Discussion 3 - Recording Notebook

DSC 20, Spring 2023

Question 1

What should be the values for x,y,z if I want to print out "summer!"

Note: Don't forget the order of evaluation! (NAO)

Question 1

What should be the values for x,y,z if I want to print out "summer!"

Note: Don't forget the order of evaluation! (NAO)

In [ ]:

```
if x and y and z:
    print('fall')
elif not x and not y:
    print('winter...')
elif not z or x:
    print('spring?')
else:
    print('summer!')
```

Question 1 Solution

Question 1 Solution

In [3]:
```
x = False
y = True
z = True

if x and y and z:
    print('fall')
elif not x and not y:
    print('winter...')
elif not z or x:
    print('spring?')
else:
    print('summer!')
```

```
summer!
```

Question 2

Given 2 lists, remove the elements from the first list if they're present in the second one.

Your solution must be 1 line and use filter

note: filter has not been covered quite yet at the time of this discussion, don't worry if this question is confusing - consider it more as a preview of friday's lecture (It is a covered topic on midterm 1).

Question 2

Given 2 lists, remove the elements from the first list if they're present in the second one.

Your solution must be 1 line and use filter

note: filter has not been covered quite yet at the time of this discussion, don't worry if this question is confusing - consider it more as a preview of friday's lecture (It is a covered topic on midterm 1).

In [2]:
```
def up_join(lst1, lst2):
    """
    >>> grades = ['A', 'B', 'C', 'D', 'F']
    >>> grade_filter = ['D', 'F']
    >>> up_join(grades, grade_filter)
    ['A', 'B', 'C']
    """
```

Question 2 Solution

Question 2 Solution

In [5]:
```
def up_join(lst1, lst2):
    """
    >>> grades = ['A', 'B', 'C', 'D', 'F']
    >>> grade_filter = ['D', 'F']
    >>> up_join(grades, grade_filter)
    ['A', 'B', 'C']
    """
    return list(filter(lambda x: x not in lst2, lst1))
```

Question 3

Given a dictionary consisting of strings for keys and values and another string, create a new dictionary where the key is the new string and the value is the list of keys who had it as their value. **Write assert statements to check input**.

Question 3

Given a dictionary consisting of strings for keys and values and another string, create a new dictionary where the key is the new string and the value is the list of keys who had it as their value. **Write assert statements to check input**.

In [16]:
```
def flip_dict(owners, pet):
    """
    >>> sample = {'ben':'cats', 'charisse':'dogs', 'nikki':'cats'}
    >>> flip_dict(sample, 'cats')
    {'cats': ['ben', 'nikki']}
    """
```

Question 3 Solution

Question 3 Solution

In [17]:

```
def flip_dict(owners, pet):
    """
    >>> sample = {'ben':'cats', 'charisse':'dogs', 'nikki':'cats'}
    >>> flip_dict(sample, 'cats')
    {'cats': ['ben', 'nikki']}
    """
    assert isinstance(owners, dict)
    assert isinstance(pet, str)
    assert all([isinstance(key, str) for key in list(owners.keys())])
    assert all([isinstance(val, str) for val in list(owners.values())])
    output = {}
    output[pet] = []
    for owner, pet_type in owners.items():
        if pet_type == pet:
            output[pet].append(owner)
    return output
```

Question 4

Given a file containing an expression on each line, write a function that classifies them as 'energetic' if it ends with !, 'confused' if it ends with a ?, and 'neutral' otherwise.

sample file

```
hello
omg!
ok?
cool
```

Question 4

Given a file containing an expression on each line, write a function that classifies them as 'energetic' if it ends with !, 'confused' if it ends with a ?, and 'neutral' otherwise.

sample file

```
hello
omg!
ok?
cool
```

In [31]:

```
def text_classifier(filepath):
    """
    >>> text_classifier(files/mood.txt)
    ['neutral', 'energetic', 'confused', 'neutral']
    """
```

Question 4 Solution

Question 4 Solution

In [1]:

```
def text_classifier(filepath):
    """
    >>> text_classifier(files/mood.txt)
    ['neutral', 'energetic', 'confused', 'neutral']
    """
    with open(filepath, 'r') as f:
        data = f.readlines()
        # can also write as a for loop, no restriction
        #for this question
        return ['energetic' if x.strip()[-1]=='!' else \
    'confused' if x.strip()[-1]=='?' else 'neutral' for x in data]
```

Question 5

Write a function that takes in a list of strings with the pattern of first_name, last_name, first_name,... and strings together first and last names with a single space in between. If the length of the list is odd, insert your own last name.

Question 5

Write a function that takes in a list of strings with the pattern of first_name, last_name, first_name,... and strings together first and last names with a single space in between. If the length of the list is odd, insert your own last name.

In [39]:
```
def combine_names(names):
    """
    >>> combine_names(['Charisse', 'Hao', 'Nicole', 'Zhang'])
    ['Charisse Hao', 'Nicole Zhang']
    >>> combine_names(['Charisse', 'Hao', 'Ben'])
    ['Charisse Hao','Ben Chen']
    """
```

Question 5 Solution

Question 5 Solution

In [57]:

```
def combine_names(names):
    """
    >>> combine_names(['Charisse', 'Hao', 'Nicole', 'Zhang'])
    ['Charisse Hao', 'Nicole Zhang']
    >>> combine_names(['Charisse', 'Hao', 'Ben'])
    ['Charisse Hao','Ben Chen']
    """
    output = []
    if len(names)%2 == 1:
        names.append('Chen')
    for i in range(0,len(names), 2):
        output.append(' '.join(names[i:i+2]))
    return output
```

Question 6

Given a dictionary that has lists as values, Write a function that returns a list that consists of the length of each list in the dictionary

Question 6

Given a dictionary that has lists as values, Write a function that returns a list that consists of the length of each list in the dictionary

In [62]:

```
def count_values(entries):
    """
    >>> count_values({1: [1,2,3], 2:[3,4,5,6]})
    [3,4]
    """
```

Question 6 Solution

Question 6 Solution

In [63]:
```
def count_values(entries):
    """
    >>> count_values({1: [1,2,3], 2:[3,4,5,6]})
    [3,4]
    """
    return [len(entry) for entry in entries.values()]
```

Question 6 Solution

In [63]:
```
def count_values(entries):
    """
    >>> count_values({1: [1,2,3], 2:[3,4,5,6]})
    [3,4]
    """
    return [len(entry) for entry in entries.values()]
```

In [64]:
```
count_values({1: [1,2,3], 2:[3,4,5,6]})
```

Out[64]:
```
[3, 4]
```

Question 7

Write a function that takes two lists of the same length that contains integers and returns true if the first list is strictly greater than the second list. **One line solution**.

Question 7

Write a function that takes two lists of the same length that contains integers and returns true if the first list is strictly greater than the second list. **One line solution**.

In [67]:
```
def greater_comparison(lst1, lst2):
    """
    >>> greater_comparison([10,20,30], [1,2,3])
    True
    >>> greater_comparison([0,0,4], [1,2,3])
    False
    """
```

Question 7 Solution

Question 7 Solution

In [71]:
```
def greater_comparison(lst1, lst2):
    """
    >>> greater_comparison([10,20,30], [1,2,3])
    True
    >>> greater_comparison([0,0,4], [1,2,3])
    False
    """
    return all([lst1[idx] > lst2[idx] for idx in range(len(lst1))])
```

Question 8

Write a function that takes in a matrix and a number. It returns the result of the multiplication. **You may only use list comprehension and the solution must be 1 line**.

Question 8

Write a function that takes in a matrix and a number. It returns the result of the multiplication. **You may only use list comprehension and the solution must be 1 line**.

In [75]:
```
def matrix_multiplication(matrix, coefficient):
    """
    >>> mtx = [[1,2,3],[4,5,6],[7,8,9]]
    >>> matrix_multiplication(mtx, 3)
    [[3, 6, 9], [12, 15, 18], [21, 24, 27]]
    """
```

Question 8 Solution

Question 8 Solution

In [77]:
```
def matrix_multiplication(matrix, coefficient):
    """
    >>> mtx = [[1,2,3],[4,5,6],[7,8,9]]
    >>> matrix_multiplication(mtx, 3)
    [[3, 6, 9], [12, 15, 18], [21, 24, 27]]
    """
    return [[element*coefficient for element in row] for row in matrix]
```