

**Assignment -3**  
Python Programming

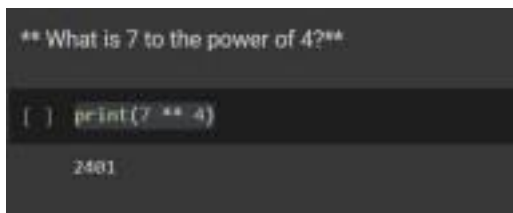
Assignment Date	7 October 2022
Student Name	Sarath S
Student Roll Number	723719106024
Maximum Marks	2 Marks

**Question-1:**

What is 7 to the power of 4?

Solution: `print(7 **`

`4)`



```
** What is 7 to the power of 4? **  
[ ] print(7 ** 4)  
2401
```

**Question-2:**

Split this string: `s =`

`"Hi there Sam!"`

into a list.

Solution: `s = "Hi there  
Sam!" x=s.split() print(x)`

Output:

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

```
** Split this string **

s = "Hi there Sam!"

*into a list.*

[ ] s = "Hi there Sam!"
    s.split()
    print(x)

Show hidden output

[ ] s = "Hi there dad!"
    s.split()
    print(x)

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

**Question-3:** Given

the variables:

planet = "Earth"

diameter = 12742

Use .format() to print the following string:

The diameter of Earth is 12742 kilometers.

Solution:

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

 Output:

The diameter of Earth is 12742 kilometers.

```
** 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

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

The diameter of Earth is 12742 kilometers.
```

**Question-4:**

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

Solution:

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

```
print(lst[3][1][2])
```

Output:

```
['hello']
```

```
** 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]
    print(lst[3][1][2])

['hello']
```

### Question-5:

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

{'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']}]}]}
print(d['k1'][3]['tricky'][3]['target'][3])
```

Output:

```
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']}]}]}
    print(d['k1'][3]['tricky'][3]['target'][3])

hello
```

### Question-6:

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

#Tuples are immutable.

#Lists consume more memory.

#Tuple iterations are faster

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

[ ] #Tuples are immutable.
    #Lists consume more memory.
    #Tuple iterations are faster
```

### Question-7:

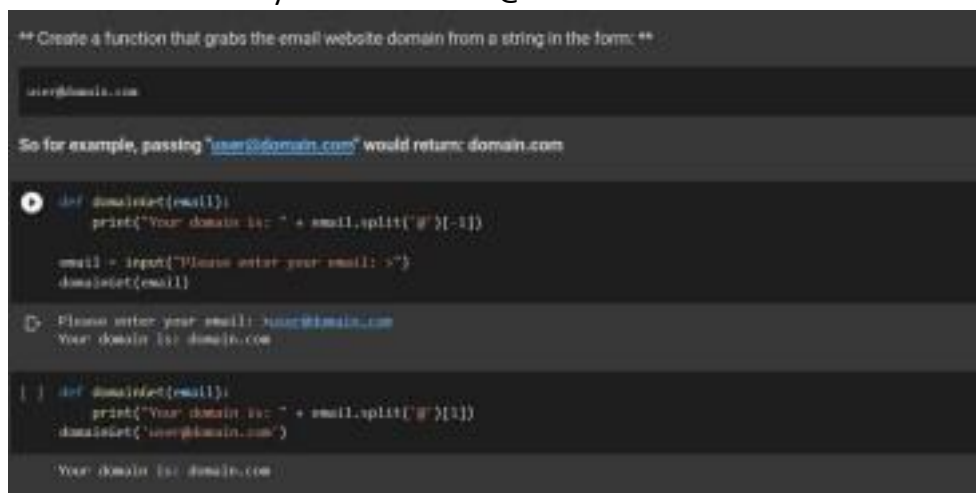
Create a function that grabs the email website domain from a string in the form [user@domain.com](#)

Solution:

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

Output:

Please enter your email: >user@domain.com Your domain is: domain.com



The screenshot shows a Jupyter Notebook with the following content:

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

### Question-8:

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

Solution: def

```
func(animal):  
    if 'dog' in animal.lower():  
        print('True')  
    else:  
        print('False')  
  
func('A dog is in the house')
```

Output:

True

```
** 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. **

[ ] def func(animals):
    if 'dog' in animals.lower():
        print('True')
    else:
        print('False')
    func('A dog is in the house')

True

[ ] x='That is a dog'
func(x)

True
```

### Question-9:

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

Solution:

```
x='I have a dog named Dog who likes the dog next door.'
```

```
def cntno(st) :

    count=0

    for txt in st.lower().split():

        if txt == 'dog' or txt == 'dogs':

            count = count + 1

    print(count)

cntno(x)
```

Output:

3

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

[ ] x='I have a dog named Dog who likes the dog next door.'
def cntno(st) :
    count=0
    for txt in st.lower().split():
        if txt == 'dog' or txt == 'dogs':
            count = count + 1
    print(count)
    cntno(x)

1

[ ] cntno('My dog looks cuter than his dog')

2
```

### Question-10:

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'

Solution:

    caught\_speeding(81,False)

    caught\_speeding(85,True)

Output:

    Big Ticket

    Small Ticket

*"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'
```

```
[ ] caught_speeding(81,False)
```

    'Big Ticket'

```
[ ] caught_speeding(85,True)
```

    'Small Ticket'

### Question-11:

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.

Solution:

```
empsal=[50000,51000,60000,65000,66000]
totsal=0
for x in empsal :
    print(x)
    totsals=totsal+x
print(totsal)
```

Output:

```
50000
51000
60000
65000
66000
292000
```

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.

```
[ ] empsal=[50000,51000,60000,65000,66000]
    totsals=0
    for x in empsal :
        print(x)
        totsals=totsal+x
    print(totsal)

50000
51000
60000
65000
66000
292000
```

### Question-12:

Create two dictionaries in Python: First one to contain fields as Empid, Empname, BasicpaySecond dictionary to contain fields as DeptName, DeptId. Combine both dictionaries  
Solution:

```
d1={'Empid': 123, 'Empname': 'Sam', 'Basicpay':98000}
d2={'DeptName': 'Marketing','DeptId': '10'}
d1.update(d2)
print(d1).
```

Output:

{'Empid': 123, 'Empname': 'Sam', 'Basicpay': 98000, 'DeptName': 'Marketing', 'DeptId': '10'}

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.

```
[ ] d1={'Empid': 123, 'Empname': 'Sam', 'Basicpay': 98000}
    d2={'DeptName': 'Marketing', 'DeptId': '10'}
    d1.update(d2)
    print(d1)
```

```
{'Empid': 123, 'Empname': 'Sam', 'Basicpay': 98000, 'DeptName': 'Marketing', 'DeptId': '10'}
```