Hello Guys as I mentioned earlier here is the notebook for **Basics of Python**. (I'll recommend you should create a new notebook try these program line by line on your own).

Also, I have added few simple tasks for practice and better understaning please try them as well.

In this notebook I haven't added anything about libraries *(except one program)* (import and all sorts of things).

I'll make next notebook on libraries and various functions.

Please take this seriously as this will be really helpful for us in placements or higher studies as well.

If you didn't get anything then just try to search it on internet (most of the doubts can solved this way) otherwise ask in WhatsApp group we will discuss there.

Please let me know if any modificcations are required in this notebook and also give feedback about how is this notebook.

Thank you!

Python Basics

- **print() function**

In [1]:
```
print("Hello World! \nHi We are ML Project Team No. 20")
```

```
Hello World!
Hi We are ML Project Team No. 20
```

In [45]:
```
print("Hello","World!") # if possible run it line by line to have good unders
print('Hello World!')
print("Hello 'World!'\n")
print('"Hello" World!')
print("'Hello' \"World\" \n")
print('''Hello
World''')
print('''Hello
\nWorld!''')
```

```
Hello World!
Hello World!
Hello 'World!'

"Hello" World!
'Hello' "World"

Hello
World
Hello

World!
```

Task 1

You have to print the text given below as it is,

" \Hello\ "

Answer is provided at the end of this notebook.

In [69]:
```
subject = "ML Project Team No. 20"
print("Hello, we are",subject)
print("Hello, we are %s"% subject)
print("Hello, we are {}".format(subject))
print(f"Hello, we are {subject}") #compatible with only python version 3.6 or
```

```
Hello, we are ML Project Team No. 20
Hello, we are ML Project Team No. 20
Hello, we are ML Project Team No. 20
Hello, we are ML Project Team No. 20
```

In [70]:
```
subject = "ML Project Team No. 20" #String value assigning
subject
```

Out[70]: 'ML Project Team No. 20'

In [81]:
```
group_members = 4                #integer
floating_point = 3.14            #Float
complex_no = 4 + 4j              #Complex no.
boolean_variable = True          #Boolean
```

In [82]:
```
print(group_members)
print(type(group_members))
print(floating_point)
print(f'{type(complex_no)}')
print("{}".format(type(complex_no))) #using .format
```

```
4
<class 'int'>
3.14
<class 'complex'>
<class 'complex'>
```

- ***Python for arithmatic operation or as calculator***

In [96]:
```
a = 4
b = 3
print("a+b = {}".format(a+b)) # Addition
print("a-b = {}".format(a-b)) # Subtraction
print("a*b = {}".format(a*b))  # Multiplication
print("a/b = {}".format(a/b))  #Division
print("a//b = {}".format(a//b)) #Quotient
print("a%b = {}".format(a%b))   #Remainder
print("a**b = {}".format(a**b)) #indices or power
```

```
a+b = 7
a-b = 1
a*b = 12
a/b = 1.3333333333333333
a//b = 1
a%b = 1
a**b = 64
```

- **Tuple()**
  Tuples are sequence of values which **can't be changed.**

In [119]:
```
tuple1 = ('Machine','Learning','Project','Team','No.',20)
print(tuple1)
```

```
('Machine', 'Learning', 'Project', 'Team', 'No.', 20)
```

In [183]:
```
print("{}".format(type(tuple1))) # OR simply you can use below one
print(type(tuple1))
```

```
<class 'tuple'>
<class 'tuple'>
```

In [142]:
```
print(tuple1[2])
tuple1[3]           #To access the elements, use the indices (starting from z
```

```
Project
```

Out[142]:
```
'Team'
```

In [169]:
```
tuple1[5], tuple1[4], tuple1[3], tuple1[2], tuple1[1], tuple1[0]
```

Out[169]:
```
(20, 'No.', 'Team', 'Project', 'Learning', 'Machine')
```

In [165]:
```
tuple1[1:4] #index slicing syntax: [inclusive:exlusive]
```

Out[165]:
```
('Learning', 'Project', 'Team')
```

In [176]:
```
print(tuple1[1:5:2]) #full syntax: [inclusive_start:exclusive_end:step_size]

#If inclusive_start or exclusive_end is blank, then it will slice from the be
print(tuple1[:4:2]) # inclusive_start is blank
print(tuple1[1::2]) # exclusive_end is blank
```

```
('Learning', 'Team')
('Machine', 'Project')
('Learning', 'Team', 20)
```

In [ ]:
```
tuple1[4] = 3 #By the way tuple does not support item assignment, so if you r
```

In [207]:
```
# Nested tuple: tuple inside tuple- try some examples for better understandir
nested_tuple1 = ('j',98,(['a','b','c'],['d',7,3,1]),'k')
print(nested_tuple1)
print("{}".format(type(nested_tuple1)))
print(nested_tuple1[2])     #please observe the output of this line carefully
print(nested_tuple1[3])
```

```
('j', 98, (['a', 'b', 'c'], ['d', 7, 3, 1]), 'k')
<class 'tuple'>
(['a', 'b', 'c'], ['d', 7, 3, 1])
k
```

- **Lists[ ]**
  Lists are similar to tuple but they can be **changed**.

In [269]:
```
list1 = ['Machine','Learning','Project','Team','No.',20]
print(list1)
print(list1[1:5:2]) #index slicing
print(list1)
#nested_list=[['a','b','c','d'],[1,2,5,7,8],['Team', 'No.', 20]]
```

```
['Machine', 'Learning', 'Project', 'Team', 'No.', 20]
['Learning', 'Team']
['Machine', 'Learning', 'Project', 'Team', 'No.', 20]
```

In [270]:
```
list1.append(['Project','details'])
print(list1)
```

```
['Machine', 'Learning', 'Project', 'Team', 'No.', 20, ['Project', 'detail
s']]
```

In [271]:
```
del list1 [1:3]
print(list1)
```

```
['Machine', 'Team', 'No.', 20, ['Project', 'details']]
```

In [272]:
```
list1.remove('Machine')
print(list1)
list1.remove(['Project','details'])
print(list1)
```

```
['Team', 'No.', 20, ['Project', 'details']]
['Team', 'No.', 20]
```

In [304]:
```
#nested list
my_list = ['T','Y','B.Tech',['Autonomy','First Batch'], 2017,'Pattern']
print(my_list)
print(my_list[1:4]) #index slicing

#Function for length of any kind of collection data type (tuples, list, dicti
len(my_list) #list inside the list (nested list) is counted as one element
print("Lenght of List is {}".format(len(my_list)))
```

```
['T', 'Y', 'B.Tech', ['Autonomy', 'First Batch'], 2017, 'Pattern']
['Y', 'B.Tech', ['Autonomy', 'First Batch']]
Lenght of List is 6
```

In [303]:
```
my_list.append ('E&TC')
print(my_list)
```

```
['T', 'Y', 'B.Tech', ['Autonomy', 'First Batch'], 2017, 'Pattern', 'E&TC']
```

- **Strings (str)**

In [309]:
```
my_string = "ML Project Team No. 20"
print(my_string)
```

```
ML Project Team No. 20
```

In [311]:
```
print(my_string)
print(my_string[3:10])
```

```
ML Project Team No. 20
Project
```

In [312]:
```
#Split the string
print(my_string.split()) #splits the string shows the individual element
```

```
['ML', 'Project', 'Team', 'No.', '20']
```

In [325]:
```
my_word = "please,split:the:string.into:parts"
print(my_word.split(':'))
print(my_word.split(','))
print(my_word.split('.'))
print(my_word.split(':',0)) #maxsplit
print(my_word.split(':',3))
```

```
['please,split', 'the', 'string.into', 'parts']
['please', 'split:the:string.into:parts']
['please,split:the:string', 'into:parts']
['please,split:the:string.into:parts']
['please,split', 'the', 'string.into', 'parts']
```

Task 2

Try to do some operation like item assignment or append or delete or remove and check if are they allowed/valid on strings...

- **Dictionaries { }**
  Instead of indexing by strictly numbers from 0 to `len` -1, we can index a *value* with any *key* as long as the *key* is immutable (can be a tuple).

In [86]:
```
project_dictionary = {'AAD':1,'ARP':2,'NGP':3,'RSG':4}
print(project_dictionary)
```

```
{'AAD': 1, 'ARP': 2, 'NGP': 3, 'RSG': 4}
```

In [87]:
```
member = 'ARP'
print(f'Group Member {project_dictionary[member]} is {member}')

#BTW you might be wondering what these 3 lettered words are?
#These are initials (alphabetically arranged) of our group members...😂😂😂
```

```
Group Member 2 is ARP
```

Task 3

Try different combinations of Initial and group member number in the above program.

- **Conditionals (`if...elif...if`)**
  Let's understand conditionals with the help of simple example *(indentation is very important in conditionals, loops, etc)*

Syntax:

```
if:condition
 elif:condition
 else:
```

In [14]:
```
c = 3
if c > 6:
    print("if condition is satisfied")
    print(c,"is greater than 6")
elif c == 6:
    print("elif condition is satisfied")
    print(c,"is equal to 6")
else:
    print("else condition is satidfied")
    print(c,"is less than 6")
```

```
else condition is satidfied
3 is less than 6
```

In [18]:
```
d = 23
if 11 < d < 13:
    print("d is in between 11 and 13")
elif d > 13:
    print("d is greater than 13")
else:
    print("d is less than 11")
```

```
d is greater than 13
```

Logical operations `and` , `or` , and `not` can also be used.

In [22]:
```
e = True
f = False
print(e and f)
print(e or f)
print(not (e and f))
```

```
False
True
True
```

Task 4

Try to use `and` , `or` to check whether student is passed/failed in VIIT exams.

Also try different examples with different conditions.

- ## Loops

  `for` and `while`

In [50]:
```
for h in [3,4,5,6]:
    print(h)
```

```
3
4
5
6
```

In [58]:
```
for i in range (12,18,2):   # range gives you a range of numbers! syntax: rar
    print(i)
```

```
12
14
16
```

In [69]:
```
repeat = 1
while repeat <= 3:
    print (repeat**2, end = ",")
    repeat+= 1
```

```
1,4,9,
```

Parsing a file for data

Here we will learn to parse the file to acces data. If `with` function is used then make sure the file in which data is stored should be in the same folder as this notebook is saved. But for the time being I am using pandas to show the data, we will look into other libraries including pandas later on. Just look at steps and observe syntax you need not remember it as you will practice you will automatically get it.

In [89]:
```
import pandas as pd
member_data = pd.read_csv(r"C:\Users\Pawar\Desktop\Project-Member-Details.csv
member_data
```

Out[89]:

	Member No.	Name	Initials	G. R. No.	Division
0	1	Arya A. Deshmukh	AAD	17U683	A
1	2	Akash R. Pimple	ARP	17U528	B
2	3	Nishant G. Pawar	NGP	17U031	B
3	4	Rohan S. Garje	RSG	17U246	A

Answers of Tasks:

- **Task 1 Solution**

In [4]:
```
print("\" \Hello\ \"")
```

```
" \Hello\ "
```

- **Task 2 Solution**
  String doesn't allows item assignment/append/remove/del

- **Task 3 Solution**
  Change the value assigned to member and observe changes

- **Task 4 Solution**
  Change the value assigned to member and observe changes

In [46]:
```
print("VIIT Exam Pass/Fail checker")
t1 = 12
t2 = 10
g = t1+t2
print("T1 + T2 =",g)
end_sem = 13
total = g + end_sem
print("Total = T1 + T2 + End-Sem. =",total)
if 0 < g >= 20 and 0 < end_sem >= 20:
    print ("You have passed exam.")
elif 0 < g >= 20 and 0 < end_sem < 20:
    print ("Failed in End-sem. exam, quickly pay reexam fees in exam section
elif 0 < g < 20 and 0 < end_sem >= 20:
    print ("Failed in In-sem. exam, quickly pay reexam fees in exam section (
elif 0< g < 20 and 0 < end_sem < 20:
    print ("Failed in both In-sem. and End-sem. exam, quickly pay reexam fees
else:
    print("Error occured contact class teacher or exam section immediately")
```

```
VIIT Exam Pass/Fail checker
T1 + T2 = 22
Total = T1 + T2 + End-Sem. = 35
Failed in End-sem. exam, quickly pay reexam fees in exam section (A303)😂
```

For more information:

- Python Wiki (https://wiki.python.org/moin/)
- Built-in Types (https://docs.python.org/3/library/stdtypes.html)
- String Formatting (https://docs.python.org/3/library/string.html#format-string-syntax)

- ***Reference: Krittika - Astronomy Club of IIT Bombay***
