Objectives for class 13

--- Chapter 7 ---

7.8 To invoke a list’s append, count, extend, index, insert, pop, remove, reverse, and sort methods (§7.2.9).

7.9 To split a string into a list using the str’s split method (§7.2.10).

7.10 To develop and invoke functions that pass list arguments (§7.6).

7.11 To develop and invoke functions that return lists (§7.7).

--- Chapter 8 ---

8.1 To learn how a two-dimensional list can represent two-dimensional data (§8.1).

8.2 To access elements in a two-dimensional list by using row and column indexes (§8.2).

8.3 To program common operations for two-dimensional lists (displaying lists, summing all elements) (§8.2).

Add more elements to a list

append(x)

Add a single element to the end

extend(anotherList)

Add all elements from another list to the end

insert(index, x)

Insert an element x at a given index

```
>>> t1=[1,2,3]
>>> t1.append(-1)
>>> t1
[1, 2, 3, -1]
>>> t2=[4,5]
>>> t1.extend(t2)
>>> t1
[1, 2, 3, -1, 4, 5]
>>> insert(0,2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'insert' is not defined
>>> t1.insert(0,2)
>>> t1
[2, 1, 2, 3, -1, 4, 5]
>>> t1.insert(2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: insert expected 2 arguments,
got 1
```

Remove more elements from a list

pop()

Remove the last element

pop(index)

Remove the element at the given index

remove(x)

Remove the first x

```
>>> t1=[1,2,2,3]
>>> t1.pop()
3
>>> t1
[1, 2, 2]
>>> t1.pop(0)
1
>>> t1
[2, 2]
>>> t1.remove(2)
>>> t1
[2]
```

Search for elements in a list

count(x)

Return the count of x

index(x)

Return the index of first x

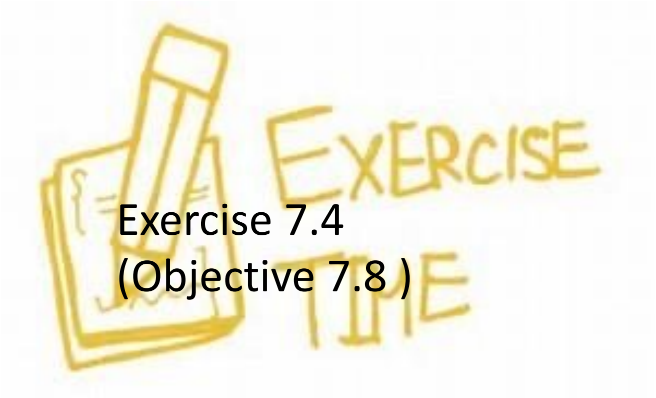
```
>>> t1=[1,2,2,3]

>>> t1.count(2)
2

>>> t1.index(2)
1
>>> t1.index(1)
0
```

Sort elements from a list

sort()

Sort the elements in the list in an ascending order

```
>>> t1=[2,3,4,1,32,4,19]

>>> t1.sort()
2

>>> t1
[1, 2, 3, 4, 4, 19, 32]
```



Exercise 7.4
(Objective 7.8)

Difference between traversing a string and a list of strings using **for… in…**

- Read a string character by character

```
cheese = 'Cheddar'
for ch in cheese:
    print(ch)
```

```
C
h
e
d
d
a
r
```

- Read a list of strings string by string

```
cheeses = ['Cheddar', 'Edam', 'Gouda']
for cheese in cheeses:
    print(cheese)
```

```
Cheddar
Edam
Gouda
```

Split a string into a list of characters/sub-strings

list(string)

Breaks a string into individual characters

split()

Breaks a string into words separated by spaces

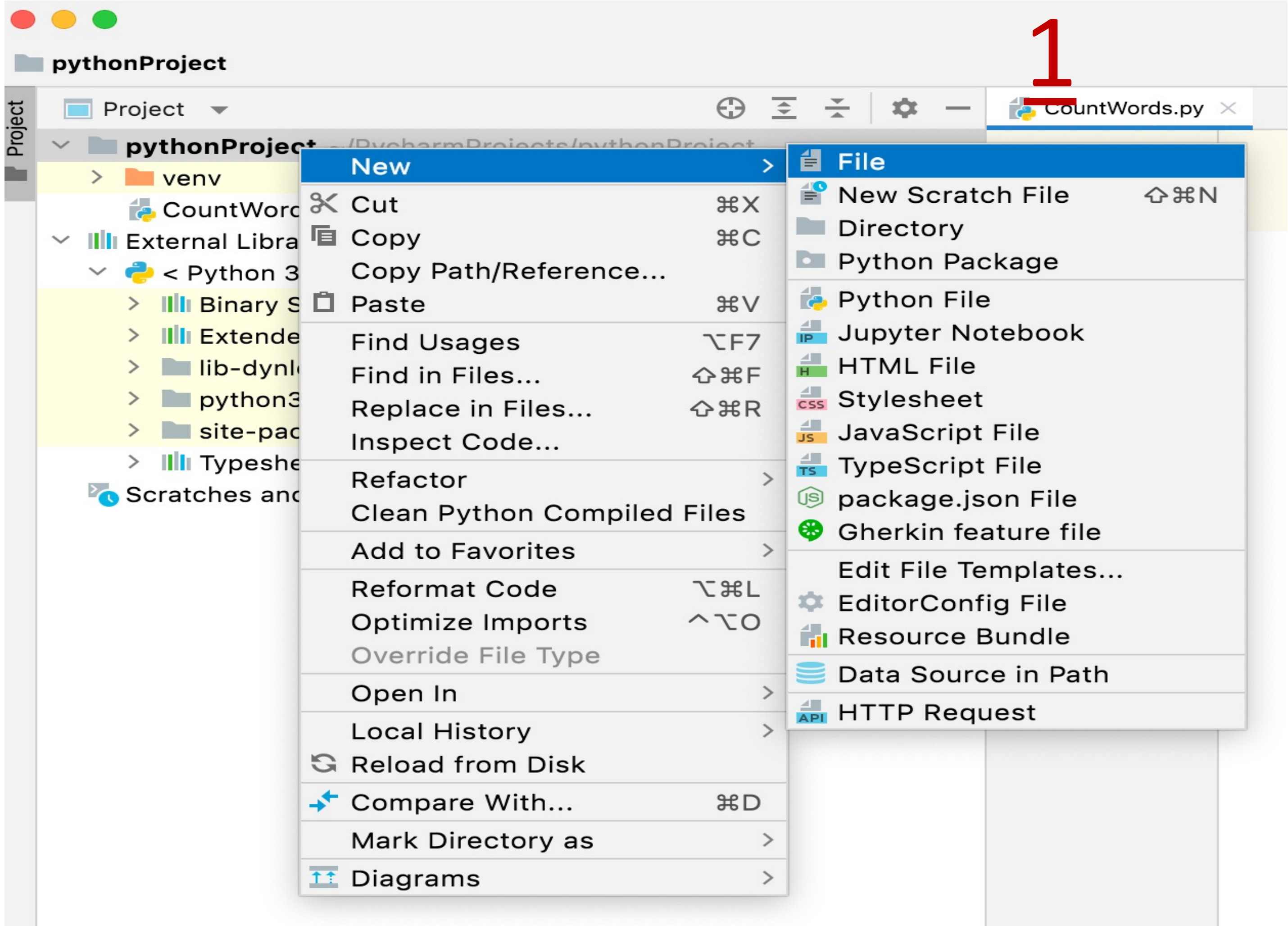
split(delimiter)

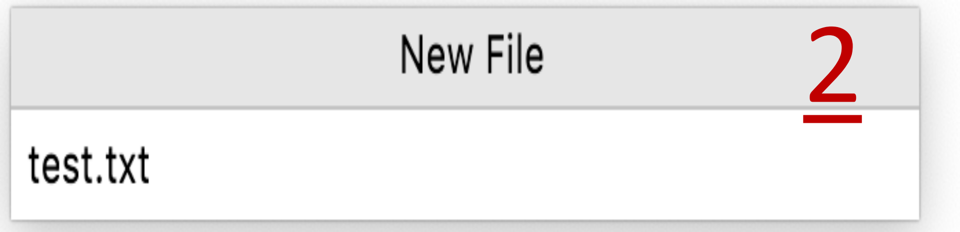
Breaks a string into a list of sub-strings separated by a given delimiter

```
>>> t="spam"
>>> s="spam"
>>> t=list(s)
>>> s = 'pining for the fjords'
>>> t=s.split()
>>> t
['pining', 'for', 'the', 'fjords']
>>> s = 'spam-spam-spam'
>>> t = s.split('-')
>>> t
['spam', 'spam', 'spam’]
>>> items = "Welcome to the US".split('-')
>>> items
['Welcome', 'to', 'the', 'US']
```

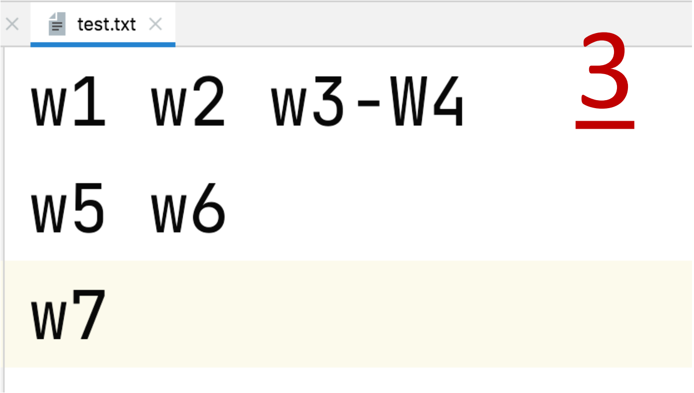
Read a file and count number of words

Step 1: Create a python program “CountWords.py” ; Then create a text file “test.txt” within the same folder as your python





test.txt



```
w1 w2 w3-W4
w5 w6
w7
```

Step 2: Open your python program “CountWords.py”, write codes according to the system design as below

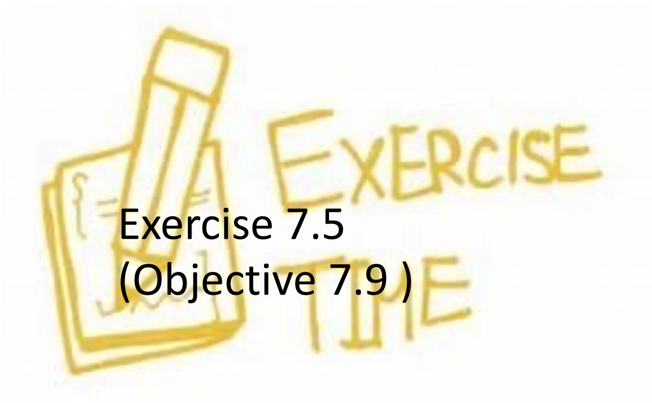
*#1. **Open** the file "test.txt"*

*#2. **Read** the file and store the lines into a **string** variable*

#3. Split the string into a list of words

#4. Get the length of list and display it

```
#1. Open the file "test.txt"
f =open("test.txt")

#2. Read the file and store the lines into a string
textStr = f.read()
print(textStr)

#3. Split the string into a list of words
words=textStr.split()

#4. Get the length of list and display
count=len(words)
print("Number words is ",count)
```



Exercise 7.5
(Objective 7.9)

Pass a list into function

- Function parameter can be a list
- A **list variable** can be passed to a function
- An **anonymous list** can also be passed to a function

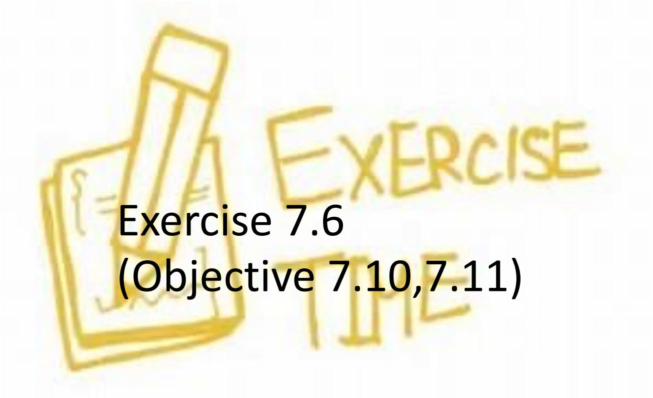
```
def printList(lst):
    for element in lst :
        print(element)


#Pass a list variable
t = [3, 1, 2, 6, 4, 2]
printList(t)


#Pass an anonymous list
printList([3, 1, 2, 6, 4, 2])
```

Return a list from a function

- A list can be returned by a function
- Note that list already has the reverse method reverse()
  - For example, list1.reverse()

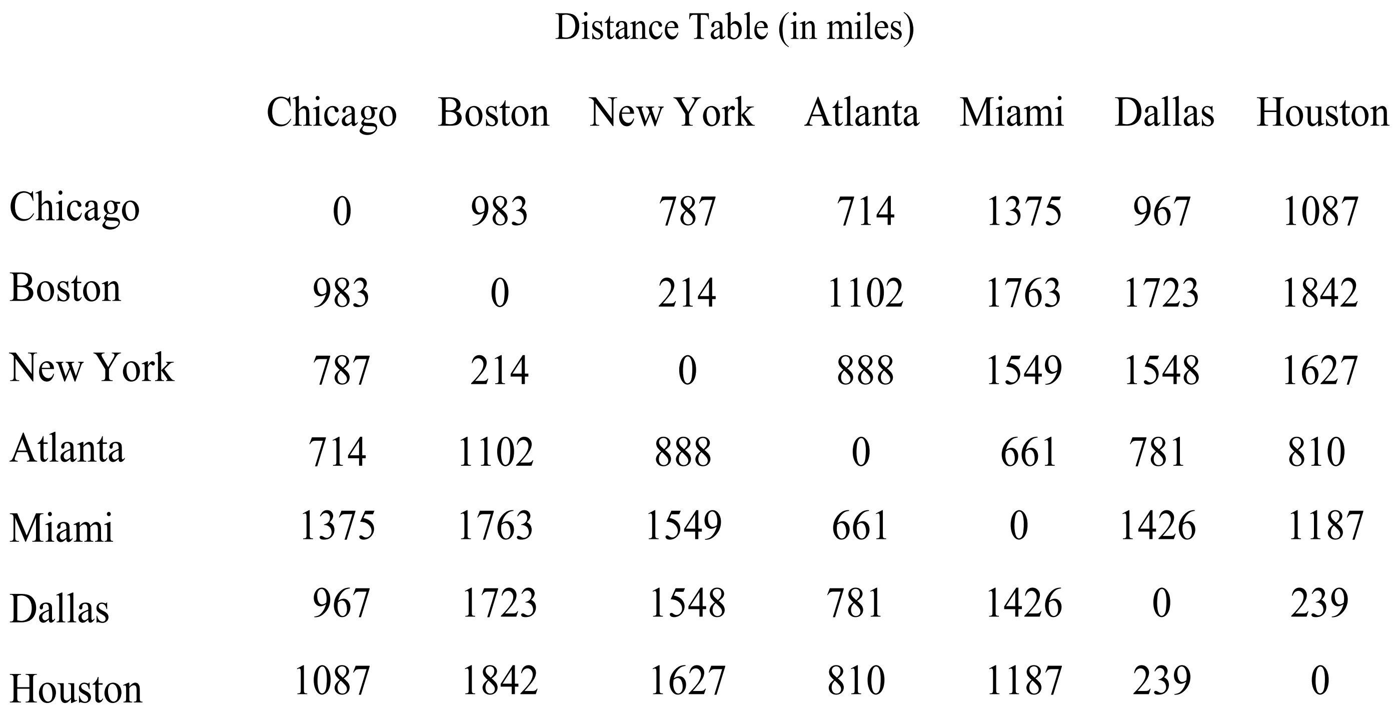
```
def reverse(list):
    result = []

    for element in list:
        result.insert(0, element)

    return result
```

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = reverse(list1)
print(list2)
```



Exercise 7.6
(Objective 7.10,7.11)

Store a table of values using a 2D list (list of sub-list)



Distance Table (in miles)

	Chicago	Boston	New York	Atlanta	Miami	Dallas	Houston
Chicago	0	983	787	714	1375	967	1087
Boston	983	0	214	1102	1763	1723	1842
New York	787	214	0	888	1549	1548	1627
Atlanta	714	1102	888	0	661	781	810
Miami	1375	1763	1549	661	0	1426	1187
Dallas	967	1723	1548	781	1426	0	239
Houston	1087	1842	1627	810	1187	239	0

```
distances = [
    [0, 983, 787, 714, 1375, 967, 1087],
    [983, 0, 214, 1102, 1763, 1723, 1842],
    [787, 214, 0, 888, 1549, 1548, 1627],
    [714, 1102, 888, 0, 661, 781, 810],
    [1375, 1763, 1549, 661, 0, 1426, 1187],
    [967, 1723, 1548, 781, 1426, 0, 239],
    [1087, 1842, 1627, 810, 1187, 239, 0]
]
```

- A table stored as a list of rows
- Each row stored as a sub-list
- A list of sub-list also called **multidimensional list**.
- A multidimensional list storing a table of values is called as **2D list**

How to access data from a 2D list?

- The rows can be accessed using the **row index**.
- The values in each row can be accessed through **column index**.

```
matrix = [
    [1, 2, 3, 4, 5],
    [6, 7, 0, 0, 0],
    [0, 1, 0, 0, 0],
    [1, 0, 0, 0, 8],
    [0, 0, 9, 0, 3],
]
```

	[0]	[1]	[2]	[3]	[4]
[0]	1	2	3	4	5
[1]	6	7	0	0	0
[2]	0	1	0	0	0
[3]	1	0	0	0	8
[4]	0	0	9	0	3

```
matrix[0] is [1, 2, 3, 4, 5]
matrix[1] is [6, 7, 0, 0, 0]
matrix[2] is [0, 1, 0, 0, 0]
matrix[3] is [1, 0, 0, 0, 8]
matrix[4] is [0, 0, 9, 0, 3]

matrix[0][0] is 1
matrix[4][4] is 3
```

Initialize a 2D list with input values

```
matrix = [] # Create an empty list
numberOfRows = int(input("Enter the number of rows: "))
numberOfColumns = int(input("Enter the number of columns: "))

for row in range(0, numberOfRows):
    matrix.append([]) # Add an empty new row

    for column in range(0, numberOfColumns):
        value = eval(input("Enter an element and press Enter: "))
        matrix[row].append(value)

print(matrix)
```

Read a 2D list and summing all elements

```
# Assume a list is given
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
total = 0

for row in range(0, len(matrix)):
    for column in range(0, len(matrix[row])):
        total += matrix[row][column]

print("Total is " + str(total)) # Print the total
```

Exercise 8
(Objective 8.1,8.2,8.3)

len(matrix) returns the number of rows.
len(matrix[row] returns the number of columns