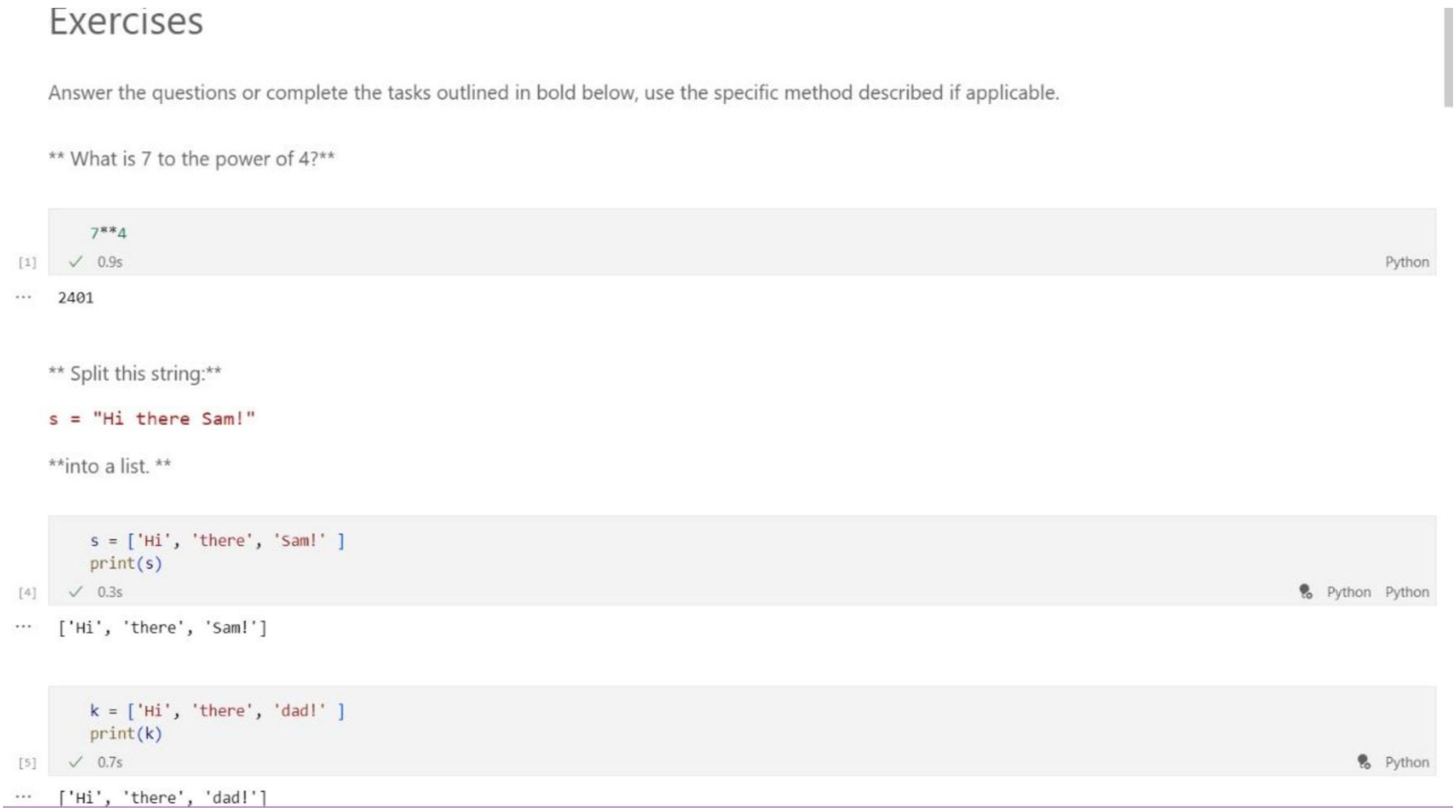


Exercises

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?**

```
7**4
```

```
2401
```

** Split this string:**

`s = "Hi there Sam!"`

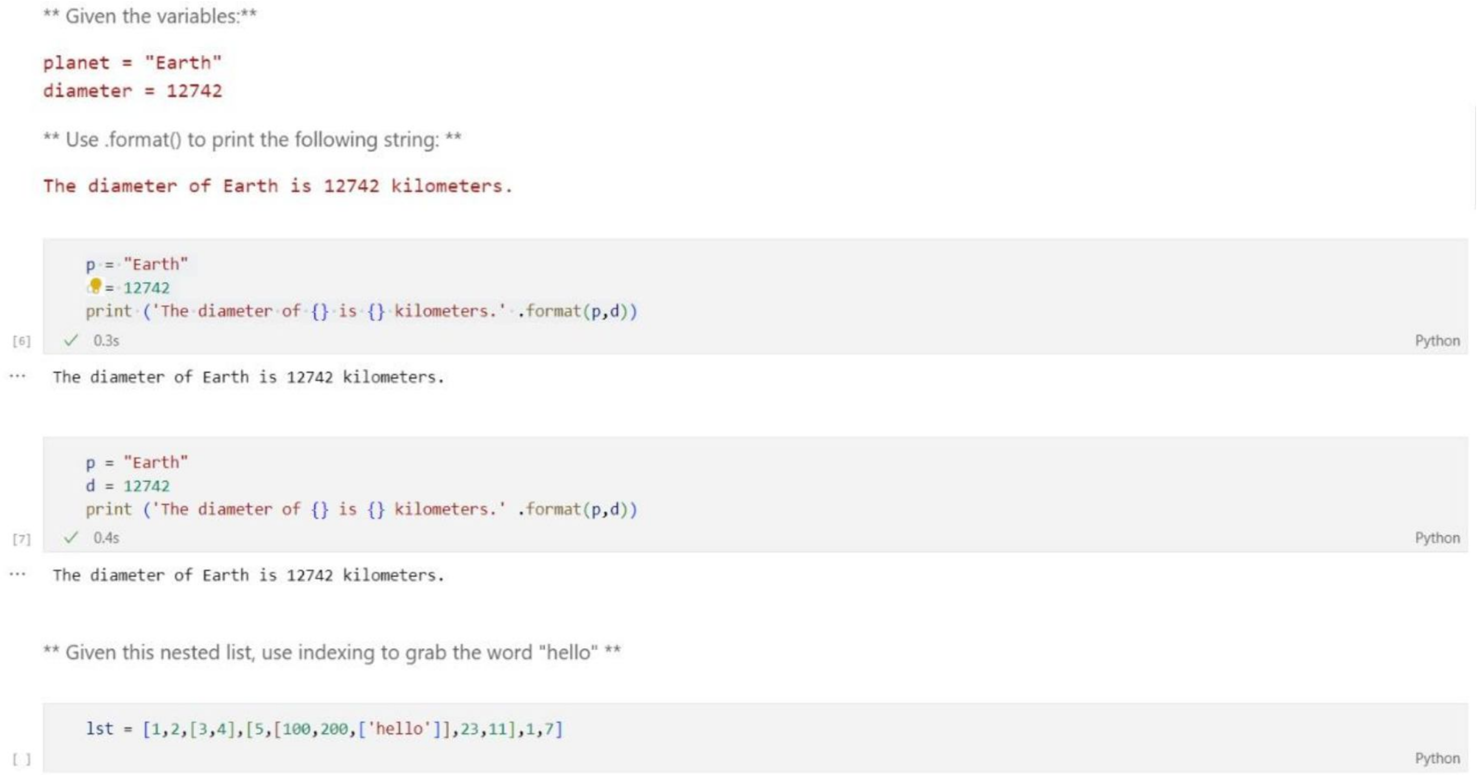
**into a list. **

```
s = ['Hi', 'there', 'Sam!' ]
print(s)
```

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

```
k = ['Hi', 'there', 'dad!' ]
print(k)
```

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



** Given the variables:**

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

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

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

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

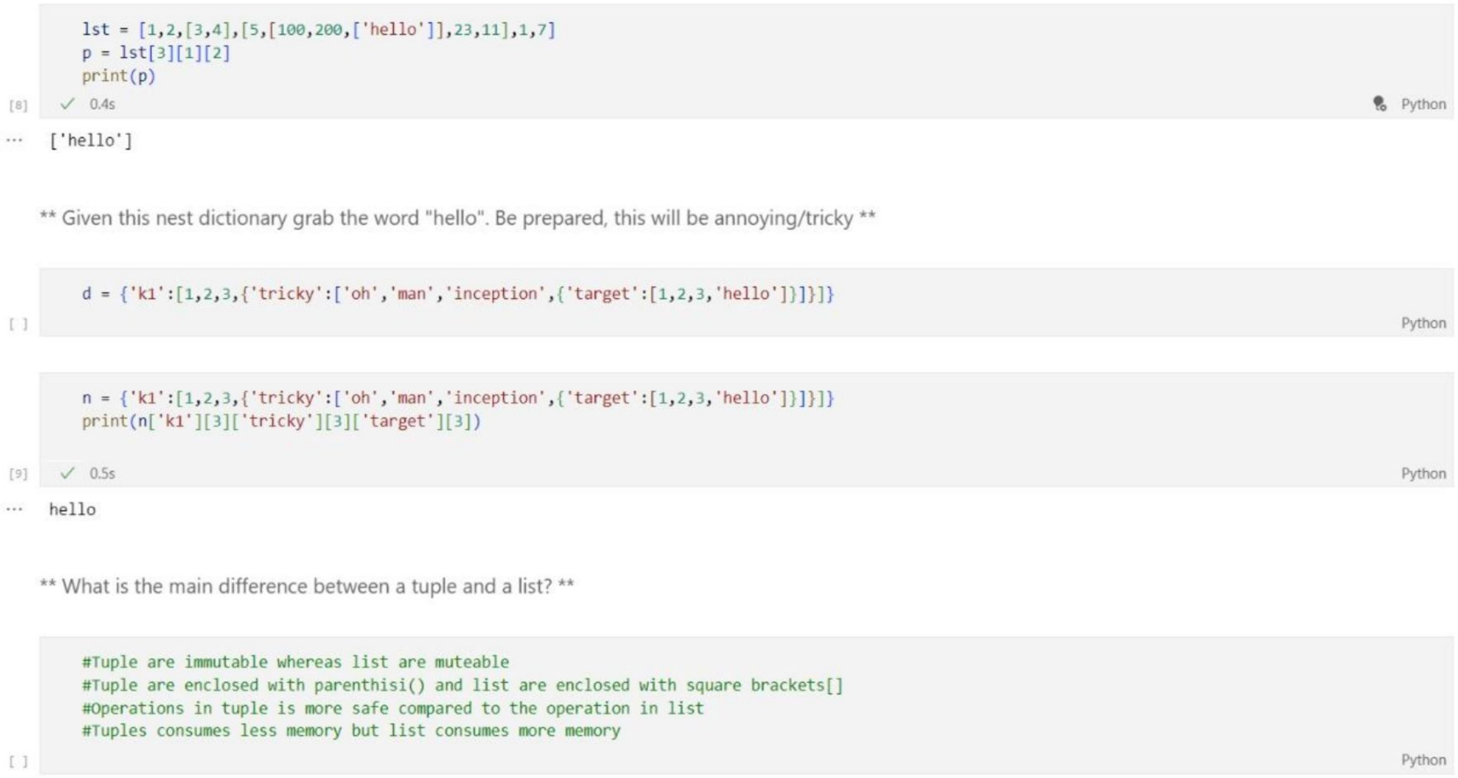
```
The diameter of Earth is 12742 kilometers.
```

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

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

```
['hello']
```

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

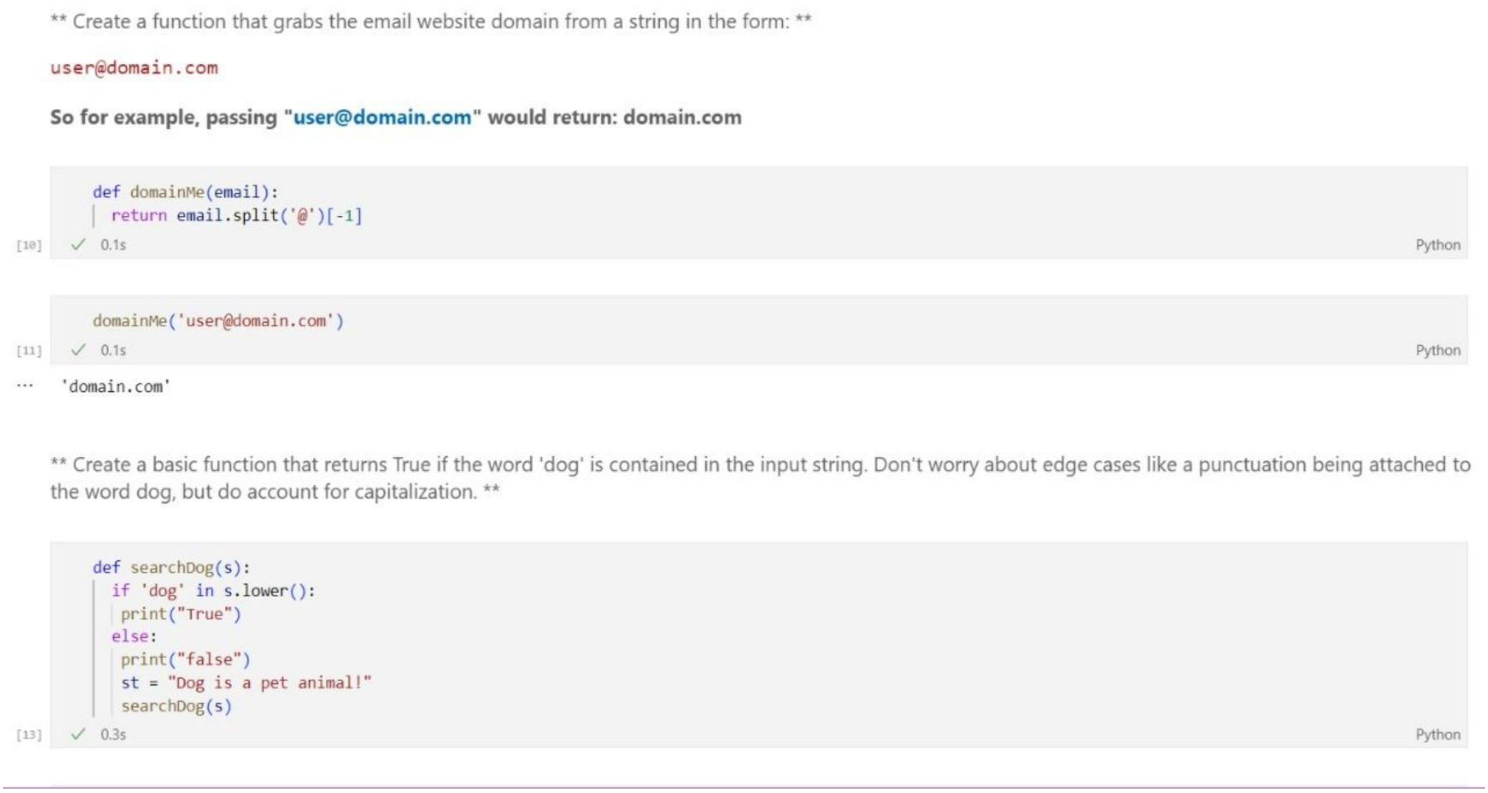
```
d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
```

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

```
hello
```

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

```
#Tuple are immutable whereas list are muteable
#Tuple are enclosed with parenthisi() and list are enclosed with square brackets[]
#Operations in tuple is more safe compared to the operation in list
#Tuples consumes less memory but list consumes more memory
```



** 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

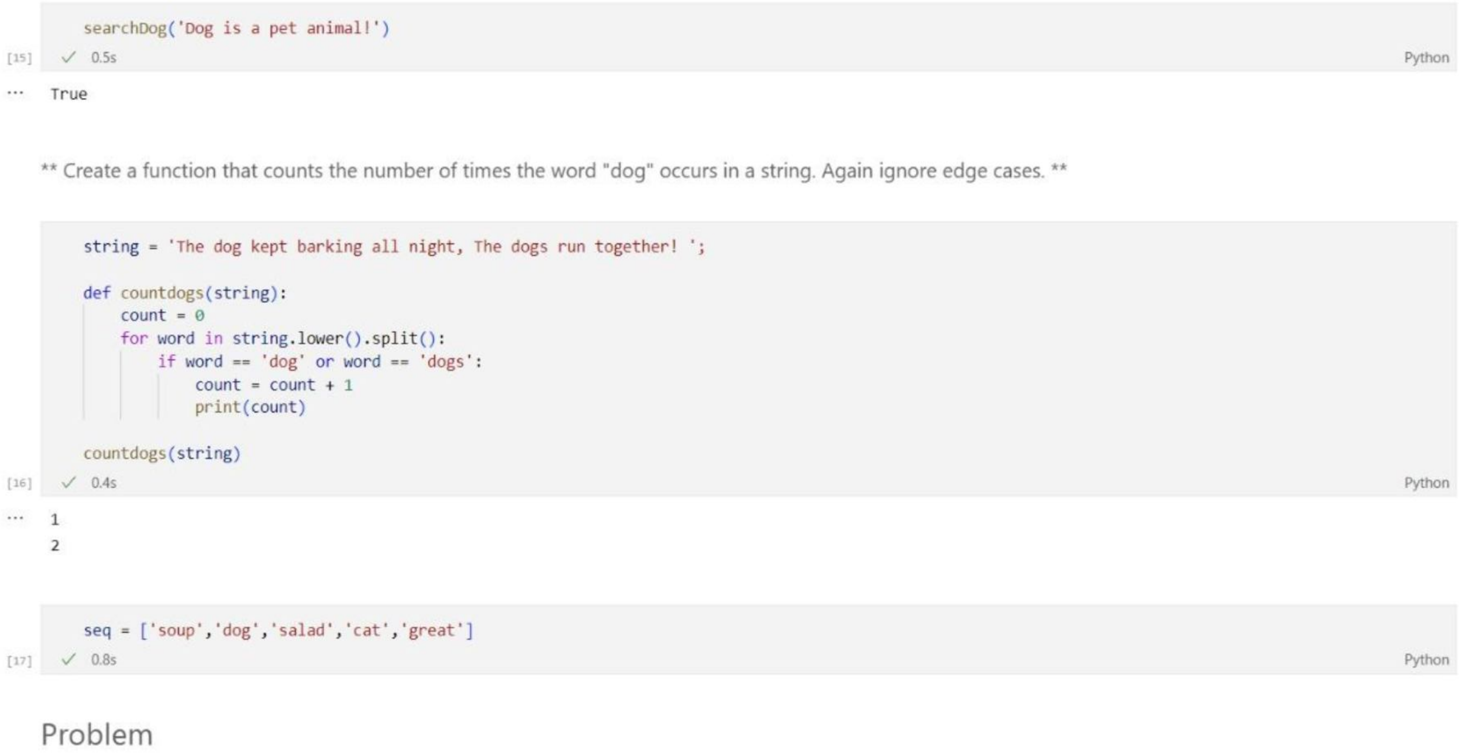
```
def domainMe(email):
    return email.split('@')[-1]
```

```
domainMe('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. **

```
def searchDog(s):
    if 'dog' in s.lower():
        print("True")
    else:
        print("false")
        st = "Dog is a pet animal!"
        searchDog(s)
```



```
searchDog('Dog is a pet animal!')
```

```
True
```

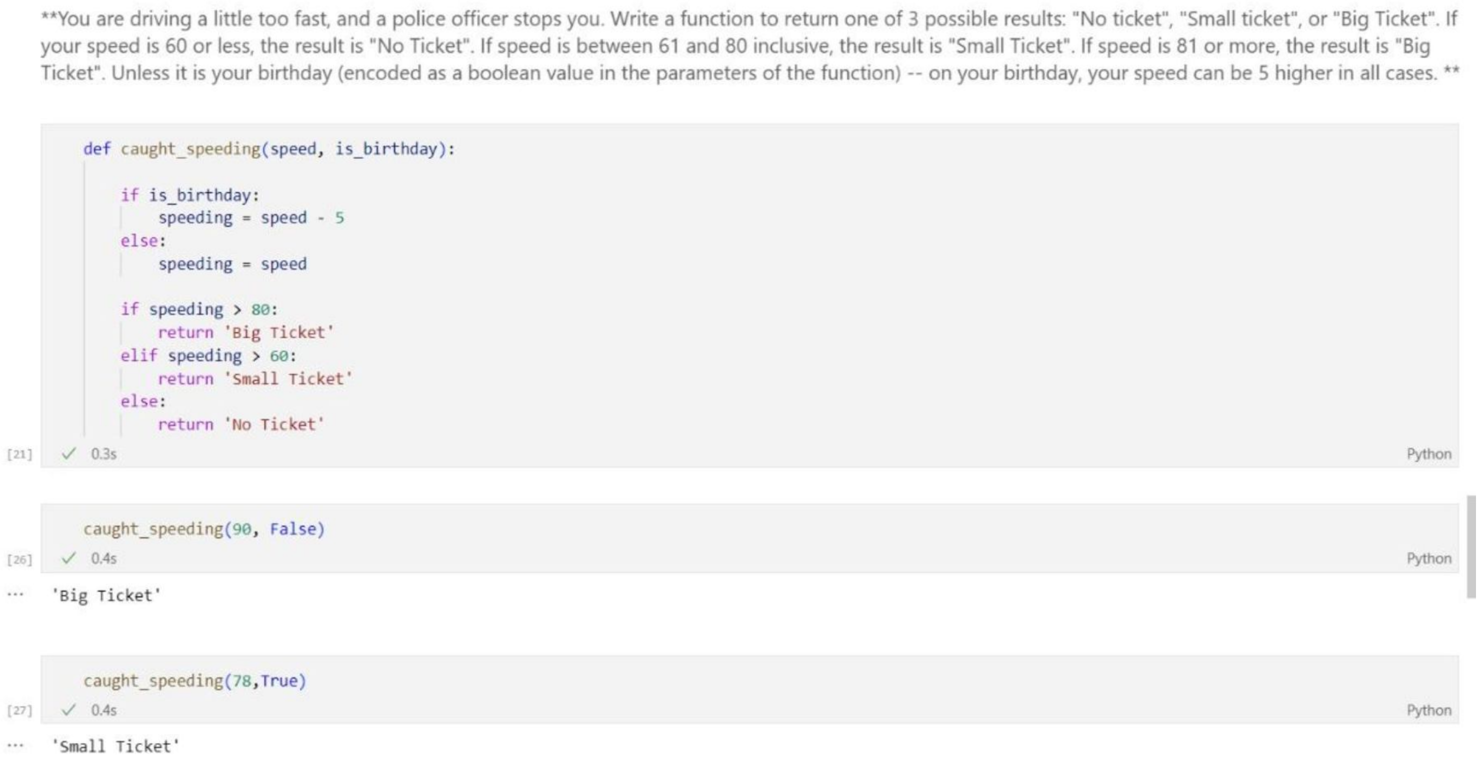
** Create a function that counts the number of times the word "dog" occurs in a string. Again ignore edge cases. **

```
string = 'The dog kept barking all night, The dogs run together! ';

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

countdogs(string)
```

```
1
2
```

```
seq = ['soup','dog','salad','cat','great']
```

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'
```

[21] ✓ 0.3s Python

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

[26] ✓ 0.4s Python

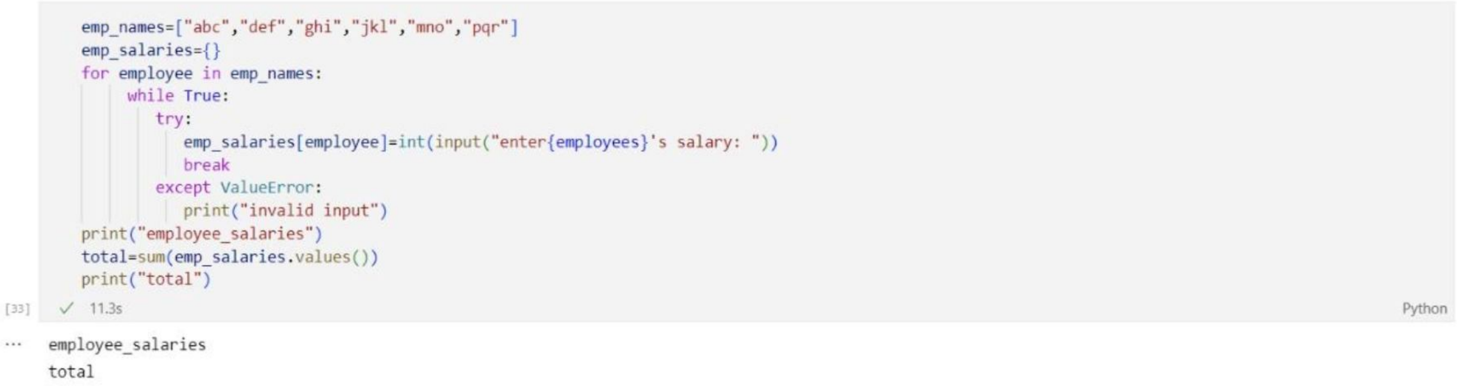
```
··· 'Big Ticket'
```

```
caught_speeding(78,True)
```

[27] ✓ 0.4s Python

```
··· '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.



```
emp_names=["abc","def","ghi","jkl","mno","pqr"]
emp_salaries={}
for employee in emp_names:
    while True:
        try:
            emp_salaries[employee]=int(input("enter{employees}'s salary: "))
            break
        except ValueError:
            print("invalid input")
print("employee_salaries")
total=sum(emp_salaries.values())
print("total")
```

[33] ✓ 11.3s Python

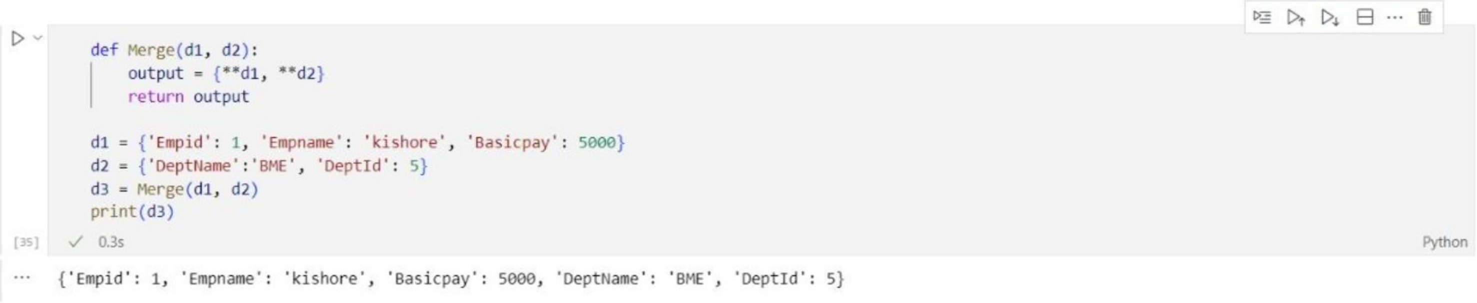
```
employee_salaries
total
```

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.



```python
def Merge(d1, d2):
    output = {**d1, **d2}
    return output

d1 = {'Empid': 1, 'Empname': 'kishore', 'Basicpay': 5000}
d2 = {'DeptName':'BME', 'DeptId': 5}
d3 = Merge(d1, d2)
print(d3)
```

```
{'Empid': 1, 'Empname': 'kishore', 'Basicpay': 5000, 'DeptName': 'BME', 'DeptId': 5}
```