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 [22]:
```python
7**4
```

Out[22]: 2401

** Split this string:**

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

into a list.

In [19]:
```python
s = "Hi there Sam!"
```

In [21]:
```python
s.split()
```

Out[21]: ['Hi', 'there', 'Sam!']

** Given the variables:**

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

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

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

In [17]:
```python
planet="Earth"
```

In [18]:
```python
diameter=12742
print('The diameter of {} is {} kilometers.' .format(planet,diameter));
```

The diameter of Earth is 12742 kilometers.

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

In [14]:
```python
lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
```

In [15]:
```python
lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
a=lst[3][1][2]
```

```
print(a)
```

```
['hello']
```

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

In [12]:
```python
d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hell
```

In [13]:
```python
print(d['k1'][3]["tricky"][3]['target'][3])
```

```
hello
```

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

In [ ]:
```
Tuples are immutable whereas Lists are mutable. Tuples consumes less memo
```

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

```
Please enter your emailteam6@gmail.com
Your domain is: gmail.com
```

In [ ]:

Out[ ]:   '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 [2]:
```python
def finddog(st):
    if 'dog' in st.lower():
        print("True")
    else:
        print("False")
st="is there a dog here?"
finddog(st)
```

```
True
```

In [3]:
```python
finddog("is there a dog here?")
```

True

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

In [11]:
```python
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

In [ ]:

Out[ ]: 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 [8]:
```python
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 [9]:
```python
caught_speeding(90,True)
```

Out[9]: 'Big Ticket'

In [10]:
```python
caught_speeding(61,False)
```

Out[10]: '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 [4]:
```python
employee = [15000,20000,25000,30000,40000]
sum=0
for i in employee:
  sum+=i
  print(i)
print(sum)
```

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

Create two dictionaries in Python: