Python Basics

Why Python ?

- Design Philosphy : user friendly,easy to learn and use and well idented
- Batteries Included : Have built in strong libraries, functions, membership operator etc.
- General Purpose : support paradigm like functional programming , OOPS .
- Libraires / Community : Having strong libraries and community support.

where as in research we use 'R'

- Note - In industry we mostly use 'Python'

Why Python for Data Science ?

- Easy to learn
- Proximity with maths : have mathematical libraries like numpy and scipy
- community

Python Output

Python is a case sensitive language

In [114…
```
print("this function is used to print anything on screen")
```
```
this function is used to print anything on screen
```

In [116…
```
print("Hii , printing mixed values",123,56.34,True,34+5j)
```

```
Hii , printing mixed values 123 56.34 True (34+5j)
```

In [122…

```python
# sep parameter used to separate multiple values in a print function
print("Hii , printing mixed values with 'sep' parameter",123,56.34,True,34+5j,sep='||')
```

```
Hii , printing mixed values with 'sep' parameter||123||56.34||True||(34+5j)
```

In [126…

```python
# end parameter is used to print next print() value with a specific end value
print("Hii , printing mixed values with 'sep' parameter",123,56.34,True,34+5j,sep='||',end="====")
print("this will come  after end of previous print with end parameter value")
```

```
Hii , printing mixed values with 'sep' parameter||123||56.34||True||(34+5j)====this will come  after end of previous print with
end parameter value
```

Python Data Type

Integer

In [138…

```python
print(8)
print(1e308) # max value in integer
print(1e309)
```

```
8
1e+308
inf
```

Decimal / Float

In [142…

```python
print(3.45)
print(1.7e308) # max value of float
print(1.7e309)
```

```
3.45
1.7e+308
inf
```

Boolean

In [145…

```python
print(True,False)
```

```
True False
```

String / Text

In [150…
```
print("this is a text")
# no char data type in python
```

```
this is a text
```

Complex

In [153…
```
print(45+89j)
```

```
(45+89j)
```

List (Alternative of array)

In [158…
```
L=[12,34,566,34.646,True,34+45j,"can have mixed data type"]
print(L)
```

```
[12, 34, 566, 34.646, True, (34+45j), 'can have mixed data type']
```

Tuple

In [163…
```
T=(1,2,3,False,34+34j,"can have mixed data")
print(T)
```

```
(1, 2, 3, False, (34+34j), 'can have mixed data')
```

Sets

In [166…
```
S={1,23,4,34,12,4,24,1,1,1,34}
print(S)
```

```
{1, 34, 4, 23, 24, 12}
```

Dictionary

In [169…
```
D={"this ia key":"this is value","another key":"another value"}
print(D)
```

```
{'this ia key': 'this is value', 'another key': 'another value'}
```

Type - Used to check data type of any value

In [173…
```
print(type(2),type(34.43),type([2,4,34,45]))
```

```
<class 'int'> <class 'float'> <class 'list'>
```

Python Variable

Variable - containers to store anything in future

In [178…
```
name="name is a variable storing this string"
print(name)
```

```
name is a variable storing this string
```

- Dynamic Typing - Python is smart enough to understand a variable's data type at run time by seeing the type of values stored in that variable; this is called dynamic typing.No

need to declare the type of a variable.

- Static Typing - We need to declare the variable data type at the time of declaration.

In [202…
```
# no need to declare type of x,y,z
x=10
y=23.5
z=True
print(x+y+z)
```

```
34.5
```

- Dynamic Binding - We can change the data type of a variable in a program at run time.
- Static Binding - We can store only specific type of values in a specific type of variable.

In [208…
```
# Dynamic Binding of variable with values
x=34
print(x+2)
x=True
print(x+1.345)
x="String"
print(x)
```

```
36
2.3449999999999998
String
```

Keywords and ,Identifiers and Comments

Keywords - Resever words in python

Identifiers - Name of any variable,function or amy user defined variable

```
- we can not start identifier with digit
- we can use '_' as a special character only
- indentifier cannot be keywords
```

Comments - lines that are not interpreted by interpreter

In [237…
```
# this is a comment
```

Type Conversion

Converting data type of a value in another data type

- Implicit Conversion - Conversion done by interpreter itself if feasible
- Explicit Conversion - Conversion done by programmer intentionally

In [258…
```
# implicit conversion
x=4+45.63
```

```
print(x,type(x))
# explicit conversion - int(),str(),complex(),float()
print(int("34"))
print(str(34))
print(float(34))
```

```
49.63 <class 'float'>
34
34
34.0
```

Python Input

In [243…

```
input("Input function is used to take any input from user")
```

Out[243…

```
'this is a input'
```

Program - Add two number

In [263…

```
first=int(input("enter first number"))
second=int(input("enter second number"))
sum=first+second
print(f"sum of {first} and {second} is {sum}")
```

```
sum of 34 and 21 is 55
```

Python Literals

Values that store in variables are literals

In [276…

```
# binary literal
bin=0b1010
# decimal literal
dec=2345
# octal literal
oct=0o310
# hexa literal
```

```
hex=0x12c
print(bin,dec,oct,hex,sep=' ')
```

```
10 2345 200 300
```

In [280…

```
# float literal
float_1=10.34
float_2=1.34e3 # 1.3x10^3
float_3=1.5e-3 # 1.5x10^-3
print(float_1,float_2,float_3)
```

```
10.34 1340.0 0.0015
```

In [284…

```
# string literal
s1='this is single quote'
s2="this is double quote"
s3="this is 'mixed' quote"
unicode=u"\U0001f600\U0001f606\U0001f923"
raw_string=r"this is \n raw \t string"
print(s1,s2,s3,unicode,raw_string,sep='\n')
```

```
this is single quote
this is double quote
this is 'mixed' quote
😀😆🤣
this is \n raw \t string
```

In [286…

```
# boolean literal
a=True+13
b=False-34
x=None
print(a,b,x)
```

```
14 -34 None
```