Excercise_1_solutions

May 30, 2019

1 LECTURE 1

What is 7 to the power of 4?

```
In [1]: print(7**4)
```

```
2401
```

Split this string: s = "Hi there Sam!" into a list and replace Sam with dad.

```
In [2]: s = 'Hi there Sam!'
        s = s.replace('Sam', 'dad')
        s = list(s.split())
        s
```

```
Out[2]: ['Hi', 'there', 'dad!']
```

```
In [3]: s = 'Hi, there is Sam!'
        Slist = list(s.split())
        for s in Slist:
          print(s)
          if s in "Sam!":
            s.replace(s,'dad')
            print('sssssss')
        print(Slist)
```

```
Hi,
there
is
Sam!
sssssss
['Hi,', 'there', 'is', 'Sam!']
```

Given the variables: planet = "Earth" diameter = 12742 Use .format() to print the following string: The diameter of Earth is 12742 kilometers.

```
In [4]: planet = 'Earth'
        diameter = 12742
        s = "The diameter of {} is {} kilometers."
        print(s.format(planet, diameter))
```

```
The diameter of Earth is 12742 kilometers.
```

Given this nested list, use indexing to grab the word "hello" lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]

```
In [5]: lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
        print(lst[3][1][2])
```

```
['hello']
```

Given this nested dictionary grab the word "hello". Be prepared, this will be annoying/tricky d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}

```
In [6]: d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
        lst = d['k1']
        dict = lst[3]
        lst2 = dict['tricky']
        dict2 = lst2[3]
        lst3 = dict2['target']
        print(lst3[3])
```

```
hello
```

What is the main difference between a tuple and a list? Intentionally generate an error by changing a tuple value.

```
In [7]: tpl = ("a", "b", 1, 3, 6)
        print(tpl[2])
        tpl[2] = 3
```

```
1
```

```
        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-7-be4f01d05ddc> in <module>
          1 tpl = ("a", "b", 1, 3, 6)
          2 print(tpl[2])
    ----> 3 tpl[2] = 3
```

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

Create a function that grabs the email website domain from a string in the form: super_user@ee.pw.edu.pl So for example, passing "super_user@ee.pw.edu.pl" would return: ee.pw.edu.pl

```
In [8]: email = 'kamil_rajewski@ee.pw.edu.pl'
        def mail_convert(mail):
          s =  mail.split("@")
          converted = s[1]
          print(converted)
        mail_convert(email)
```

```
ee.pw.edu.pl
```

Create a basic function that returns True if the word 'car' is contained in the input string. Don't worry about edge cases like a punctuation being attached to the word car, but do account for capitalization.

```
In [9]: str1 = "My car is fine"
        str2 = "I have 2 cars"
        str3 = "Meh"

        def car_exist(str):
          car = "car"
          if car in str:
            print("Success")
          else:
            print("Unsuccessfull")

        car_exist(str1)
        car_exist(str2)
        car_exist(str3)
```

```
Success
Success
Unsuccessfull
```

Create a function that counts the number of times the word "car" occurs in a string. Again ignore edge cases. Example: countCar('This car runs faster than the other car dude!')

```
In [10]: str1 = "My car is fine"
         str2 = "I have 2 car, my father got 3, sister got 5 car, everyone like cars"
         str3 = "Meh"
```

```
        def car_exist(str):
          car = "car"
          x = 0
          converted = str.split()
          converted_len = len(converted)
          for slowo in range(0, converted_len):
            if car in converted[slowo]:
              x += 1
          print(x)

        car_exist(str1)
        car_exist(str2)
        car_exist(str3)
```

```
1
3
0
```

Use lambda expressions and the filter() function to filter out words from a list that don't start with the letter 's'. For example: seq = ['soup','dog','salad','cat','great'] should be filtered down to: ['soup','salad']

```
In [11]: seq = ['soup','dog','salad','cat','great', 'asa']
         list(filter( lambda seq: 's' in seq[0], seq))
```

```
Out[11]: ['soup', 'salad']
```

Final Problem You are driving a little too fast, and a police officer stops you. Write a function to return one of 3 possible results: "No ticket", "Small ticket", or "Big Ticket". If your speed is 60 or less, the result is "No Ticket". If speed is between 61 and 80 inclusive, the result is "Small Ticket". If speed is 81 or more, the result is "Big Ticket". Unless it is your birthday (encoded as a boolean value in the parameters of the function) – on your birthday, your speed can be 5 higher in all cases.

```
In [12]: items = [1, 2, 3, 4, 5]
         squared = []
         for i in items:
             squared.append(i**2)
         squared
```

```
Out[12]: [1, 4, 9, 16, 25]
```

```
In [13]: birthday = 1
         s1 = 50
         s2 = 70
         s3 = 100
         s4 = 83
```

```
def check_speed(speed, is_birthday):
  if is_birthday == True:
    speed = speed - 5
  if speed <= 60:
    print("No Ticket")
  if speed > 60 and speed <= 80:
    print("Small ticket")
  if speed >80:
    print("Big Ticket")

check_speed(s1, birthday)
check_speed(s2, birthday)
check_speed(s3, birthday)
check_speed(s4, birthday)

birthday = 0

check_speed(s1, birthday)
check_speed(s2, birthday)
check_speed(s3, birthday)
check_speed(s4, birthday)
```

```
No Ticket
Small ticket
Big Ticket
Small ticket
No Ticket
Small ticket
Big Ticket
Big Ticket
```

In [ ]:

In [ ]: