

ASSIGNMENT-3

Date	13 October 2022
Team ID	PNT2022TMID54263
Project Name	Retail Store Stock Inventory Analytics
Team members	Prithiv R S, Lokeshwaran k, Maharaja AbishekA, Mahammad Mubharak M

```
** What is 7 to the power of 4?**
```

Exercises

Answer the questions or complete the tasks outlined in bold below, use the specific method described if applicable.

```
7**4
```

```
2401
```

```
** Split this string:**
```

```
s = "Hi there Sam!"
```

into a list.

```
s = "Hi there Sam!"
```

```
s = 'Hi there sam!'
s.split()
```

```
['Hi', 'there', 'dad!']
```

```
** Given the variables:**
```

```
planet = "Earth"
diameter = 12742
```

```
** Use .format() to print the following string: **
```

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

```
planet = "Earth"
diameter = 12742
print('The diameter of {} is {} kilometers.'.
      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]
```

```
lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
lst[3][1][2][0]
```

```
'hello'
```

```
** Given this nest 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']}]}]}
```

```
d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
d['k1'][3]['tricky'][3]['target'][3]

'hello'
```

**** What is the main difference between a tuple and a list? ****

*#The list is dynamic, whereas the tuple has static characteristics.
This means that lists*

**** Create a function that grabs the email website domain from a string in the form: ****

user@domain.com

So for example, passing "user@domain.com" would return: domain.com

```
def domainGet(email):
    return email.split('@')[-1]
domainGet('user@domain.com')
```

'domain.com'

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

True

**** Create a function that counts the number of times the word "dog" occurs in a string. Again ignore edge cases. ****

```
def findDog(st):
    return 'dog' in st.lower().split()
findDog('Is there a dog here?')
```

```
File "<ipython-input-1-ce1248879a23>", line 2
    return 'dog' in st.lower().split()
    ^
```

IndentationError: expected an indented block

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.

```
def caught_speeding(speed, is_birthday):
```

```
    if is_birthday:
        speeding = speed - 5
    else:
        speeding = speed

    if speeding > 80:
        return 'Big Ticket'
    elif speeding > 60:
        return 'Small Ticket'
    else:
        return 'No Ticket'
```

```
def caught_speeding(speed, is_birthday):
```

```
if is_birthday:
    speeding = speed - 5
else:
    speeding = speed
if speeding > 80:
    return 'Big Ticket'
elif speeding > 60:
    return 'Small Ticket'
else:
    return 'No Ticket'
```

```
File "<ipython-input-2-825563dfa41a>", line 2
```

```
    if is_birthday:
    ^
```

IndentationError: expected an indented block

```
def caught_speeding(speed, is_birthday):
```

```
if is_birthday:
    speeding = speed - 5
else:
    speeding = speed
if speeding > 80:
    return 'Big Ticket'
elif speeding > 60:
```

```
    return 'Small Ticket'
else:
    return 'No Ticket'

'Small Ticket'
```

Create an employee list with basic salary values(at least 5 values for 5 employees) and using a for loop retrieve each employee salary and calculate total salary expenditure.

```
list = [50000,60000,55000,35000,40000]
tot_salary = 0
for salary in list:
    tot_salary += salary
print(salary)
print('Total Salary:',tot_salary)
```

Create two dictionaries in Python:

First one to contain fields as Empid, Empname, Basicpay

Second dictionary to contain fields as DeptName, DeptId.

Combine both dictionaries.

```
dict_1={'Empid': '22', 'Empname': 'IBM', 'Basicp': '25'}
dict_2={'DeptName': 'ECE', 'DeptId': 'ECEDA'}
dict_3=**dict_1,**dict_2
print(dict_3)
```