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

Out[ ]:
```
2401
```

Split this string:

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

into a list.

In [ ]:
```
s = "Hi there Sam!"
d=s.split(' ')
print(d)
```

In [ ]:
```
# # method 1 -- user input
# s=input("enter the input :").split() #input-- Hi there dad!
# print(s)
#method 2 -- manual input
s= "Hi there dad!"
x=s.split()
print(x)
```

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

In [ ]:
```

```

```
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 [ ]:
```
x=lst[3][1][2]
print(*x)
```

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

Out[ ]:
```
'hello'
```

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

Tuple:

-- Enclosed with round brackets.

-- immutable

List:

-- Enclosed with square brackets.

-- mutable

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 [ ]:
```
d="user@domain.com"
def fun(d):
  a=""
  res= False
  for i in d:
    if(res):
      a+=i
  if(i=="@"):
    res = True
  return a

print(fun(d))
```

```
'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 [ ]:
```
def word_checker(sentence,word):
 s=sentence.split(" ")
 for i in s:
   if(i==word):
    return True
   return False
sentence="My Dog name is Dora"
Target_word = "Dog"
if(word_checker(sentence,Target_word)):
 print("True")
else:
 print("False")
```

```
True
```

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

In [ ]:
```
def word_checker(sentence, Target_word):
    a = sentence.split(" ")
    c = 0
    for i in range(0, len(a)):
       if (Target_word== a[i]):
          c = c+ 1
    return c
sentence="Dog is a pet and i love Dog"
Target_world="Dog"
print(word_checker(sentence,Target_word))
```

```
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 [ ]:
```
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'
```

```
'Big Ticket'
```

In [ ]:
```
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'
print(caught_speeding(71,False))
```

Out[ ]:
```
'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 [ ]:
```
employees = ["Dharshan","Pavi", "Abisha", "Dhayalini", "Savitha"]
salary={}
for emp in employees:
 amount=int(input(f'Enter salary for {emp}: '))
 salary[emp]=amount
print('Total salary ', sum(salary.values()))
```

```
Enter salary for Dharshan: 10000

Enter salary for Pavi: 10000

Enter salary for Abisha: 1200

Enter salary for Dhayalini: 3030

Enter salary for Savitha: 3333

Total salary 27563
```

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 [ ]:
```
dict1={'Empid' : 401, 'Empname':"Dharshan","Basicpay": 10000}
dict2= {'DeptName': "Excecutive", 'Deptid': 12345}
def checker(dict1,dict2):
 result=dict1.copy()
 result.update(dict2)
 return result
result=checker(dict1,dict2)
print(z)
```

```
{'Empid': 401, 'Empname': 'Dharshan', 'Basicpay': 10000, 'DeptName': 'Excecutive', 'Deptid': 12345}
```