

40731123 C VENKATA NAGA SRIHARSHA

what is 7 to the power of 4

In [1]:

```
7**4
```

Out[1]:

2401

Split this string into a list.

In [3]:

```
s = "Hi there sam!"
```

In [4]:

```
s.split()
```

Out[4]:

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

Given the variables planet = "Earth" diameter = 12742 use. format() to print the following string the diameter of earth is 12742 kilometers

In [6]:

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

In [7]:

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

The diameter of Earth is 12742 kilometers

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

In [9]:

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

In [10]:

```
lst[3][1][2][0]
```

Out[10]:

```
'hello'
```

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

In [12]:

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

In [13]:

```
d['k1'][3]['tricky'][3]['target'][3]
```

Out[13]:

```
'hello'
```

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

In [16]:

```
"Tupple is immutable"
```

Out[16]:

```
'Tupple is immutable'
```

Create a function that grabs the email website domain from a string in the form:

In [20]:

```
user@domain.com
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-20-ec340c6c4788> in <module>
----> 1 user@domain.com
```

**NameError:** name 'user' is not defined

So for example, passing “[user@domain.com \(mailto:user@domain.com\)](mailto:user@domain.com)” would return: domain.com

In [21]:

```
def getDomain(s):
    return s.split('@')[1]
```

In [22]:

```
getDomain("user@domain.com")
```

Out[22]:

```
'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 [23]:

```
def findDog(s):  
    return "dog" in s.lower()
```

In [24]:

```
findDog("Is there any Dog there ?")
```

Out[24]:

True

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

In [25]:

```
def countDog(s):  
    count = 0  
    for word in s.lower().split():  
        if "dog" in word:  
            count += 1  
    return count
```

In [26]:

```
countDog("Those doggies is dodging me")
```

Out[26]:

1

Use lambda expressions and the filter() function to filter out words from a list that don't start with the letter 's'.  
For example:

In [33]:

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

In [34]:

```
['soup', 'salad']
```

Out[34]:

```
['soup', 'salad']
```

In [35]:

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

In [36]:

```
list(filter(lambda s: s[0]!='s', seq))
```

Out[36]:

```
['soup', 'salad']
```

You are driving a little too fast, and a police officer stops you. Write a function to return one of the 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 [37]:

```
def ticket(speed, birthday):  
    if birthday is True:  
        speed -= 5  
    if speed > 80:  
        return "Big Ticket"  
    elif speed >60:  
        return "Small Ticket"  
    else :  
        return "No Ticket"
```

In [38]:

```
ticket(66, True)
```

Out[38]:

```
'Small Ticket'
```

In [39]:

```
ticket(81, False)
```

Out[39]:

```
'Big Ticket'
```

In [ ]: