# 6. PYTHON – DATA TYPES

## Table of Contents

# 6. PYTHON – DATA TYPES

## 1. What is a Data Type in python?

- ✓ A data type represents the type of the data stored into a variable or memory.

Program Name: print different kinds of variables
demo1.py

```
emp_id = 1
name = "Daniel"
salary = 10000.56

print("My employee id is: ", emp_id)
print("My name is: ", name)
print("My salary is: ", salary)
```

Output

```
My employee id is:  1
My name is: Daniel
My salary is:  10000.56
```

## Note

- ✓ By using type(p) function we can check the data type of each variable.

## 2. type(p) function

- ✓ type(p) is predefined function in python.
- ✓ By using this we can check the type of the variables.

Program Name: Check the type of variables
demo2.py

```
emp_id = 1
name = "Daniel"
salary = 10000.56

print("My employee id is: ", emp_id)
print("My name is: ", name)
print("My salary is: ", salary)
print()
print("emp_id type is: ", type(emp_id))
print("name type is: ", type(name))
print("salary type is: ", type(salary))
```

Output

```
My employee id is:  1
My name is:  Daniel
My salary is:  10000.56

emp_id type is:  <class 'int'>
name type is:  <class 'str'>
salary type is:  <class 'float'>
```

## 3. Different types of data types

- ✓ There are two type of data types.
  1. Built-in data types
  2. User defined data types

## 3.1 Built-in data types:

- ✓ The data types which are already existing in python are called built-in data types.

1. Numeric types
   - int
   - float
2. bool (boolean type)
3. None
4. Sequence
   - str
   - list
   - tuple
   - set
   - dict
   - range

## 1. Numeric types

- ✓ The numeric types represent numbers, these are divided into three types,

    1. int
    2. float
    3. complex

## 1.1 int data type

- ✓ The int data type represents a number without decimal values.
- ✓ In python there is no limit for int data type.
- ✓ It can store very large values conveniently.

Program Name: To print integer value
demo3.py

```
a = 20
print(a)
print(type(a))
```

output

```
20
<class 'int'>
```

Program Name: int data type can store bigger values too
demo4.py

```
b = 9999999999999999999
print(b)
print(type(b))
```

output

```
9999999999999999999
<class 'int'>
```

## 1. 2. float data type

- ✓ The float data type represents a number with decimal values.

| | |
|---|---|
| Program Name | To print float value and data type<br>demo5.py |
| | salary = 10000.56<br>print(salary)<br>print(type(salary)) |
| Output | |
| | 10000.56<br><class 'float'> |

## 2. bool data type (boolean data type)

- ✓ bool data type represents boolean values in python.
- ✓ bool data type having only two values those are,
  - True
  - False

Make a note

- ✓ Python internally represents,
  - True as 1
  - False as 0

Program Name: boolean values
demo6.py

```
a = True
b = False

print(a)
print(b)
```

output

```
True
False
```

## 3. None data type

- ✓ None data type represents an object that does not contain any value.
- ✓ If any object having no value, then we can assign that object with **None** data type.

| | |
|---|---|
| Program Name | None data type<br>demo7.py |
| | a = None<br>print(a)<br>print(type(a)) |
| output | |
| | None<br><class 'NoneType'> |

**Note**

- ✓ A function and method can return **None** data type.
- ✓ This point we will understand more in functions and oops chapters.

## 4. Sequences in Python

- ✓ Sequence means an object.
- ✓ Sequence object can store a group of values,

1. string
2. list
3. tuple
4. set
5. dict
6. range

**Make a note**

- ✓ Regarding sequences like string, list, tuple, set and dict we will discuss in upcoming chapters
  - Python String chapter
  - Python List Data Structure chapter
  - Python Tuple Data Structure chapter
  - Python Set Data Structure chapter
  - Python Dictionary Data Structure chapter

## 4.1 string data type

- ✓ A group of characters enclosed within single quotes or double quotes or triple quotes is called as string.

Program Name: Creating a string
demo8.py

```
name1 = 'Daniel'
name2 = "Daniel"
name3 = '''Daniel'''
name4 = """Daniel"""

print(name1)
print(name2)
print(name3)
print(name4)
print(type(name1))
print(type(name2))
print(type(name3))
print(type(name4))
```

Output

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

## 4.2 list data structure

- ✓ We can create list data structure by using square brackets **[]**
- ✓ list can store a group of values.

Program Name: Creating a list data structure
demo9.py

```
values = [10, 20, 30, 40]
print(values)
print(type(values))
```

Output

```
[10, 20, 30, 40]
<class 'list'>
```

## 4.3 tuple data structure

- ✓ We can create tuple data structure by using parenthesis symbol **()**
- ✓ tuple can store a group of values.

Program Name: Creating a tuple data structure
demo10.py

```
values = (10, 20, 30, 40)
print(values)
print(type(values))
```

Output

```
(10, 20, 30, 40)
<class 'tuple'>
```

## 4.4 set data structure

- ✓ We can create set data structure by using curly braces **{}**
- ✓ set can store a group of values.

Program Name: Creating a set data structure
demo11.py

```
values = {10, 20, 30, 40}
print(values)
print(type(values))
```

Output

```
{40, 10, 20, 30}
<class 'set'>
```

## 4.5 dictionary data structure

- ✓ We can create dictionary data structure by using curly braces **{}**
- ✓ Dictionary can store a group of values in the form of key value pair.

Program Name: Creating a dictionary data structure
demo12.py

```
details = {1: "Jeswanth", 2: "Kumari", 3: "Prasad", 4: "Daniel"}
print(details)
print(type(details))
```

Output

```
{1: 'Jeswanth', 2: 'Kumari', 3: 'Prasad', 4: 'Daniel'}
<class 'dict'>
```

## 4. 6 range data type

- ✓ range is a data type in python.
- ✓ Generally, range means a group of values from starting to ending.

## Creating range of values

- ✓ We can create range of values by using range(p) predefined function

## 1. range(p) function

- ✓ As discussed, we can create range of values by using range(p) function.
- ✓ Here p should be integer, otherwise we will get error.

Program creating a range of values 0 to 4
Name demo13.py

```
a = range(5)
print(a)
print(type(a))
```

Output

```
range(0, 5)
<class 'range'>
```

Note:

- ✓ If we provide range(5), then range object holds the values from 0 to 4

## 2. range(start, end) function

- ✓ As discussed we can create range of values by using range(start, end) function.
- ✓ Here start means starting value and end means till to end-1 value

Program Name: creating a range of values 1 to 9
demo14.py

```
a = range(1, 10)
print(a)
```

Output

```
range(1, 10)
```

Note:

- ✓ If we provide range(1, 10), then range object holds the values from 1 to 9

## Accessing range values by using for loop

- ✓ We can access range values by using for loop.

Program Name: Access elements from range data type
demo15.py

```
r = range(10, 15)

for value in r:
        print(value)
```

output

```
10
11
12
13
14
```

## 3.2 User defined data types

- ✓ Data types which are created by programmer.
- ✓ The datatype which are created by the programmers are called 'user-defined' data types, example is class, module, array etc.
- ✓ We will discuss about in OOPS chapter.