

# Data Types (Int,Float,String,Bool,Complex)

```
In [3]: iphone = 560 # Value with out decimal)
        type (iphone)
```

```
Out[3]: int
```

```
In [23]: samsungphone = 980
         type (samsungphone)
```

```
Out[23]: int
```

```
In [19]: type(samsung phone)
```

```
Cell In[19], line 1
      type(samsung phone)
            ^
```

**SyntaxError:** invalid syntax. Perhaps you forgot a comma?

```
In [25]: print(type(samsungphone))
```

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

```
In [27]: print(type iphone)
```

```
Cell In[27], line 1
      print(type iphone)
            ^
```

**SyntaxError:** invalid syntax

```
In [29]: print(type(iphone))
```

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

```
In [31]: bike=66.50
         type(bike)
```

```
Out[31]: float
```

```
In [33]: print(type(bike))
```

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

```
In [35]: print(bike)
```

```
66.5
```

```
In [45]: book="Bible"
         type(book)
```

```
Out[45]: str
```

```
In [47]: print(type(book))
```

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

```
In [49]: lenova=True  
type(lenova)
```

```
Out[49]: bool
```

```
In [51]: print(type(lenova))
```

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

Complex data type (a+bj)- a = real , b=imaginary, j -squareroot of -1

```
In [145... c1=10+20j  
c1.real
```

```
Out[145... 10.0
```

```
In [147... c1.imag
```

```
Out[147... 20.0
```

```
In [68]: rt=35+45j  
rt.real
```

```
Out[68]: 35.0
```

```
In [70]: rt.imag
```

```
Out[70]: 45.0
```

```
In [72]: rt
```

```
Out[72]: (35+45j)
```

```
In [76]: rt.real
```

```
Out[76]: 35.0
```

```
In [78]: print(c1)  
print(rt)
```

```
(10+20j)
```

```
(35+45j)
```

```
In [80]: c1+rt #addition
```

```
Out[80]: (45+65j)
```

```
In [84]: c1-rt # subtraction
```

```
Out[84]: (-25-25j)
```

```
In [86]: rt-c1
```

Out[86]: (25+25j)

In [94]: c2=20+30j

In [96]: print(c1)  
print(c2)

(10+20j)

(20+30j)

In [98]: c1\*c2

Out[98]: (-400+700j)

In [102... verse='''God is with us always  
blessins he will shower'''  
verse

Out[102... 'God is with us always\nblessins he will shower'

In [108... word='''rejoice in the lord  
always'''  
word

Out[108... 'rejoice in the lord\nalways'

Slicing - [:]

Indexing - 0,1,2,3,4,5....-1

forward index and backward index

In [125... s='rejoiceinthelordalways'

In [129... s[5] *#forward index*

Out[129... 'c'

In [131... s[-3] *#backward index*

Out[131... 'a'

In [133... s[-1]

Out[133... 's'

In [141... s[6:9]

Out[141... 'ein'

In [143... len(s)

Out[143... 22

Type Casting - other data types to int

In [149... `int(900.55) # float to int`

Out[149... 900

In [151... `int (670.29)`

Out[151... 670

In [153... `int(True) # Bool to Int`

Out[153... 1

In [155... `int(False)`

Out[155... 0