DataTypes_IfCondition_DataStructures

November 9, 2025

Making Data Speak-iVeman

```
[1]: # Basic Data Type in Python
```

```
[2]: num1 = 100
     num2 = 200
```

```
[152]: type(num1)
```

```
[152]: int
```

```
[153]: id(num2)
```

```
[153]: 135987260521584
```

```
[154]: num2 = 500
```

```
[155]: id(num2)## Hence for for each data saved ID is created!
```

```
[155]: 135987257920976
```

```
[7]: type(num2)
```

```
[7]: int
```

```
[8]: id(num1)
```

```
[8]: 8888488
```

```
[9]: id(num2)
```

```
[9]: 135987260521584
```

```
[10]: num1+num2
```

```
[10]: 600
```

```
[11]: #Floating data
```

```
[12]: Float_x = 3.45
```

```
[13]: type(Float_x)
```

```
[13]: float
```

```
[14]: id(Float_x)
```

```
[14]: 135987260520048
```

```
[15]: Float_x
```

```
[15]: 3.45
```

```
[16]: print(Float_x)
```

```
3.45
```

```
[17]: Float_x = 9.45
```

```
[18]: print(Float_x)
```

```
9.45
```

```
[19]: Float_y = 10.3425
```

```
[20]: Float_x = Float_y
```

```
[21]: print(Float_x)
```

```
10.3425
```

```
[22]: ## String Variables
```

```
[23]: City = 'Mumbai'
```

```
[24]: State = "Maharastra"
```

```
[25]: len(City)
```

```
[25]: 6
```

```
[26]: print(State[0])
```

```
M
```

```
[27]: print(State[0:4])
```

```
      Maha

[28]: print(State[0:8:2])

      Mhrs

[29]: print(State[-2])

      r

[30]: print(State[-2:])

      ra

[31]: Message = "Making Data Speak"

[32]: len(Message)

[32]: 17

[33]: print(Message)

      Making Data Speak

[34]: print(Message[7:])

      Data Speak

[35]: #reverse Order

[36]: print(Message[-17:])

      Making Data Speak

[37]: print(Message[-17:2])

      Ma

[38]: print(Message[-17:7:])

      Making

[39]: print(Message[::-1])

      kaepS ataD gnikaM

[40]: type(Message)

[40]: str
```

```
[41]: #split function
```

```
[42]: words = Message.split(" ")
```

```
[43]: type(words)
```

```
[43]: list
```

```
[44]: print(words)
```

```
['Making', 'Data', 'Speak']
```

```
[45]: print(words[2])
```

```
Speak
```

```
[46]: Country = "India,USA,UAE,Japan,China,Germany"
```

```
[47]: type(Country)
```

```
[47]: str
```

```
[48]: Country[0:8]
```

```
[48]: 'India,US'
```

```
[49]: len(Country)
```

```
[49]: 33
```

```
[50]: Country[10:13]
```

```
[50]: 'UAE'
```

```
[51]: Cr = Country.split(",")
```

```
[52]: Cr
```

```
[52]: ['India', 'USA', 'UAE', 'Japan', 'China', 'Germany']
```

```
[53]: Name = "University"
```

```
[54]: id(Name)
```

```
[54]: 135987257821424
```

```
[55]: Name = "Mumbai University"
```

[56]:
```
id(Name)
```

[56]:
```
135988493814240
```

[57]:
```
Name+" "+Country
```

[57]:
```
'Mumbai University India,USA,UAE,Japan,China,Germany'
```

[58]:
```
X = 10
```

[59]:
```
Y = 20
```

[60]:
```
cond1 = X < Y
```

[61]:
```
cond1
```

[61]:
```
True
```

[62]:
```
id(cond1)
```

[62]:
```
8883552
```

[63]:
```
type(cond1)
```

[63]:
```
bool
```

[64]:
```
X = int(input("Enter a number: "))
Y = int(input("Enter a number: "))

if X>Y:
    print(X,"is Greater then Y")
elif X<Y:
    print(Y,"is Greater then X")
else:
    print("Oho")
```

```
Enter a number:  11
Enter a number:  12
```

```
12 is Greater then X
```

[65]:
```
type(X)
```

[65]:
```
int
```

[68]:
```
X = int(input("Enter Number: "))
if X>1:
    print("This is Postive Number: ")
```

```
elif X<1:
    print("This is negative number: ")
else:
    print("This is neutral")
```

```
Enter Number:  3

This is Postive Number:
```

[69]:

```
100%3 ==0
```

[69]:

```
False
```

[70]:

```
X = int(input("Enter a number: "))
if X%5==0:
    print(X," is is divisible by 5 ")
else:
    print(X," is not divisible by 5")
```

```
Enter a number:  18

18  is not divisible by 5
```

[71]:

```
A = int(input("Enter a number: "))
B = int(input("Enter a number: "))
C = int(input("Enter a number: "))

if A>B and A>C:
    print(A," is greater then ",B,C)
elif B>A and B>C:
    print(B," is greater then ",A,C)
elif C>A and C>B:
    print(C," is greater then ",A,B)
else:
    print("Ohoo")
```

```
Enter a number:  15
Enter a number:  30
Enter a number:  20

30  is greater then  15 20
```

[72]:

```
A = int(input("Enter a Number: "))
if A%3 == 0 or A%5 == 0:
    print(A," Is divisible by 3 or 5")
else:
    print("Not divisible")
```

```
Enter a Number:  15

15  Is divisible by 3 or 5
```

```
[73]: listt=["Mango","Apple","Chickoo","Orange","Pineapple","Avacoda"]
```

```
[74]: type(listt)
```

```
[74]: list
```

```
[75]: id(listt)
```

```
[75]: 135987257880832
```

```
[76]: print(listt[2])
```

```
Chickoo
```

```
[77]: print(listt[-1])
```

```
Avacoda
```

```
[78]: ##Slicing
```

```
[79]: listt[0:3:2]
```

```
[79]: ['Mango', 'Chickoo']
```

```
[80]: listt[0:4]
```

```
[80]: ['Mango', 'Apple', 'Chickoo', 'Orange']
```

```
[81]: listt[0:]
```

```
[81]: ['Mango', 'Apple', 'Chickoo', 'Orange', 'Pineapple', 'Avacoda']
```

```
[82]: ##printing Mango,Chickoo,Pineapple
```

```
[83]: listt[0:6:2]
```

```
[83]: ['Mango', 'Chickoo', 'Pineapple']
```

```
[84]: for i in range(0,4):
          print(listt[i])
```

```
Mango
Apple
Chickoo
Orange
```

[85]:

```
for x in listt:
    print(x)
```

```
Mango
Apple
Chickoo
Orange
Pineapple
Avacoda
```

[86]:

```
#print in reverse order

for i in range(5,-1,-1):
    print(listt[i])
```

```
Avacoda
Pineapple
Orange
Chickoo
Apple
Mango
```

[87]:

```
##Inserting new element in list

listt.append("Strawberry")
```

[88]:

```
listt
```

[88]:

```
['Mango', 'Apple', 'Chickoo', 'Orange', 'Pineapple', 'Avacoda', 'Strawberry']
```

[89]:

```
type(listt)
```

[89]:

```
list
```

[90]:

```
id(listt)
```

[90]:

```
135987257880832
```

[91]:

```
#list of hetrogenouse mutable ordering!
```

[92]:

```
student = [1,"Rohan","O+",[56,35,76]]
```

[93]:

```
student
```

```
[93]: [1, 'Rohan', 'O+', [56, 35, 76]]
```

```
[94]: type(student)
```

```
[94]: list
```

```
[95]: type(student[0])
```

```
[95]: int
```

```
[96]: type(student[1])
```

```
[96]: str
```

```
[97]: type(student[2])
```

```
[97]: str
```

```
[98]: type(student[3])
```

```
[98]: list
```

```
[99]: print(student[3])
```

```
[56, 35, 76]
```

```
[100]: print(student[3][2])
```

```
76
```

```
[101]: student.append("Sports")
```

```
[102]: student
```

```
[102]: [1, 'Rohan', 'O+', [56, 35, 76], 'Sports']
```

```
[103]: #imutable(id number is saved)
       x = 10
```

```
[104]: id(x)
```

```
[104]: 8885608
```

```
[105]: x = 20
```

```
[106]: id(x)
```

```
[106]: 8885928
```

[107]:
```
## Mutable
```

[108]:
```
id(listt)
```

[108]:
```
135987257880832
```

[109]:
```
listt
```

[109]:
```
['Mango', 'Apple', 'Chickoo', 'Orange', 'Pineapple', 'Avacoda', 'Strawberry']
```

[110]:
```
listt.append("Watermelon")
```

[111]:
```
listt
```

[111]:
```
['Mango',
 'Apple',
 'Chickoo',
 'Orange',
 'Pineapple',
 'Avacoda',
 'Strawberry',
 'Watermelon']
```

[112]:
```
print(listt)
```

```
['Mango', 'Apple', 'Chickoo', 'Orange', 'Pineapple', 'Avacoda', 'Strawberry',
'Watermelon']
```

[113]:
```
id(listt)
```

[113]:
```
135987257880832
```

[114]:
```
address = ["Malad(West)","Mumbai","40064"]
```

[115]:
```
type(address)
```

[115]:
```
list
```

[116]:
```
ad = 23,"34","adsd"
```

[117]:
```
type(ad)
```

[117]:
```
tuple
```

[118]:
```
student.extend(address)
```

[119]:
```
student
```

[119]:
```
[1, 'Rohan', 'O+', [56, 35, 76], 'Sports', 'Malad(West)', 'Mumbai', '40064']
```

[120]:
```
# Adding element in between the List

student = [1,"Rohan","O+",[56,35,76]]
address = ["Malad(West)","Mumbai","40064"]
student.extend(address)
student.insert(3,True)
print(student)
```

```
[1, 'Rohan', 'O+', True, [56, 35, 76], 'Malad(West)', 'Mumbai', '40064']
```

[121]:
```
# Removing element in between the List
student.remove("O+")
```

[122]:
```
student
```

[122]:
```
[1, 'Rohan', True, [56, 35, 76], 'Malad(West)', 'Mumbai', '40064']
```

[123]:
```
# Removing element in between the List with index

del student[2]
```

[124]:
```
student
```

[124]:
```
[1, 'Rohan', [56, 35, 76], 'Malad(West)', 'Mumbai', '40064']
```

[125]:
```
#altertion we will update here the value for this
listt
```

[125]:
```
['Mango',
 'Apple',
 'Chickoo',
 'Orange',
 'Pineapple',
 'Avacoda',
 'Strawberry',
 'Watermelon']
```

[126]:
```
listt[1] = "AlterMango"
```

[127]:
```
listt
```

[127]:
```
['Mango',
 'AlterMango',
 'Chickoo',
 'Orange',
 'Pineapple',
 'Avacoda',
 'Strawberry',
 'Watermelon']
```

```
[131]: Fruit_name = str(input("Enter Fruit Name: "))
       if Fruit_name in listt:
           print(Fruit_name," is in this list",listt.index(Fruit_name)+1," postion")
       else:
           print(Fruit_name," is not in the list")
```

```
Enter Fruit Name:  Banana

Banana  is not in the list
```

```
[132]: Weights = [70,80,45,50]
```

```
[133]: maximum = max(Weights)
       print(maximum)
```

```
80
```

```
[134]: total = sum(Weights)
       mean = total/len(Weights)
       mean
```

```
[134]: 61.25
```

```
[135]: tuple_student = (1,"Rohan","o+",True,[56,78,39],"Malad(East)","Mumbai",40923)
```

```
[136]: tuple_student
```

```
[136]: (1, 'Rohan', 'o+', True, [56, 78, 39], 'Malad(East)', 'Mumbai', 40923)
```

```
[137]: type(tuple_student)
```

```
[137]: tuple
```

```
[138]: tuple_student[4][0] = 20
       tuple_student[4][1] = 30
       tuple_student[4][2] = 40
```

```
[139]: tuple_student
```

```
[139]: (1, 'Rohan', 'o+', True, [20, 30, 40], 'Malad(East)', 'Mumbai', 40923)
```

```
[140]: for i in tuple_student:
           print(i)
```

```
1
Rohan
o+
True
[20, 30, 40]
Malad(East)
Mumbai
40923
```

[141]:
```
## Dictonory
```

[142]:
```
Student_dict = {"name": "Veman","rollno": 7,"City": "Bhiwandi","marks":
 ↪[5,6,7,8,8]}
```

[143]:
```
type(Student_dict)
```

[143]:
```
dict
```

[144]:
```
Student_dict.values()
```

[144]:
```
dict_values(['Veman', 7, 'Bhiwandi', [5, 6, 7, 8, 8]])
```

[145]:
```
id(Student_dict)
```

[145]:
```
135987258043392
```

[146]:
```
Student_dict["marks"]
```

[146]:
```
[5, 6, 7, 8, 8]
```

[147]:
```
Student_dict
```

[147]:
```
{'name': 'Veman', 'rollno': 7, 'City': 'Bhiwandi', 'marks': [5, 6, 7, 8, 8]}
```

[148]:
```
Student_dict["rollno"]
```

[148]:
```
7
```

[149]:
```
del Student_dict["City"]
```

[150]:
```
Student_dict
```

[150]:
```
{'name': 'Veman', 'rollno': 7, 'marks': [5, 6, 7, 8, 8]}
```

[151]:
```
Student_dict.get("Veman","Not Found")
```

[151]:
```
'Not Found'
```

[ ]: