Datatypes in Python – (Sequences)

Prof. Kriti Jaiswal
Assistant Professor
Computer Science & Engineering
School of Engineering & Technology

NAVRACHANA UNIVERSITY
a UGC recognized University

Five Types of Built-in Datatypes

NAVRACHANA UNIVERSITY
a UGC recognized University

1. None Type
2. Numeric Type
3. Sets
4. Sequences
5. Mappings

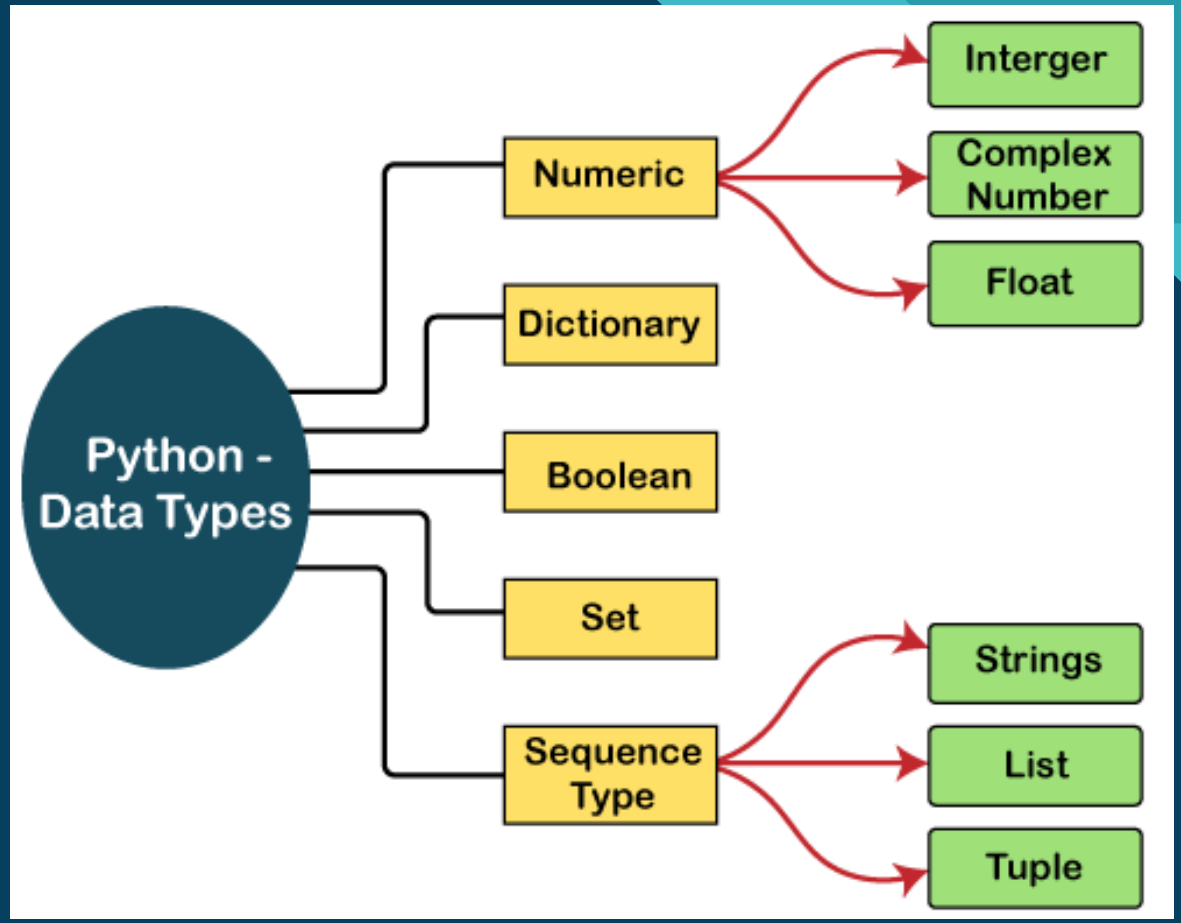

Table of Contents

- Sequences
  - str
  - bytes
  - bytearray
  - list
  - tuple
  - range

Sequences

NAVRACHANA UNIVERSITY
a UGC recognized University

- A sequence represents a group of elements or items.

str Datatype

- In Python, str represents string datatype.
- A string is represented by group of characters.
- Strings are enclosed in single quotes or double quotes.
- The **slice operator** represents square bracket [ ] to retrieve the pieces of string.
- The **repetition operator** is denoted by '*' symbol.

Str Datatype Example:

```
str ="Welcome"

str ='Welcome‘


str1="""This is Python's Lecture

which discusses all the topics of python."""“


str='''This is "Python Lecture" session'''

print(str)
```

Str Datatype Example Contd.:

NAVRACHANA UNIVERSITY
a UGC recognized University

```
s='Welcome to Python Lecture'

print(s)


print(s[0])        # display's 0th character from s

print(s[3:7])      #display's from 3rd to 7th characters

print(s[11:])      #display from 11th character onwards till end

print(s[-1])       #display first character from the end
```

bytes Datatype

- The bytes represents a group of byte numbers.
- Bytes array can store numbers in the range from 0 to 255.
- It cannot even store negative numbers.
- We cannot modify or edit any element in the bytes type array.

bytes Datatype Example:

```
elements = [10, 26, 0, 45, 15]    #this is the list of byte numbers

x= bytes(elements)          #convert the list into byte bytearray

print(x[0])                 #display 0th element
```

bytearray Datatype

- Similar to bytes datatype.
- The bytetype array can be modified.
- It means any element or all the elements of the bytearray type can be modified.

bytearray Datatype Example:

```
elements = [10, 26, 0, 45, 15]        #this is the list of byte numbers
x= bytearray(elements)                #convert the list into bytearray type array
print(x[0])                           #display 0th element
print(x[1])
x[0]=80                               #replace 0th element by 80
x[1]=99
print(x[0])
print(x[1])
```

list Datatype

- Represents a group of elements.
- Can store different types of elements.
- Lists can grow dynamically in memory.
- Represented using **square brackets [ ]** and elements are separated by commas.
- We can perform **slicing operation.**

list Datatype Example:

```
list = [10, -30, 5.6, 'Ram', "Pooja"]

print(list)

#use of slice operator

print(list[0])

print(list[1:3])

print(list[-2])


print(list*3) #use of Repetition operator
```

tuple Datatype

- A tuple is similar to list.
- Contains a group of elements which can be of different types.
- Enclosed in **parentheses ( ) .**
- Not possible to modify the tuple elements.
- Can be treated as a read-only list.

range Datatype

- Represents a sequence of numbers.
- The numbers in the range are not modifiable.
- Used for repeating a for loop for a specific number of times.

range Datatype Example:

NAVRACHANA UNIVERSITY
a UGC recognized University

```
r= range(10) #range object is created from 0 to 9

for i in r: print(i) #display the number from 0 to 9

r=range(30,40,2) #starting number 30 and an ending number 39.
The step size is 2.

for i in r: print(i)


#Create a list with a range of numbers

lst=list(range(10))

print(lst)
```

NAVRACHANA UNIVERSITY
a UGC recognized University

Thank You