# Python - tutorial 1

* fundamental types : [int / float / str / complex / bool / list / tuple]

- You don't need to declare variable type in python unlike other pro. lang.
- if you want to know kind of data stored in any variable (say i), then use, type(i) → int / float / str / complex
- if you want to declare string then use, 'Madhu' or "Madhu".
- complex → i = 1 + 7j (1 = real, 7j = img.)
- there are some inbuilt functions in Python.

ex: k = 3 + 6j
k.img = 6 (img is inbuilt func.)

- some inbuilt functions work with (), & some without (). [Mostly with ()]
- to know address or id or location of a variable inside memory we use, id(k).
- i = False → type(i) = bool (true/false) boolean
- value of true (1), value of false (0).

so, True + True = 2, False + False = 0 & so on...

- "MADHU"S" , "MADHU'S" , 'MADHU's' , 'MADHU"S'
  error ✗ ✓ ✗ ✓
- """MADHU"S""" , 1 + "MADHU"
  ✓ error

{ interpreted →
{ compiled →
{ Python is interpreted as well as compiled depends on library/package/installations we use.

- (1 + "MADHU"), ('1' + "MADHU" = '1MADHU'). (X)
- Typecasting: str(1) + "MADHU = '1MADHU'.
- integer → string (Possible)
  - string (string of int.) → integer (Possible)
    - eg: int('62') + "MADHU" = '62MADHU'.
  - string (string of char.) → integer (Not Possible)
    - eg: int('MAD') + "HU" → error.
- Multiplication operator:
  "MA" * 10 = MAMAMAMAMAMAMAMAMAMA
  4 * 10 = 40
  4 ** 10 = 1048576 (4 ** 10 = $4^{10}$) (** : power)
- shift + tab = details of function used.
- to access any variable (i = 5 < print(i))
- Boolean evaluation: (AND, OR, NOT).
  True or False = True ... (1 or 0 = 1)
  not True = False (~1 = 0)
  True is True = True (1 is 1 = 1)
  True is False = False (1 is 0 = 0)
  True * False = False (1 * 0 = 0)
  True - False = True (1 - 0 = 1)
  True / False = error (we get this result in Numpy)
- Python supports scripting & OOPS.

Note: mind indentation while writing Python prog.
(don't worry its automatic).

- Conditional statement: (if, else)
(if, else are keywords).

```
i = 9
if i==10:        (in other lng we use {}, here use :)
    pass         (pass is keyword, when you don't want to write body)
```

```
i = 9
if (i<6):        (: use () or don't no matter)
    print("Cool")
elif i<3:
    print("choice")
:
else:
    print("default statement")
```

Indentation

- i = 7
id(i) = 1342
i = 9
id(i) = 1538

- Nested if/else:

```
s = "MADHU"
if True:          (True means if will be executed)
    if s=="sudh":
        print("Yeah")
    else:
        print("Naah")
```

function → we give some data & receive output
key-word →

else:
    print("Haan")
[output: Naah].

- i = 3+8j
  j = type(i)
  type(j) = type
- Lists: store multiple data-set at a time:
  k = [ ]  OR  k = list( )
  type(k) = list  type(k) = list
(homogeneous / heterogeneous)

LR → 0 1 2 3
k = ["MADHU", 5, 8.32, 2+7j]
    -4  -3  -2  -1 ← RL

k[2] = 5, type(k[3]) = float
k[1:3] = [5, 8.32]  ([1:2] = 1 to 2 index)
k[0:3:2] = ['MADHU', 8.32]
(or) k[:3:2]  (by default from starting 0-index)
(start end gap)
k[3:1:1] = [ ] (blank)
k[3:1:-1] = [2+7j, 8.32]
s = "MADHU" → s[::-1] = 'UHDAM'

```
k = ["MA", "SU", 3, "TA", 5.82]
for i in k:
    if type(i) == str:
        print(i)
```

output:
MA
SU
TA

- b = [ ]
- for i in k:
    if type(i) == str:
    b.append(i)
  print(b)

Output

('MA', 'SU', 'TA')

- k = 4, 5, 6
  type(k) = tuple