Data Type in Python

In [ ]:

```
Numeric Data Types    >>  int , float , complex
Text Data types       >>  str
Sequence Data Types   >>  list , tuple , set , frozenset
Mapping Data Types    >>  dict
Boolean Data Types    >>  bool  >> True/False
```

Numeric Data Types

1. int

In [1]:

```
Integer  >> whole number  +ve  or -ve
```

```
  Cell In[1], line 1
    Integer  >> whole number  +ve  or -ve
                      ^
SyntaxError: invalid syntax
```

In [ ]:

```
56
-989
0
345
-89


```

In [3]:

```
m=200   # No need to explicitly define Data Type
print(f"Value of m is {m}")

```

```
Value of m is 200
```

In [ ]:

```
#other language   :   # explicitly define data type of var
    int age=40
    float marks= 78.90
```

In [8]:

```
m=200
print(f"Value of m is {m}")
print(f"Data Type of m is {type(m)}")
```

```
Value of m is 200
Data Type of m is <class 'int'>
```

In [9]:

```
x=700
x
type(x)   # type () is built-in function which returns data type of variable
# print(type(x))
```

Out[9]:

```
int
```

In [11]:

```
Age=30
print(Age)
print(type(Age))
```

```
30
<class 'int'>
```

2.float

In [ ]:

```
float  >> + ve or -ve decimal number
decimal point

56.89
4.78
-79.34
-0.000045
0.0000000023
```

In [13]:

```
marks=89.76
print(f"Marks is  : {marks}")
print(f"Data Type of marks is  : {type(marks)}")
```

```
Marks is  : 89.76
Data Type of marks is  : <class 'float'>
```

In [15]:

```
x=0.00789
x
type(x)
```

Out[15]:

```
float
```

In [16]:

```
y=-0.6450
print(y)
print(type(y))
```

```
-0.645
<class 'float'>
```

3.complex

In [ ]:

```
3+5j
3 >>> real part
5  >>> imaginary part

```

In [18]:

```
x=3+5j
print(x)
print(type(x))
```

```
(3+5j)
<class 'complex'>
```

In [23]:

```
# Member Activity : Take help of google
# Seperate out Real Part and imaginary part from Complex Number
print("Real Part is ",x.real)
print("Imaginary part is ",x.imag)

```

```
Real Part is  3.0
Imaginary part is  5.0
```

In [24]:

```
z=5+0.7j
print(z)
print(type(z))
```

```
(5+0.7j)
<class 'complex'>
```

In [25]:

```
z=5.8+0.7j
print(z)
print(type(z))
```

```
(5.8+0.7j)
<class 'complex'>
```

In [ ]:

```

```