Unit4_Assessment

June 16, 2023

1 Unit 4 Career Preparation: Technical Assessment

1.1 Problem 1

Write a script that: * Reads the file `problem1.txt`. * Adds each line to a new list. * Prints the new list.

[41]:
```
# assign the absolute path to file_name
file_name = "/voc/public/problem1.txt"

# open file to read it by declaring variable open_file
open_file = open(file_name, "r")

# assign new_list as variable f with a list containing each line in the file as␣
 ↪a list item
with open_file as f:
    new_list = f.readlines()

# variable lst holds the list to remove a new line character
lst = ['item1\n','item2\n','item3\n','item4\n', 'item5\n']

# declaring an empty list in a name of new_lst
rep = []
# Creating for loop to iterate till the end of the list
for x in lst:
    # using the append() function to add at the end of the list
    # using strip function to remove the newline character
    rep.append(x.replace("\n", ""))
# print new list
print(list(rep))
```

```
['item1', 'item2', 'item3', 'item4', 'item5']
```

1.2 Problem 2

Write a script that: * Reads the file `problem2.txt`. * Counts how many times 192.168.1.1 appears in the file. * Prints the result.

[132]:
```
# assign the absolute path to f
f = "/voc/public/problem2.txt"

#Make the IP address address a variable
ip = ("192.168.1.1")

# Create a counter and set it 0
counter = 0

# Open the absolute path in read mode as variable f
with open("/voc/public/problem2.txt", "r") as f:

# Create a list from problem2.txt contents
    List = open("/voc/public/problem2.txt").readlines()

# Remove all new lines from List
    newList = [item.strip() for item in List]


# Iterate over the lines of the file for i in newlist
    for i in newList:
        if i == ip:

            counter += 1

# Print the list of lines
print(counter)
```

```
5
```

1.3 Problem 3

Write a script using a function (`dedupe`) that: * Takes a list `l = [1,5,7,2,4,3,5,1,6,2,6]`. * Returns a new list that contains all of the elements from the first list, excluding duplicates.

[144]:
```
def dedupe():
#declare variable some_list
    some_list = [1,5,7,2,4,3,5,1,6,2,6]

#print original list with duplicates
    print ("The list is: " + str(some_list))

#declare new variable new_list that turns former list into a set without␣
 ↪duplicates
    new_list = list(set(some_list))

#print new list without duplicates
print(new_list)
```

```
[1, 2, 3, 4, 5, 6, 7]
```

1.4 Problem 4

Write a program (using a function) that: * Asks the user for a long string containing multiple words. * Prints back the same string, except with the words in reverse order.

For example, if the user types the string: 'My name is robert', it will print 'robert is name My'.

[169]:
```
# assign user variable with input containing multiple words
user = input("What are 5 names?")

# assign reversee as user by slicing list and reversing the order
reversee =user[::-1]

# print final reversed list
print(reversee)
```

```
What are 5 names? tim jon hun alex tus

sut xela nuh noj mit
```

1.5 Problem 5

Write a script that: * Opens the file `problem5.txt`. * Counts each port and puts the results in a dictionary.

[170]:
```
# assign absolute path to file_name
file_name = "/voc/public/problem5.txt"

# make empty list for port_counts
port_counts = {}

# open file probelm5.txt for reading
with open(file_name, "r") as file:

# assign contents as file but read
    contents = file.read()
# assign ports as contents but split string into a list
    ports = contents.split("\n")

# for loop organizes ports
for port in ports:
    if port not in port_counts:
        port_counts[port] =1
    else:
        port_counts[port] +=1

# within port, view objects of dictionary are returned and result is printed
for port, count in port_counts.items():
```

```
    print(f"Port {port}: {count} occurences")
```

```
Port 80: 7 occurences
Port 443: 3 occurences
Port 22: 5 occurences
Port 21: 2 occurences
Port 25: 3 occurences
Port 389: 1 occurences
Port 3389: 1 occurences
Port 445: 3 occurences
Port : 2 occurences
```

```
[ ]:
```

END