Assignment -3

Student Name	KAMALIE K
College Roll Number	19TUCS128
Maximum Marks	2

Exercises

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

1. What is 7 to the power of 4?

In [3]:
```
7**4
```

Out[3]:
```
2401
```

1. Split this string:

   s = "Hi there Sam!"

into a list.

In [3]:
```
S='Hi there Sam!'
```

In [4]:
```
S.split()
```

Out[4]:
```
['Hi', 'there', 'Sam!']
```

1. Given the variables:

   planet = "Earth" diameter = 12742

   Use .format() to print the following string:

   The diameter of Earth is 12742 kilometers.

In [2]:
```
planet="Earth"
```

In [7]:
```
diameter=12742
print('The diameter of {} is {} kilometers.' .format(planet,diameter));
```

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

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

In [8]:
```
lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
```

In [9]:
```
lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
a=lst[3][1][2]
print(a)
```

```
['hello']
```

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

In [10]:
```
d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
```

In [11]:
```
print(d['k1'][3]["tricky"][3]['target'][3])
```

```
hello
```

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

In [ ]:
```

```

Tuples are immutable whereas Lists are mutable. Tuples consumes less memory whereas Lists consume more memory. Tuples does not have many built-in methods whereas Lists have several built-in methods.

1. 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 [12]:
```
def domainGet(email):
  print("Your domain is: "+email.split('@')[-1])
email=input("Please enter your email")
domainGet(email)
```

```
Please enter your email:sreedhar23@gmail.com
Your domain is: gmail.com
```

1. 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 [14]:
```
def finddog(st):
  if 'dog' in st.lower():
    print("True")
  else:
    print("False")
st="is there a dog here?"
finddog(st)
```

```
True
```

In [15]:
```
finddog("is there a dog here?")
```

```
True
```

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

In [19]:
```
def countdogs(value):
  cnt=0;
  for word in value.lower().split():
    if word == 'dog' or word == 'dogs':
      cnt+=1
  print(cnt)
value='this dog run faster than the other dogs'
countdogs(value)
```

```
2
```

Problem

1. 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 [20]:
```
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 [21]:
```
caught_speeding(90,True)
```

Out[21]:
```
'Big Ticket'
```

In [22]:
```
caught_speeding(61,False)
```

Out[22]:
```
'Small Ticket'
```

1. 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 [23]:
```
employee = [15000,20000,25000,30000,40000]
sum=0
for i in employee:
  sum+=i
  print(i)
print(sum)
```

```
15000
20000
25000
30000
40000
130000
```

1. 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]:
```
dict_1={"Empid":"1","Empname":"sree","Basicpay":"40000"}
dict_2={"DeptName":"Developer","DeptId":"1001"}
dict_3={**dict_1, **dict_2}
print(dict_3)
```

```
{'Empid': '1', 'Empname': 'sree', 'Basicpay': '40000', 'DeptName': 'Developer', 'DeptId': '1001'}
```