y')  
vowels
```

```
[ ]: {'a', 'e', 'i', 'o', 'u'}
```

```
[ ]: {'a', 'e', 'i', 'o', 'u', 'y'}
```

```
[ ]: ten = set(range(10))  
lows = {0, 1, 2, 3, 4}  
odds = (1, 3, 5, 7, 9)  
  
lows.add(9)  
lows  
  
lows.difference(odds)  
  
lows.intersection(odds)
```

```
[ ]: {0, 1, 2, 3, 4, 9}
```

```
[ ]: {0, 2, 4}
```

```
[ ]: {1, 3, 9}
```

```
[ ]: lows.issubset(ten)  
  
lows.issuperset(odds)  
  
lows.remove(0)  
lows  
  
lows.symmetric_difference(odds)  
  
lows.union(odds)  
  
lows.clear()  
lows
```

```
[ ]: True
```

```
[ ]: False
```

```
[ ]: {1, 2, 3, 4, 9}
```

```
[ ]: {2, 4, 5, 7}
```

```
[ ]: {1, 2, 3, 4, 5, 7, 9}
```

```
[ ]: set()
```

```
[ ]: lows = set([0, 1, 2, 3, 4])
      odds = set([1, 3, 5, 7, 9])

      lows - odds

      lows & odds

      lows <= odds

      lows >= odds

      lows | odds

      lows ^ odds
```

```
[ ]: {0, 2, 4}
```

```
[ ]: {1, 3}
```

```
[ ]: False
```

```
[ ]: False
```

```
[ ]: {0, 1, 2, 3, 4, 5, 7, 9}
```

```
[ ]: {0, 2, 4, 5, 7, 9}
```

1.2 Storing Data Using Tuples

```
[ ]: bases = ('A', 'C', 'G', 'T')
      for base in bases:
          print(base)
```

A

C

G
T

```
[ ]: (8)

type((8))

(8,)

type((8,))

(5 + 3)

(5 + 3,)
```

```
[ ]: 8
```

```
[ ]: int
```

```
[ ]: (8,)
```

```
[ ]: tuple
```

```
[ ]: 8
```

```
[ ]: (8,)
```

```
[ ]: life = (['Canada', 76.5], ['United States', 75.5], ['Mexico', 72.0])
life[0] = life[1]
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[15], line 2
      1 life = (['Canada', 76.5], ['United States', 75.5], ['Mexico', 72.0])
----> 2 life[0] = life[1]

TypeError: 'tuple' object does not support item assignment
```

```
[ ]: life = (['Canada', 76.5], ['United States', 75.5], ['Mexico', 72.0])
life[0][1] = 80.0
life
```

```
[ ]: (['Canada', 80.0], ['United States', 75.5], ['Mexico', 72.0])
```

```
[ ]: canada = ['Canada', 76.5]
usa = ['United States', 75.5]
mexico = ['Mexico', 72.0]
```

```
life = (canada, usa, mexico)
life
```

```
[ ]: (['Canada', 76.5], ['United States', 75.5], ['Mexico', 72.0])
```

```
[ ]: mexico = ['Mexico', 72.5]
life
```

```
[ ]: (['Canada', 76.5], ['United States', 75.5], ['Mexico', 72.0])
```

```
[ ]: life[0][1] = 80.0
canada
```

```
[ ]: ['Canada', 80.0]
```

1.2.1 Assigning to Multiple Variables Using Tuples

```
[ ]: (x, y) = (10, 20)
x
y
```

```
[ ]: 10
```

```
[ ]: 20
```

```
[ ]: 10, 20
x, y = 10, 20
x
y
```

```
[ ]: (10, 20)
```

```
[ ]: 10
```

```
[ ]: 20
```

```
[ ]: s1 = 'first'
s2 = 'second'
s1, s2 = s2, s1
s1
s2
```

```
[ ]: 'second'
```

```
[ ]: 'first'
```

1.3 Storing Data Using Dictionaries

```
[ ]: from typing import TextIO, List, Any
      from io import StringIO
```

```
[ ]: def count_birds(observations_file:TextIO) -> List[List[Any]]:
      bird_counts = []
      for line in observations_file:
          bird = line.strip()
          found = False

          for entry in bird_counts:
              if entry[0] == bird:
                  entry[1] = entry[1] + 1
                  found = True
          if not found:
              bird_counts.append([bird, 1])

      return bird_counts
```

```
[ ]: if __name__ == '__main__':
      with open('observations.txt') as observations_file:
          bird_counts = count_birds(observations_file)

      for entry in bird_counts:
          print(entry[0], entry[1])
```

```
canada goose 5
long-tailed jaeger 2
snow goose 1
northern fulmar 1
```

```
[ ]: bird_to_observations = {'canada goose': 3, 'northern fulmar': 1}
      bird_to_observations

      bird_to_observations['northern fulmar']

      bird_to_observations['canada goose']
```

```
[ ]: {'canada goose': 3, 'northern fulmar': 1}
```

```
[ ]: 1
```

```
[ ]: 3
```

```
[ ]: bird_to_observations['long-tailed jaeger']
```

```

KeyError                                Traceback (most recent call last)
Cell In[30], line 1
----> 1 bird_to_observations['long-tailed jaeger']

KeyError: 'long-tailed jaeger'

```

```

[ ]: dict1 = {'canada goose': 3, 'northern fulmar': 1}
dict2 = {'northern fulmar': 1, 'canada goose': 3}
dict1 == dict2

```

```

[ ]: True

```

1.3.1 Updating and Checking Membership

```

[ ]: bird_to_observations = {}
bird_to_observations['snow goose'] = 33
bird_to_observations['eagle'] = 999

bird_to_observations

bird_to_observations['eagle'] = 9
bird_to_observations

```

```

[ ]: {'snow goose': 33, 'eagle': 999}

```

```

[ ]: {'snow goose': 33, 'eagle': 9}

```

```

[ ]: bird_to_observations = {'snow goose': 33, 'eagle': 9}
del bird_to_observations['snow goose']
bird_to_observations

```

```

[ ]: {'eagle': 9}

```

```

[ ]: del bird_to_observations['gannet']

```

```

-----
KeyError                                Traceback (most recent call last)
Cell In[35], line 1
----> 1 del bird_to_observations['gannet']

KeyError: 'gannet'

```

```

[ ]: bird_to_observations = {'eagle': 990, 'snow goose': 33}
'eagle' in bird_to_observations

if 'eagle' in bird_to_observations:

```

```
print('eagles have been seen')
```

```
[ ]: True
```

```
[ ]: del bird_to_observations['eagle']  
      'eagle' in bird_to_observations  
  
      if 'eagle' in bird_to_observations:  
          print('eagles have been seen')
```

```
[ ]: False
```

1.3.2 Looping over Dictionaries

```
[ ]: bird_to_observations = {'canada goose': 183, 'long-tailed jaeger': 71, 'snow_↵  
      'goose': 63, 'northern fulmar': 1}  
      for bird in bird_to_observations:  
          print(bird, bird_to_observations[bird])
```

```
canada goose 183  
long-tailed jaeger 71  
snow goose 63  
northern fulmar 1
```

1.3.3 Dictionary Operations

```
[ ]: scientist_to_birthdate = {'Newton': 1642, 'Darwin': 1809, 'Turing': 1912}  
  
      scientist_to_birthdate.keys()  
  
      scientist_to_birthdate.values()  
  
      scientist_to_birthdate.items()  
  
      scientist_to_birthdate.get('Newton')  
  
      scientist_to_birthdate.get('Curie', 1867)  
  
      scientist_to_birthdate
```

```
[ ]: dict_keys(['Newton', 'Darwin', 'Turing'])
```

```
[ ]: dict_values([1642, 1809, 1912])
```

```
[ ]: dict_items([('Newton', 1642), ('Darwin', 1809), ('Turing', 1912)])
```

```
[ ]: 1642
```



```
[ ]: 1867
```

```
[ ]: {'Newton': 1642, 'Darwin': 1809, 'Turing': 1912}
```

```
[ ]: scientist_to_birthdate = {'Newton': 1642, 'Darwin': 1809, 'Turing': 1912}
researcher_to_birthdate = {'Curie': 1867, 'Hopper': 1906, 'Franklin': 1920}

scientist_to_birthdate.update(researcher_to_birthdate)
scientist_to_birthdate

researcher_to_birthdate

researcher_to_birthdate.clear()
researcher_to_birthdate
```

```
[ ]: {'Newton': 1642,
      'Darwin': 1809,
      'Turing': 1912,
      'Curie': 1867,
      'Hopper': 1906,
      'Franklin': 1920}
```

```
[ ]: {'Curie': 1867, 'Hopper': 1906, 'Franklin': 1920}
```

```
[ ]: {}
```

Reference * Title: Physics Programming Lecture Note (INU) * Author: Jeongwoo Kim, Ph.D. *
Availability: <https://sites.google.com/view/jeongwookim>

Copyright (C) 2023 201800294_DongilKim All rights reserved (<https://KimTein.github.io>)