# Useful operators, List comprehensions and nested loops

January 2, 2022

## 1 Some useful operators in python

[1]:
```
# range()
```

[2]:
```
# A single argument can also be provided.
for num in range(10):
    print(num)
```

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

[3]:
```
for num in range(0,11):
    print(num)
```

```
0
1
2
3
4
5
6
7
8
9
10
```

[4]:
```
# step count can also be given in range

for num in range(0,11,2):
    print(num)
```

```
0
2
4
6
8
10
```

[5]:
```
# to cast to a list
list(range(0,11,2))
```

[5]:
```
[0, 2, 4, 6, 8, 10]
```

[6]:
```
index_count = 0
for letter in 'abcdef':
    print(f'at index {index_count} the letter is {letter}') # formatted string␣
 ↪literals
    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
at index 5 the letter is f
```

## 2 enumerate()

[9]:
```
word = 'abcde'
for index,letter in enumerate(word):
    print(index)
    print(letter)
    print('\n')
```

```
0
a


1
b


2
c


3
d
```

```
4
e
```

## 3 zip

[ ]:
```
# works in a sort of opposite way to enumerate
```

[11]:
```
mylist1 = [1,2,3]
mylist2 = ['a','b','c']
for item in zip(mylist1,mylist2):
    print(item)
```

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

[13]:
```
# if no of items in a list are uneven, the smallest list interms of items is␣
 ↪taken into consideration

mylist1 = [1,2,3,4,5]
mylist2 = ['a','b','c']
mylist3 = ['hi','hello']
for item in zip(mylist1,mylist2,mylist3):
    print(item)
```

```
(1, 'a', 'hi')
(2, 'b', 'hello')
```

## 4 Some boolean examples

[ ]:
```

```

[15]:
```
'x' in [1,2,3]
```

[15]:
```
False
```

[16]:
```
'x' in ['x','y','z']
```

[16]:
```
True
```

[17]:
```
'a' in 'a ward'
```

[17]:
```
True
```

[19]:
```
'k1' in {'k1': 2}
```

[19]: True

[21]:
```
d = {'key1': 2}
3 in d.values()
```

[21]: False

## 5 min and max in a list

[22]:
```
mylist = [10,20,30,40,50]
max(mylist)
```

[22]: 50

[24]:
```
min(mylist)
```

[24]: 10

## 6 List comprehensions

[25]:
```
# these are a unique way in creating lists
```

[26]:
```
mystring = 'harsha'
mylist = []
for letter in mystring:
    mylist.append(letter)
```

[27]:
```
mylist
```

[27]: ['h', 'a', 'r', 's', 'h', 'a']

[28]:
```
# we can also perform above operation using list comprehension as
mystring = 'harsha'
mylist = [letter for letter in mystring]
mylist
```

[28]: ['h', 'a', 'r', 's', 'h', 'a']

[30]:
```
mylist = [x for x in 'word']
mylist
```

[30]: ['w', 'o', 'r', 'd']

[31]:
```
mylist = [num for num in range(0,11)]
mylist
```

[31]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

[32]:
```
mylist = [num ** 2 for num in range(0,11)]
mylist
```

[32]: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

[33]:
```
# using if statement
mylist = [x for x in range(0,11) if x % 2 == 0]
mylist
```

[33]: [0, 2, 4, 6, 8, 10]

[34]:
```
# celsius to farenheit

celsius = [0,10,20,34.5]
farenheit = [((9/5) * temp + 32) for temp in celsius]
farenheit
```

[34]: [32.0, 50.0, 68.0, 94.1]

## 7 nested loops

[36]:
```
for x in [2,3,4]:
    for y in [5,6,7]:
        print(x * y)
```

```
10
12
14
15
18
21
20
24
28
```

[37]:
```
mylist = []
for x in [2,3,4]:
    for y in [5,6,7]:
        mylist.append(x * y)
mylist
```

[37]: [10, 12, 14, 15, 18, 21, 20, 24, 28]

[38]:
```
mylist = [x * y for x in [2,4,6] for y in [1,10,100] ]
mylist
```

[38]: [2, 20, 200, 4, 40, 400, 6, 60, 600]