USEFUL FUNCTIONS

(1) RANGE

→ This function allows you quickly to generate a list of integers.

Syntax:

range(start, stop, step)

Note: range() is a generator func^n, so to get a list out of it we need to ~~use~~ cast it to a list.
Generator is a special type of function that would generate info. & no need to save it on memory

For e.g.

Qn: list(range(0,5))

O/P: [0, 1, 2, 3, 4]

Qn: list(range(0,11,2))

O/P: [0, 2, 4, 6, 8, 10]

(2) ENUMERATE

Used with for loops, used to keep track of index.

```
In: for i, letter in enumerate('abc')
        print("At index {} the letter is {}".
              format(i, letter))

O/P: At index 0 the letter is a
     ---11--- 1 ---11--- b
     ---11--- 2 ---11--- c
```

(3) ZIP

```
In: list(enumerate('abcde'))

O/P: [(0,'a'),(1,'b'),(2,'c'),(3,'d'),(4,'e')]
```

→ This is a list of tuples, hence we could use tuple unpacking during our for loop

→ In the same way we can create a list of 2 tuples by zipping 2 lists.

```
In: l1=[1,2,3,4]
    l2=['a','b','c','d']

In: zip(l1,l2)

O/P: <zip at 0x1d2058093f08>
```

eg: list(zip(l1, l2))

o/p: [(1, 'a'), (2, 'b'), (3, 'c'), (4, 'd')]

(4) in

Used to check if a particular item is there in an iterable object.

eg: 'x' in ['x', 'y', 'z']

o/p: True

eg: 'x' in [1, 2, 3]

o/p: False

(5) not in

Checks whether some object is not there

eg: 'x' not in ['x', 'y', 'z']

o/p: False

(6) min & max

returns min. & max. values in a iterable object

In: ll = [1, 2, 3, 4, 5]
In: min(ll)

O/P: 1

In: max(ll)
O/P: 5

random

In: from random import shuffle

In: shuffle(ll)

In: ll

O/P: [4, 1, 5, 3, 2]

In: from random import randint

In: randint(0, 100)

O/P: 25

input:

Takes input from user

qn: input('Enter something')

o/p: Enter something 123
'123'

LIST COMPREHENSION

It helps us to build our data using a diff. notation.
qn short,
It's a one line for loop built inside a bracket

E.g. (1)

qn: # Scale every letter on the string

qn: lst=[x for x in 'word']

qn: lst

o/p: ['w', 'o', 'r', 'd']

E.g. (2)
square nos in range & turn into a list

qn: lst=[x**2 for x in range(0,11)]

qn: lst

o/p: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

E.g ③ # Check for even nos in a range

```
In: lst=[x for x in range(11) if x%2==0]
In: lst
o/p: [0,2,4,6,8,10]
```

E.g. ④ Complex Arithmetic

Celsius to farenheit

```
In: celsius=[0,10,20.1,34.5]
    far=[((9/5)*temp+32) for temp in celsius]
    far
o/p: [32.0, 50.0, 68.18, 94.1]
```

E.g. ⑤ Nested Comprehension

```
In: lst=[x**2 for x in [x**2 for x in range(5)]]
o/p: [0, 1, 16, 81, 256]
```