

# Datatypus

Numeric. — 

- integral — int (Whole Numbers)
- non-integral — float (Fractions)



Boolean → bool  
True / False

$x = 10$       $\boxed{10}$   
 $x \rightarrow \text{int.}$

Collections — 

- Sequence. —
  - Mutable. → list
  - Immutable → tuple, str  
( ), " "
- Mapping → dict

$\begin{matrix} 23 \rightarrow M1 \rightarrow 53 & M24 \rightarrow \\ & \vdots \\ & \vdots \\ & \vdots \\ & M23 \rightarrow 65 \end{matrix}$

Common properties

-7	-6	-5	-4	-3	-2	-1	→ elements
35	46	75	96	58	85	89	
0	1	2	3	4	5	6	→ index.

(1) positional Indexing-

lst = [35, 46, 75, 96, 58, 85, 89] ✓  
 type(lst) → list

lst2 = [10, 5.172, True, 'Apple', [20, 30, 40],  
 (1, 2, 3)]  
 → Heterogeneous

Access Values:-

lst[3] → 96  
 ↳ extractor  
 lst[-4] → 96

t1[-1] → last elements  
 s2[-1] → last elements

str1 = 'Hello'

→ sequence of 'characters'

s2 = "python 3.0" → Homogeneous  
 type(s2) → string.

t1 = (25, 43, 74, 60)

type(t1) → tuple. str1[1] → 'e'

t1[1] → 1



Sequences are iterable

Iterate → take one element at a time.

Shop-list = [ 'Apples', 'Rice', 'Soap', 'Shampoo', 'Batter' ]



for mark in mark-list: ↗ 5 times

- if mark > 80  
distinction
- if mark 60 and 80  
First
- if mark 40 and 60  
Second

5 times loop

marks = [ 12, 58, 75, 64, 52 ]

if marks[0] > 80 → dist-  
if marks[0] between 60 and 80 → First-  
if marks[0] between 40 and 60 → second

if marks[1] > ...  
-----  
-----  
-----

if marks[2] > 80 → dist  
-----  
-----  
-----

→ for elem in shop-list:  
    → print(elem) ↗ loop  
    print('Hi') Apples  
                                Rice  
                                Soap

l1 = [1, 2, 3]  
→ l2 = l1 Assign  
↓  
[1, 2, 3]

l1 ✓ → [1, 2, 3]  
l2 → [100]

l2.append(100) ✓  
l2 → 1, 2, 3, 100  
l1 → 1, 2, 3, 100

l2 = l1.copy() ←  
201

l1 = [1, 2, 3]

l2 = [1, 2, 3, 100]

