- Files
- List Comprehension

List

It is collections non-homogenous data types of elements

1 - 100 elements into a list

In [1]:

```
li = []
for i in range(1,101):
    li.append(i)
print(li)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78,
79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99, 100]
```

In [2]:

```
li = [i for i in range(1,101)]
print(li)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78,
79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99, 100]
```

1. Dictionary Comprehension

2. Set Comprehensions

3. Modules and Packages

Dictionary Comprehension

if the we arrange the the data in the form key:value pairs we can that is dictionary

Syntax

```
{key:value for iter_variable in groupOfElements}
```

I want to create a dictionary with number as key and value as a square of the number

{1:1,2:4,3:9,4:16 .....}

In [4]:

```
squares = {}
for i in range(1,11):
    squares[i] = i**2
print(squares)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

In [5]:

```
square = {i:i**2 for i in range(1,10)}
print(square)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
```

I want to create a dictionary with even number as key and value as a square of the even number

In [8]:

```
even_square = {i:i**2 for i in range(1, 10) if i%2 == 0}
print(even_square)
```

```
{2: 4, 4: 16, 6: 36, 8: 64}
```

create a dictionary key as number and value as "Even"/"Odd" using dictionary comprehension

{1:"Odd", 2:"Even"....}

In [9]:

```
even_odd = {i:"Even" if i%2 == 0 else "Odd" for i in range(1, 10)}
print(even_odd)
```

```
{1: 'Odd', 2: 'Even', 3: 'Odd', 4: 'Even', 5: 'Odd', 6: 'Even', 7: 'Odd', 8:
'Even', 9: 'Odd'}
```

Set comprehensions

In [14]:

```
li = [1,2,3,4,8,6,7,2,4,6,3,3,4,8,6,9,2,5]

unique = {ele for ele in li if ele%2 == 0}
print(unique)
```

```
{8, 2, 4, 6}
```

In [17]:

```
string = "Python is one of the top programming language used for many application"
```

In [20]:

```
vowels = "AEIOUaeiou"
vowel_chars = [char for char in string if char in vowels]
unique_vowels = {char for char in string if char in vowels}

print(vowel_chars)
print(unique_vowels)
```

```
['o', 'i', 'o', 'e', 'o', 'e', 'o', 'o', 'a', 'i', 'a', 'u', 'a', 'e', 'u',
'e', 'o', 'a', 'a', 'i', 'a', 'i', 'o']
{'o', 'e', 'a', 'i', 'u'}
```

In [ ]:

```

```

#include<stdio.h>

- math.h
- string.h
- conio.h
- stdlib.h

- printf()
- scanf()

Modules and Packages

print()

input()

Module

it is a simple python file(.py) consists of variables, functions and clases

Package

if is a group of modules included in a single directory

In [24]:

```
import myModule
```

In [29]:

```
dir(myModule)
```

Out[29]:

```
['__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'even',
 'factorial',
 'pi']
```

moduleName.function()/variable

In [25]:

```
print(myModule.even(8))
```

```
True
```

In [26]:

```
print(myModule.even(7))
```

```
False
```

In [27]:

```
print(myModule.factorial(5))
```

```
120
```

In [30]:

```
from myModule import factorial

print(factorial(5))
```

```
120
```

In [31]:

```
from myModule import factorial as fact

print(fact(5))
```

```
120
```

In [32]:

```
from myModule import *


print(factorial(5))
print(even(5))
```

```
120
False
```

In [35]:

```
import myModule
import myModule as my
from myModule import factorial
from myModule import factorial as fact
from myModule import fact
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-35-949754a6de92> in <module>
      3 from myModule import factorial
      4 from myModule import factorial as fact
----> 5 from myModule import fact

ImportError: cannot import name 'fact' from 'myModule' (C:\Users\Jesus\Deskt
op\PYTHON FIP MAY 2020\myModule.py)
```

In [36]:

```
print(myModule.pi)
```

```
3.14
```

In [37]:

```
print(myModule.p)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-37-28cd6f5e0a75> in <module>
----> 1 print(myModule.p)

AttributeError: module 'myModule' has no attribute 'p'
```

In [39]:

```
from myPackage import myModule
from myPackage.myModule import factorial
from myPackage.myModule import factorial as fact
from myPackage.myModule import *
```

In [40]:

```
print(myModule.even(5))
print(factorial(5))
print(fact(5))
print(pi)
```

```
False
120
120
3.14
```

In [1]:

```
from myPackage.myModule import *
```

In [2]:

```
print(x)
```

```
5
```

In [3]:

```
def factorial(number):
    fact=1
    for i in range(1,number+1):
        fact *= i
    return fact
```

In [4]:

```
factorial(5)
```

Out[4]:

```
120
```

pip install packageName

conda install packageName

In [ ]:

```
```