Range function in Python

The range() function is a built in function in Python

It returns a sequence of numbers starting from zer and increment by 1 by default and stop before the given number

Now that we know the definition of range, lets see the syntax:

range(start, stop, step)

It has three paramters, in which two are optional:

start: Its an optional paramter used to define the starting point of the sequence. By default, it is zero

stop: Its a mandatory paramter, used to define the stopping point of the sequence

step: Its also an optional paramter used to specify the incrementation on each iteration; by default, the value is one.

General Usage: As it returns a sequence of numbers, most of the developers use the range to write loops. This comes handy when you dont have a list or tuple, but only a specific value to implement the loop

In [1]:

```
mylist =[1,2,3]
```

In [4]:

```
for num in range(10):
    print(num)
```

```
0
1
2
3
4
5
6
7
8
9
```

In [5]:

```
for num in range(3,10):
    print(num)
```

```
3
4
5
6
7
8
9
```

We can have a step size as well..like so:

In [7]:

```
for num in range(0,10,2):
    print(num)
```

Loading [MathJax]/extensions/Safe.js

```
0
2
4
6
8
```

In [9]:

```
range(0,10,2)
```

Out[9]:

```
range(0, 10, 2)
```

In [10]:

```
list(range(0,10,2))
```

Out[10]:

```
[0, 2, 4, 6, 8]
```

Enumerate() in Python

Often when dealing with iterators, we also get a need to keep a count of iterations. Python eases the programmers task by providing a built in function enumerate() for this task

Enumerate() method adds a counter to an interable and returns it in a form of enumerating object. This enumerated object can then be used directly for loops or converted into a list of tuples using the list() method

In [11]:

```
index_count = 0

for letter in 'abcde':
    print('At index {} the letter is {}' .format(index_count,letter))
    index_count += 1
```

```
At index 0 the letter is a
At index 1 the letter is b
At index 2 the letter is c
At index 3 the letter is d
At index 4 the letter is e
```

In [15]:

```
index_count = 0
word = 'abcde'

for letter in word:

    print(word[index_count])
    index_count+= 1
```

```
a
b
c
d
e
```

In [16]:

```
word = 'abcde'

for item in enumerate(word):
    print(item)
```

```
(0, 'a')
(1, 'b')
(2, 'c')
```

```
(3, 'd')
(4, 'e')
```

In [17]:

```
word = 'abcde'

for index,letter in enumerate(word):
    print(index)
    print(letter)
    print('\n')
```

```
0
a


1
b


2
c


3
d


4
e


```

Now, lets dicuss the zip() function in Python

Python zip() method takes iterable or containers and returns a single iterator object, having mappes values from all the containers

It is used to map the similar index of multiple containers so that they can be used just using a single entity.

In [18]:

```
mylist1 = [1,2,3,4,5]
mylist2 = ['a','b','c']
```

In [19]:

```
zip(mylist1,mylist2)
```

Out[19]:

```
<zip at 0x1fd52779840>
```

All it does is save as zip in our computers memory, we need to actually iterate through it like so:

In [20]:

```
for item in zip (mylist1,mylist2):
    print(item)
```

```
(1, 'a')
(2, 'b')
(3, 'c')
```

In [22]:

```
mylist1 = [1,2,3,4,5]
mylist2 = ['a','b','c']
mylist3 = [100,200,300]
```

```
In [24]: for item in zip (mylist1,mylist2,mylist3):
             print(item)
```

```
(1, 'a', 100)
(2, 'b', 200)
(3, 'c', 300)
```

```
In [25]: mylist1 = [1,2,3,4,5,6,7]
         mylist2 = ['a','b','c']
         mylist3 = [100,200,300]
```

```
In [26]: for item in zip (mylist1,mylist2,mylist3):
             print(item)
```

```
(1, 'a', 100)
(2, 'b', 200)
(3, 'c', 300)
```

Zip will only go as far as the shortest list

```
In [27]: list(zip(mylist1,mylist2))
```

```
Out[27]: [(1, 'a'), (2, 'b'), (3, 'c')]
```

In 'in' operator is used to check if a value exists in a sequence or not.

Evaluate to true if it finds a variable in the specified sequence and false otherwise

```
In [28]: 'x' in [1,2,3]
```

```
Out[28]: False
```

```
In [30]: 'x' in ['a','b','c']
```

```
Out[30]: False
```

```
In [31]: 'x' in ['y','z','x']
```

```
Out[31]: True
```

```
In [32]: 2 in [1,2,3]
```

```
Out[32]: True
```

```
In [33]: 'a' in 'a world'
```

```
Out[33]: True
```

```
In [34]: 'mykey' in{'mykey':345}
```

```
Out[34]: True
```

Loading [MathJax]/extensions/Safe.js

```
In [36]:  d = {'mykey':345}

          345 in d.values()

Out[36]:  True
```

```
In [3]:   d = {'mykey':345}

          345 in d.keys()

Out[3]:   False
```

Now, lets discuss min and max functions.

The max() function is used to find the largest value in a list of values. The min() function is used to find the lowest value in a list. The list of values can contain either strings or numbers.

You may encounter a situation where you want to find the minimum or maximum value in a list or a string. For instance, you may be writing a program that finds the most expensive car sold at your dealership.

In Python, you can use min() and max() to find the smallest and largest value, respectively, in a list or a string.

```
In [4]:   mylist = [10,20,30,40,100]
```

```
In [5]:   min(mylist)

Out[5]:   10
```

```
In [7]:   max(mylist)

Out[7]:   100
```

Now, lets import functions from a library:

After import hit TAB and you will see a list of functions to import.

```
In [10]:  from random import shuffle
```

```
In [11]:  mylist = [1,2,3,4,5,6,7,8,9,10]
```

```
In [12]:  shuffle(mylist)
```

```
In [13]:  mylist

Out[13]:  [3, 1, 4, 6, 8, 7, 2, 10, 5, 9]
```

```
In [14]:  shuffle(mylist)
```

Loading [MathJax]/extensions/Safe.js

```
In [15]:  mylist

Out[15]:  [2, 8, 4, 3, 6, 7, 10, 9, 5, 1]
```

We can also select a random integer like so:

```
In [17]:  from random import randint

In [18]:  randint(0,100)

Out[18]:  3

In [19]:  randint(0,100)

Out[19]:  81

In [21]:  mynumber = randint(0,10)

In [22]:  mynumber

Out[22]:  6
```

Now, lets see how we can accept user input, like so:

```
In [23]:  input('Enter a number here: ')

          Enter a number here: 50
Out[23]:  '50'
```

We can also save the user input, like so:

```
In [24]:  result = input('Enter a number here: ')

          Enter a number here: 50

In [25]:  result

Out[25]:  '50'

In [26]:  result = input('What is your name? ')

          What is your name? Keegan

In [27]:  result

Out[27]:  'Keegan'

In [28]:  result = input('Favorite Number: ')

          Favorite Number: 30
```

```
In [29]: result
```

```
Out[29]: '30'
```

Whatever is inputted, will be a string. We will need to set the data type like so:

```
In [30]: float(result)
```

```
Out[30]: 30.0
```

```
In [31]: int(result)
```

```
Out[31]: 30
```

```
In [ ]:
```