

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. **

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

```
print(d)
```

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

**** Given the variables: ****

```
plane
```

```
t =
```

```
"Eart
```

```
h"
```

```
diame
```

```
ter =
```

```
12742
```

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

The diameter of Earth is 12742 kilometers.

```
planet
=
"Earth
"
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" ****

```
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]a = lst[3][1][2];
print(a)
```

```
['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
```

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

```
#Tuple are immutable and list are mutable
#Tuple are enclosed with() and list are
enclosed with[]#Tuples operations are safe
and List are less safe
#Tuples consumes less memory and 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 [@
HYPERLINK "mailto:user@domain.com"domain.com](mailto: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 findDog(st):
    if 'dog' in
        st.lower():
        print("True
")
```

```

else:
    print("false")

st = "Is there is a
dog here?"
findDog(st)

True

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. ****

```

value = 'This dog runs faster than the other
dog dude!';def countdog(value):

```

```

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

```

```

countdog(value)

```

```

1
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. **

```
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(81, True)
```

```
    'Small Ticket'
```

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.

```
#we should the data set
```

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.

#datasets required

✓●✗[Colab](#) HYPERLINK

["https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links"](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links) HYPERLINK

["https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links"](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links) paid HYPERLINK

["https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links"](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links) HYPERLINK

["https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links"](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links) products - Cancel HYPERLINK

["https://colab.research.google.com/cancel-subscription"](https://colab.research.google.com/cancel-subscription)

HYPERLINK

"https://colab.research.google.
com/cancel-
subscription"contracts

HYPERLINK

"https://colab.research.google.co
m/cancel-subscription"

HYPERLINK

"https://colab.research.google.
com/cancel-subscription"here

0s completed at 11:31 PM