Software Programming Lab.

06 - List, Tuple, Dictionary, Set

SWPLab06

Exercises

```
nlist = [ i*i for i in range(10) if i > 3 ]
print(nlist)

d = { 'student' + str(x+1) : x**2 for x in range(10)}
print(d)
```

```
d = {'money': 12, 'candy': 3, 'tissues': 20}
d['candy'] = d['candy'] + 2
d['tissues'] = 10
print(d)
```

```
x = [3, 5, 7, 9]
x.reverse()
print(x)
```

```
x = [3, 5, 7, 9]
print(x[2::-1])
print(x[:1:-1])
print(x[::-1])
```

```
counts = dict()
names = ['영구', '길동', '일등', '길동', '일등', '일등', '길동', '농민']
for name in names :
    counts[name] = counts.get(name, 0) + 1
print(counts)
```

```
d = {'pork': 25.3, 'beef': 33.8, 'chicken':22.7}

for k in d:
    print(k)

for v in d.values():
    print(v)

for k, v in d.items():
    print(k, v)
```

```
a = set([2, 4, 5, 6, 10, 13, 2, 2])
b = set([1, 3, 4, 5, 8, 9, 10, 10, 10, 10])
c = set([2, 10])

print(a)
print(b)
print(a - b )
print(a | b )
print(a & b )
print(a ^ b )
print(a > c)
```

Problems

Problem 1

- Write a program to combine following dictionaries to create a new one.

```
dic1 = {'a': 100, 'b': 150, 'c': 240}
dic2 = {'b': 200, 'e': 450, 'f': 330}
dic3 = {'c': 100, 'f': 110, 'e': 500, 'a':60 }
```

- Example outputs

```
{'a': 160, 'b': 350, 'c': 340, 'e': 950, 'f': 440}
```

Problem 2

- Write a program that find the **10 most frequent words** from a text file (alice.txt)
    1. Read all content from alice.txt
    2. Make a list of words
    3. Change all to uppercase
    4. Count all words using a dict - YOUR CODE
    5. Make a list of (value, key) - YOUR CODE
    6. Sort the list in descending order - YOUR CODE
    7. Print the sorted word list

- Example outputs

```
 1) THE          1631
 2) AND           844
 3) TO            721
 4) A             627
 5) SHE           537
 6) IT            526
 7) OF            508
 8) SAID          462
 9) I             398
10) ALICE         385
```

- Given code (1)

```
# read all content from alice.txt

with open('alice.txt', 'r') as f:
    content = f.read()
    # remove punctuation
    import re
    content = re.sub(r'[^\w\s]','', content)


# make a list of words

dict_words = {}
words = content.split()


# change all to uppercase

for i in range(len(words)):
    words[i] = words[i].upper()


......
```

- Given code (2)

```
# count all words

for word in words:
    dict_words[word] = None # MODIFY THIS (1)


# make a list of (value, key)

list_words = []
for k, v in dict_words.items():
    list_words.append( (None, None) ) # MODIFY THIS (2)


# sort the list in descending order
# (hint) to sort in descending order, you can sort like this: [ some_list.sort(reverse=True) ]

# YOUR CODE HERE (3)

# print the sorted word list
for i, t in enumerate( list_words[:10] ):
    print( '{:2d}) {:10} {:5}'.format(i+1, t[1], t[0]) )
```

Problem 3

- Write a program that estimates the average housing prices by city from a text file (transactions.csv).
  - For each rows in the file:
    1. Skip the first row(a header row).
    2. Get all columns using **split()** method from a row. - YOUR CODE
    3. Accumulate count and price using two dicts - YOUR CODE
    4. Use a set to save unique city names.
  - Convert the set into a list to sort and iterate the city names. - YOUR CODE
  - For each cities:
    - Calculate and print the average housing price by city - YOUR CODE

- Example outputs

```
ANTELOPE              $ 232.50k
AUBURN                $ 405.89k
CAMERON PARK          $ 267.94k
CARMICHAEL            $ 295.68k
CITRUS HEIGHTS        $ 187.11k
COOL                  $ 300.00k
DIAMOND SPRINGS       $ 216.03k
EL DORADO             $ 247.00k
EL DORADO HILLS       $ 491.70k
ELK GROVE             $ 271.16k
ELVERTA               $ 132.87k
FAIR OAKS             $ 303.50k
FOLSOM                $ 414.96k
FORESTHILL            $ 194.82k
GALT                  $ 236.94k
GARDEN VALLEY         $ 490.00k
GOLD RIVER            $ 358.00k
GRANITE BAY           $ 678.73k
GREENWOOD             $ 395.00k
LINCOLN               $  96.54k
LOOMIS                $ 567.00k
MATHER                $ 237.80k
MEADOW VISTA          $ 230.00k
NORTH HIGHLANDS       $ 135.66k
ORANGEVALE            $ 279.16k
PENRYN                $ 506.69k
PLACERVILLE           $ 363.86k
POLLOCK PINES         $ 240.30k
RANCHO CORDOVA        $ 263.41k
RANCHO MURIETA        $ 297.75k
RIO LINDA             $ 172.73k
ROCKLIN               $ 381.84k
ROSEVILLE             $ 324.53k
SACRAMENTO            $ 197.74k
SHINGLE SPRINGS       $ 275.00k
SLOUGHHOUSE           $   2.00k
WALNUT GROVE          $ 380.00k
WEST SACRAMENTO       $ 170.70k
WILTON                $ 617.51k
```

- Given code (1)

```
set_cities = set()
dict_count = {}
dict_sum = {}

with open('transactions.csv', 'r') as f:

    is_first = True

    # for each row
    for line in f:

        # skip header row
        if is_first:
            is_first = False
            continue

        # split a row into columns
        columns = None ### MODIFY THIS (1)

        # read a city name from column 1
        city = None ### MODIFY THIS (2)
        # read price from column 9
        price = None ### MODIFY THIS (3)

        # increase count by 1
        dict_count[city] = None ### MODIFY THIS (4)
        # accumulate the price
        dict_sum[city] = None ### MODIFY THIS (5)

        # save a city name
        set_cities.add(city)

......
```

- Given code (2)

```
# convert the set to a list and sort
list_cities = None ### MODIFY THIS (6)
list_cities ### MODIFY THIS (7)


# for each city
for city in list_cities:

    # access dict_count with a city name
    count = None ### MODIFY THIS (8)
    # access dict_sum with a city name
    price = None ### MODIFY THIS (9)

    print('{:20} ${:7.2f}k'.format(city, price / count / 1000 ) )
```