Data types

Int Data Type

```
In [3]:  a = 100
```

```
In [4]:  print(a)
```

100

```
In [5]:  type(a)
```

Out[5]: int

```
In [6]:  a = 123
```

```
In [7]:  print(a)
```

123

```
In [8]:  type(a)
```

Out[8]: int

```
In [9]:  a = 134
         print(a,type(a))
```

134 <class 'int'>

Float Data Type

```
In [12]:  a = 12.34
          print(a, type(a))
```

12.34 <class 'float'>

```
In [13]:  a = 0.9999
          print(a,type(a))
```

0.9999 <class 'float'>

```
In [15]:  a = 3455.9
          print(a, type(a))
```

3455.9 <class 'float'>

```
In [16]:  a = 3e2
          print(a,type(a))
```

300.0 <class 'float'>

```
In [17]:   a = 4e-2
           print(a,type(a))
```

```
0.04 <class 'float'>
```

```
In [19]:   a = 34e-4
           print(a,type(a))
```

```
0.0034 <class 'float'>
```

Bool Data Type

```
In [20]:   a = True
           print(a, type(a))
```

```
True <class 'bool'>
```

```
In [21]:   b = False
           print(b,type(b))
```

```
False <class 'bool'>
```

```
In [22]:   a = True
           b = False
```

```
In [24]:   c = a+b
           print(c)
```

```
1
```

```
In [26]:   c = a-b
           print(c)
```

```
1
```

```
In [27]:   print(2*True+True-False)
```

```
3
```

```
In [28]:   print(2*False-False)
```

```
0
```

Complex Data type

```
In [29]:   a = 2+3j
           print(a,type(a))
```

```
(2+3j) <class 'complex'>
```

```
In [30]:   a = 2-3j
           print(a,type(a))
```

```
(2-3j) <class 'complex'>
```

```
In [31]:   c = 2.5+3.6j
           print(c,type(c))
```

```
(2.5+3.6j) <class 'complex'>
```

In [32]:
```
d = -3.4-6.7j
print(d,type(d))
```

```
(-3.4-6.7j) <class 'complex'>
```

In [33]:
```
a = 2+3j
print(a.real)
```

```
2.0
```

In [35]:
```
print(a.imag)
```

```
3.0
```

In [36]:
```
a = -3.5+4j
print(a.real)
```

```
-3.5
```

In [38]:
```
print(a.imag)
```

```
4.0
```

In [39]:
```
a = -2.3-4.5j
print(a.real)
```

```
-2.3
```

In [41]:
```
print(a.imag)
```

```
-4.5
```

String Data Type

single line string data

In [45]:
```
s1 = "PYTHON"
print(s1,type(s1))
```

```
PYTHON <class 'str'>
```

In [46]:
```
s2 = 'PYTHON'
print(s2,type(s2))
```

```
PYTHON <class 'str'>
```

In [47]:
```
s3 = 'k'
print(s3,type(s3))
```

```
k <class 'str'>
```

In [62]:
```
s1 = "123456"
print(s1,type(s1))
```

```
123456 <class 'str'>
```

In [63]:
```
s2 = "python3.11"
print(s2,type(s2))
```

```
python3.11 <class 'str'>
```

In [64]:
```python
s3 = "123$456_abc"
print(s3,type(s3))
```

```
123$456_abc <class 'str'>
```

In [65]:
```python
s4 = "@#$%^&8912"
print(s4,type(s4))
```

```
@#$%^&8912 <class 'str'>
```

Multi line string data

In [48]:
```python
s4 = """Rossum
H.NO:3-4-12, Hill Towers
CWI Nether Lands"""
print(s4)
```

```
Rossum
H.NO:3-4-12, Hill Towers
CWI Nether Lands
```

In [49]:
```python
s5 = '''James Godling
H.NO: 4-5, Fort side
Sun Micxro System
USA'''
print(s5)
```

```
James Godling
H.NO: 4-5, Fort side
Sun Micxro System
USA
```

In [50]:
```python
s6 = """python is an oop lang"""
print(s6)
```

```
python is an oop lang
```

In [52]:
```python
s6 = '''python is an oop lang'''
print(s6)
```

```
python is an oop lang
```

Operation on str data

indexing

In [53]:
```python
s = "PYTHON"
print(s)
```

```
PYTHON
```

In [54]:
```python
print(s[0])
```

```
P
```

In [56]:
```python
print(s[-6])
```

P

In [57]:
```python
print(s[5])
```

N

In [58]:
```python
print(s[-1])
```

N

In [59]:
```python
print(s[3])
```

H

In [60]:
```python
print(s[-3])
```

H

Slicing

In [61]:
```python
s = "PYTHON"
print(s)
```

PYTHON

In [66]:
```python
s[0:3]
```

Out[66]: 'PYT'

In [67]:
```python
s[1:5]
```

Out[67]: 'YTHO'

In [68]:
```python
s[3:6]
```

Out[68]: 'HON'

In [69]:
```python
s[-6:-2]
```

Out[69]: 'PYTH'

In [70]:
```python
s [-5:-3]
```

Out[70]: 'YT'

In [72]:
```python
s[-4:-2]
```

Out[72]: 'TH'

In [73]:
```python
s = "PYTHON"
print(s,type(s))
```

PYTHON <class 'str'>

In [74]: `s[2:-2]`

Out[74]: `'TH'`

In [75]: `s[1:-1]`

Out[75]: `'YTHO'`

In [76]: `s[3:-1]`

Out[76]: `'HO'`

In [77]: `s[1:-3]`

Out[77]: `'YT'`

In [78]: `s[3:]`

Out[78]: `'HON'`

In [79]: `s[2:]`

Out[79]: `'THON'`

In [80]: `s[5:]`

Out[80]: `'N'`

In [81]: `s[1:]`

Out[81]: `'YTHON'`

In [82]: `s[-5:]`

Out[82]: `'YTHON'`

In [ ]: