Numbers

Write an equation that uses multiplication, division, an exponent, addition, and subtraction that is equal to 100.25.

Hint: This is just to test your memory of the basic arithmetic commands, work backwards from 100.25

```
X = ((((10 ** 3)*2) - 1500)/5)+0.25
print (X)
```

Answer these 3 questions without typing code. Then type code to check your answer.

What is the value of the expression 4 * (6 + 5)

What is the value of the expression 4 * 6 + 5

What is the value of the expression 4 + 6 * 5

44

```
4 * (6 + 5)
```

29

```
4 * 6 + 5
```

34

```
4 + 6 * 5
```

What is the type of the result of the expression 3 + 1.5 + 4?

Float

What would you use to find a number's square root, as well as its square?

sqrt(), n ** 2

BLACKJACK:

Given three integers between 1 and 11, if their sum is less than or equal to 21, return their sum. If their sum exceeds 21 and there's an eleven, reduce the total sum by 10. Finally, if the sum (even after adjustment) exceeds 21, return 'BUST'

```
def blackjack(a,b,c):
    m = a+b+c
    if m <= 21:
        return m
    elif m > 21 and (a ==11 or b ==11 or c==11):
```

```
        if m-10 >21:
            return 'BUST'
        else:
            return m-10

#blackjack(5,6,7)
#blackjack(9,9,9)
blackjack(9,9,11)
```

SUMMER OF '69: Return the sum of the numbers in the array, except ignore sections of numbers starting with a 6 and extending to the next 9 (every 6 will be followed by at least one 9). Return 0 for no numbers.

```
def summer_of_69(n):
    sum = 0
    flag = False
    for item in n:
        if item==9:
            sum = sum +0
            flag = True
        elif item==6 or flag==True:
            flag = True
            sum = sum +0
        else:
            flag = False
            sum = sum+item
    return sum

summer_of_69([1, 3, 5])

summer_of_69([4, 5, 6, 7, 8, 9])

summer_of_69([2, 1, 6, 9, 11])
```

CHALLENGING PROBLEMS

SPY GAME: Write a function that takes in a list of integers and returns True if it contains 007 in order

```
def spy_game(a):
    i=0
    for _ in a:
        if a[i]==0:
            if a[i+1]==0:
                if a[i+2]==7:
                    i +=1
```

```
                    return True
        else:
            i +=1
    return False

spy_game([1,2,4,0,0,7,5])


spy_game([1,0,2,4,0,5,7])

spy_game([1,7,2,0,4,5,0])
```

syntax of map: First thing --> we should have a function. def square(num): return num**2

my_nums = [1,5,8,9,8]

for item in map(square,my_nums): print(item)

[::-1] This is the python way of reversing strings, list and so on.

Write a Python function that accepts a string and calculates the number of upper case letters and lower case letters.

Write a python func. that checks whether a string is pangram or not.

```
import string

def is_pangram(s):
    a = s.replace(" ",'')
    a = a.lower()
    z = set(string.ascii_lowercase)
    a = set(a)
    return a == z


is_pangram("The quick brown fox jumps over the lazy dog")
```

```
def clear_list():
    for x in range(len(list1)):
        list1[x] = [' ']

list1 = [' ',' ' ,' ',' ',' ' ,' ',' ',' ' ,' ',' ']

def display_game():


    print(list1[7], end = ' ')
```

```
    print(list1[8], end = ' ')
    print(list1[9])
    print(list1[4], end = ' ')
    print(list1[5], end = ' ')
    print(list1[6])
    print(list1[1], end = ' ')
    print(list1[2], end = ' ')
    print(list1[3])
    continue_pl()
```

```
def choose_pos():
    posi = ['1','2','3','4','5','6','7','8','9']
    s = input("enter your position: ")

    while s not in posi:
        s= input("you've entered an invalid number, please choose
again: ")
    replace_function(s)
```

```
def replace_function(sttr):
    a = int(sttr)
    d = input("choose your symbol, X or O: ")
    list1[a] = d
    display_game()
```

```
#def continue_pl():
#    d = "False"
#    while d!="True":
#        d = input("do you want to keep playing? , Y or N: ")
#        if d=="Y":
#            choose_pos()
#        elif d =="N":
#            d= "True"
#            clear_list()
#            from IPython.display import clear_output
```

```
def continue_pl():
    d = False
    while (d!=True):
        d = input("do you want to keep playing? , Y or N: ")
        if d == "Y":
            choose_pos()
        elif d =="N":
            d = True
            clear_list()
            from IPython.display import clear_output


print("Who would like to go first? ")
choose_pos()
```

```
Who would like to go first?
enter your position: 7
choose your symbol, X or O: X
X
```