

Data Handling in Python

As per CBSE curriculum
Class 11



Chapter- 3

By-

Neha Tyagi

PGT (CS)

KV 5 Jaipur(II Shift)

Jaipur Region

Introduction

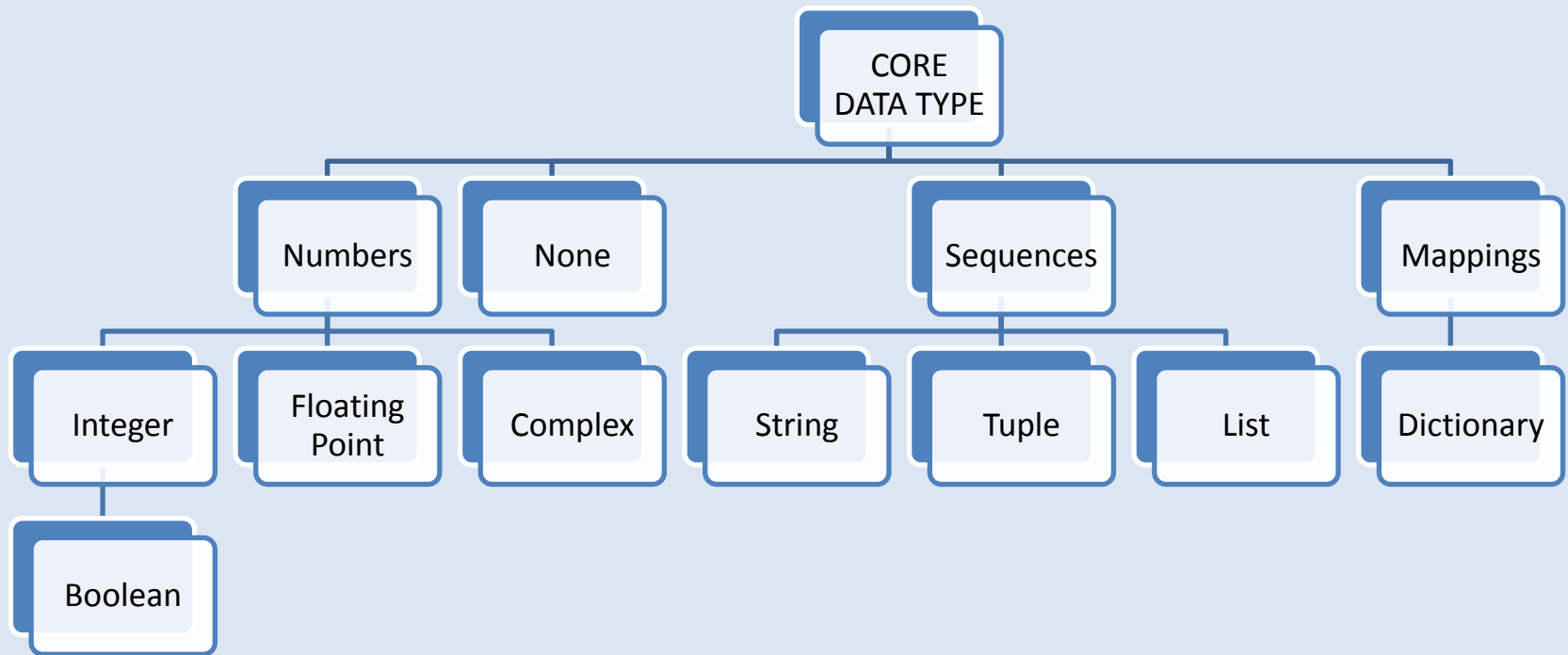
- In this chapter we will learn data types, variables, operators and expression in detail.
- Python has a predefined set of data types to handle the data in a program.
- We can store any type of data in Python.

DATA TYPES

- Data can be of any type like- character, integer, real, string.
- Anything enclosed in “ “ is considered as string in Python.
- Any whole value is an integer value.
- Any value with fraction part is a real value.
- True or False value specifies boolean value.
- Python supports following core data types-
 - I. Numbers (int like 10, 5) (float like 3.5, 302.24) (complex like 3+5i)
 - II. String (like “pankaj”, ‘pankaj’, ‘a’, “a”)
 - III. List like [3,4,5,”pankaj”] its elements are Mutable.
 - IV. Tuple like(3,4,5,”pankaj”) its elements are immutable.
 - V. Dictionary like {‘a’:1, ‘e’:2, ‘l’:3, ‘o’:4, ‘u’:5} where a,e,i,o,u are keys and 1,2,3,4,5 are their values.

CORE DATA TYPES

Graphical View



Mutable and Immutable Types

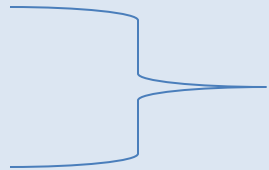
- In Python, Data Objects are categorized in two types-
 - Mutable (Changeable)
 - Immutable (Non-Changeable)

Look at following statements carefully-

p = 10

q = p

r = 10



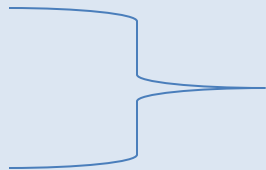
they will represent 10, 10, 10

Now, try to change the values-

p = 17

r = 7

q = 9



did the values actually change?

Answer is NO.

Because here values are objects and p, q, r are their reference name.

To understand it, lets see the next slide.

Variables and Values

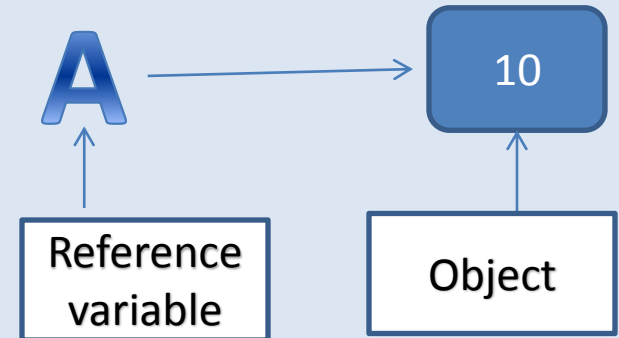
An important fact to know is-

- In Python, values are actually objects.
- And their variable names are actually their reference names.

Suppose we assign 10 to a variable A.

A = 10

Here, value 10 is an object and A is its reference name.



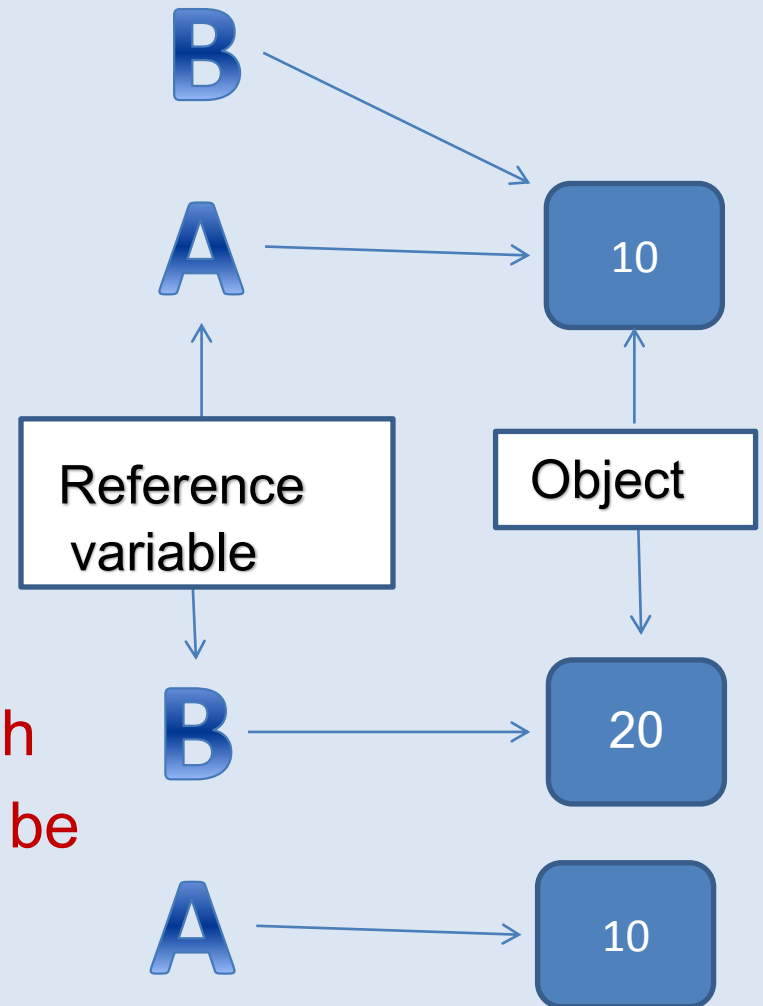
Variables and Values

If we assign 10 to a variable B,
B will refer to same object.

Here, we have two variables,
but with same location.

Now, if we change value of B like
B=20

Then a new object will be created with
a new location 20 and this object will be
referenced by B.



Mutable and Immutable Types

Following data types comes under mutable and immutable types-

- **Mutable (Changeable)**
 - lists, dictionaries and sets.
- **Immutable (Non-Changeable)**
 - integers, floats, Booleans, strings and tuples.

Variable Internals

The Type of an Object

- Pay attention to the following command-

```
>>> a=4
```

```
>>> type(4)
```

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

```
>>> type(a)
```

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

here 4 is an object and its class is int

here a is referring to the object which is of int class.

Variable Internals

The Value of an Object

- Pay attention to the following command-

```
>>> print(4)
4
>>> print(a)
4
```

here value output is coming via print()

The ID of an Object

- Pay attention to the following command-

```
>>> id(4)
1817668720
>>> id(a)
1817668720
>>> |
```

Here value 4 and variable a are showing same id which means 4 is an object being referenced by a that's why they are keeping same id.

Operators

- The symbols that shows a special behavior or action when applied to operands are called operators. For ex- + , - , > , < etc.
- Python supports following operators-
 - I. Arithmetic Operator
 - II. Relation Operator
 - III. Identity Operators
 - IV. Logical Operators
 - V. Bitwise Operators
 - VI. Membership Operators

Arithmetic Operators

- Python has following binary arithmetic operator -
 - For addition + for ex- $2+3$ will result in to 5
 - For subtraction – for ex- $2-3$ will result in to -1
 - For multiplication * for ex- $2*3$ will result in to 6
 - For division / its result comes in fraction.
for ex- $13/2$ will result in to 6.5
 - For quotient // its result comes as a whole number
for ex- $13/2$ will result into 6.
 - For remnant % its result comes as a whole remnant number.
For ex- $13/2$ will result into 1.
 - For exponent ** it will come as per exponent value.
For ex- 2^3 will result into 8.

Assignment Operators and shorthand

- Python has following assignment operator and shorthand -
 - = $a=10$, 10 will be assigned to a.
 - += $a+=5$ is equal to $a=a+5$.
 - -= $a-=5$ is equal to $a=a-5$.
 - *= $a*=5$ is equal to $a=a*5$.
 - /= $a/=5$ is equal to $a=a/5$.
 - //= $a//=5$ is equal to $a=a//5$.
 - %= $a%=5$ is equal to $a=a\%5$.
 - **= $a**=5$ is equal to $a=a**5$.

Relational Operators

- Python uses Relational operators to check for equality. These results into true or false. Relational Operator are of following types-

- < Less Than like $a < b$
- > Greater Than like $a > b$
- <= Less Than and Equal to like $a \leq b$
- >= Greater Than and Equal to like $a \geq b$
- == Equal to like $a == b$
- != not Equal to like $a != b$

Identity Operators

Identity operator is also used to check for equality. These expression also results into True or False. Identity Operators are of following types-

- “is” operator if a=5 and b=5 then a is b will come to True
- “is not” operator if a=5 and b=5 then a is not b will come to False
- Relational Operator (==) and Identity operator (is) differs in case of strings that we will see later.

Logical Operators

- Python has two binary logical operators -
 - or operator
 - » if a = True and b = False then **a or b** will return *True*.
 - and operator
 - » If a = True and b = False then **a and b** will return *False*.
- Python has one Unary logical operator –
 - not operator
 - if a = True then **not a** will return *False*.

Operator Associativity

- In Python, if an expression or statement consists of multiple or more than one operator then operator associativity will be followed from left-to-right.

```
>>> 7*8/5//2  
5.0
```

- In above given expression, first $7*8$ will be calculated as 56, then 56 will be divided by 5 and will result into 11.2, then 11.2 again divided by 2 and will result into 5.0.

*Only in case of ******, associativity will be followed from right-to-left.

```
>>> 3**3**2  
19683
```

Above given example will be calculated as $3^{(3^2)}$.

Expressions

- Python has following types of expression -
 - Arithmetic Expressions like $a+b$, $5-4$ etc.
 - Relational Expressions like $a>b$, $a==b$ etc.
 - Logical Expressions like $a>b$ and $a>c$, a or b etc.
 - String Expressions like “Pankaj” + “Kumar” etc.

Type Casting

- As we know, in Python, an expression may be consists of mixed datatypes. In such cases, python changes data types of operands internally. This process of internal data type conversion is called implicit type conversion.
- One other option is explicit type conversion which is like-
<datatype> (identifier)

For ex-

```
a="4"  
b=int(a)
```

Another ex-

If a=5 and b=10.5 then we can convert a to float.

Like d=float(a)

In python, following are the data conversion functions-

(1) int () (2) float() (3) complex() (4) str() (5) bool()

Working with math Module of Python

- Python provides math module to work for all mathematical works. We need to write following statement in our program-

```
import math
```

output of this program will be 5.0

```
import math
a=25
print (math.sqrt (a) )
```

```
>>> import math
>>> dir (math)
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

```
>>>
```

To get to know functions of a module, give following command-

```
>>>dir (math)
```

Taking Input in Python

- In Python, input () function is used to take input which takes input in the form of string. Then it will be type casted as per requirement. For ex- to calculate volume of a cylinder, program will be as-

```
#Program to calculate vloume of a cone
radius=int(input("Enter the radius of the Cylinder : "))
height=int(input("Enter the height of the Cylinder : "))
volume = 3.14* radius*radius*height
print("The Volume of the cylinder is : ",volume)
|
```

- Its output will be as-

```
>>>
RESTART: C:/Users/KVBBKServer/AppData/Local/Programs/Python/Python36/VolOfCyl
inder.py
Enter the radius of the Cylinder : 10
Enter the height of the Cylinder : 5
The Volume of the cylinder is : 1570.0
>>> |
```

Thank you

Please follow us on our blog-

www.pythontrends.wordpress.com