Assignment -3

Python Programming

Assignment Date	9 October 2022
Student Name	ARAVINTHRENGANATHAN S
Student Roll Number	913319104004
Maximum Marks	2 Marks

TASK :

QUESTION LINK

To write and execute given python problem

https://drive.google.com/file/d/1UnwWqw2PCFBDro1w0T__lvTKxX2dcsRL/view

SOLUTION

https://drive.google.com/file/d/1UnwWqw2PCFBDro1w0T__lvTKxX2dcsRL/view?usp=sharing

Exercises

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

What is 7 to the power of 4?

In [1]:

```
print(pow(7,4))
```

```
2401
```

Split this string:

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

into a list.

In [3]:

```
s = "Hi there Sam!"
h=s.split()
print(h)
```

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

In [4]:

```
y="Hi there dad!"
z=y.split()
print(z)
```

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

Given the variables:

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

Use .format() to print the following string:

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

In [5]:

```
planet = "Earth"
diameter = 12742
s="The diameter of {} is {} kilometers"
print(s.format(planet,diameter))
```

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

In [7]:

```
planet = "Earth"
diameter = 12742
k="The diameter of {} is {} kilometers"
print(k.format(planet,diameter))
```

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

Given this nested list, use indexing to grab the word "hello"

In [ ]:

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

In [8]:

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

```
['hello']
```

Given this nest dictionary grab the word "hello". Be prepared, this will be annoying/tricky

In [ ]:

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

In [9]:

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

```
hello
```

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

In [ ]:

```
Tuple are immutable but List are mutable
List operations are more error prone but Tuples operations are safe
```

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

In [10]:

```
def domain(text):
  f=text.split('@')
  print(f[-1])
```

In [11]:

```
domain('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.

In [12]:

```
def findDog(text):
  if 'dog' in text.lower():
    print("True")
  else:
    print("False")
```

In [13]:

```
findDog("my dog is pretty")
```

True

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

In [18]:

```
def dogcount(value):
  count=0
  for word in value.lower().split():
    if word=='dog' or word=='dogs':
      count=count+1
  print(count)
```

In [20]:

```
dogcount("It's not just a dog , dog is like a friend")
```

2

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 [23]:

```
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'
```

In [25]:

```
caught_speeding(90,False)
```

Out[25]:

'Big Ticket'

In [27]:

```
caught_speeding(85,True)
```

Out[27]:

'Small Ticket'

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

In [28]:

```
employee=[300,270,530,170,1050]
sum=0
print("salary of First employer is",employee[0])
print("salary of Second employer is",employee[1])
```

```
print("salary of Third employer is",employee[2])
print("salary of Fourth employer is",employee[3])
print("salary of Fifth employer is",employee[4])
for g in employee:
  sum=sum+g
print("The total salary of employee is",sum)
```

```
salary of First employer is 300
salary of Second employer is 270
salary of Third employer is 530
salary of Fourth employer is 170
salary of Fifth employer is 1050
The total salary of employee is 2320
```

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.

In [30]:

```
dictionary1={'Empid':458,'Empname':"vijk",'Basicpay':5380}
dictionary2={'DeptName':"Infotech",'DeptId':362}
print({**dictionary1 ,**dictionary2})
```

```
{'Empid': 458, 'Empname': 'vijk', 'Basicpay': 5380, 'DeptName': 'Infotech', 'DeptId': 362
}
```