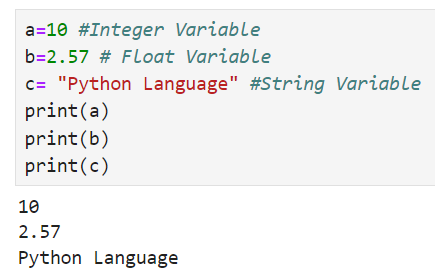
Day-6

6 November 2023

**Variable Assignment:**

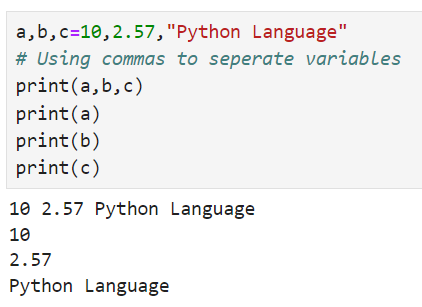
In Python, variable assignment is the process of storing a value in a variable. The syntax for assignment uses the equals sign (=).

**Example:**

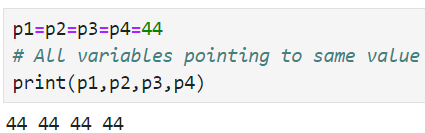


**Multiple Assignment:**

In Python, multiple assignment allows you to assign values to multiple variables in a single statement. This is done by separating the variables and corresponding values with commas. The number of variables must match the number of values being assigned.

**Example:**

Here, a is assigned the value 10, b is assigned 2.57, and c is assigned “Python Language” in one line.



The value 44 is assigned to all four variables: p1, p2, p3, and p4 in a single line. This works by assigning the value from right to left. First, 44 is assigned to p4, and then this value is assigned to p3, p2, and finally p1. As a result, all four variables reference the same value 44.

**Data Types:**

They specify what type of data you have. We have 3 types of data:

1. Numeric Type:

a) Integer Data – Belongs to all the whole numbers. It can be positive or negative. Eg:1, -1, etc.

b) Float Data – Contains decimal values. Eg:2.5, 4.2, 3.6, etc.

c) Complex Data – Contains real and imaginary values. Eg:20+5j {20 is real and 5j is imaginary}

1. Boolean Type:

Here we have only 2 values. That is, True and False

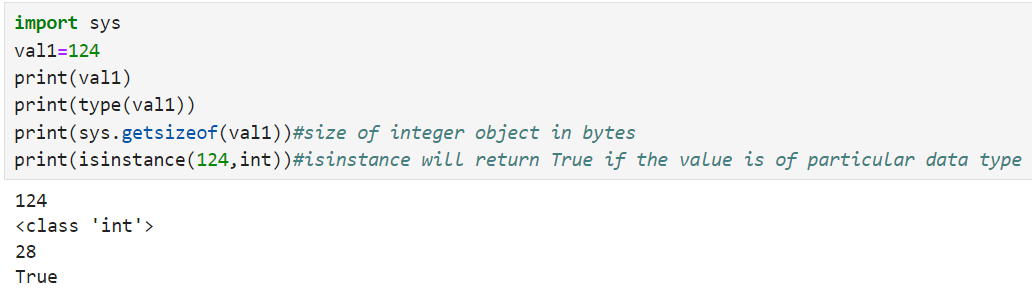
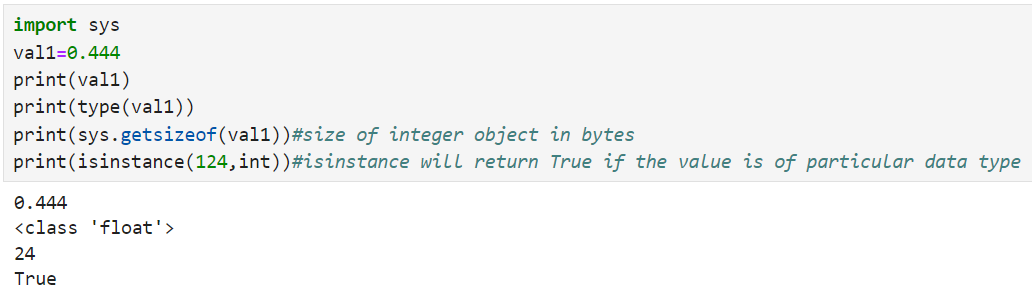
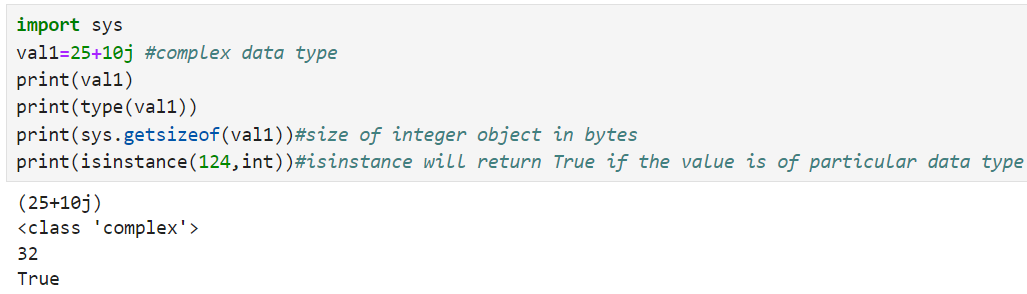
Where, True is denoted by 1 and

False is denoted by 0.

1. String Data Type:

It is a collection of characters.

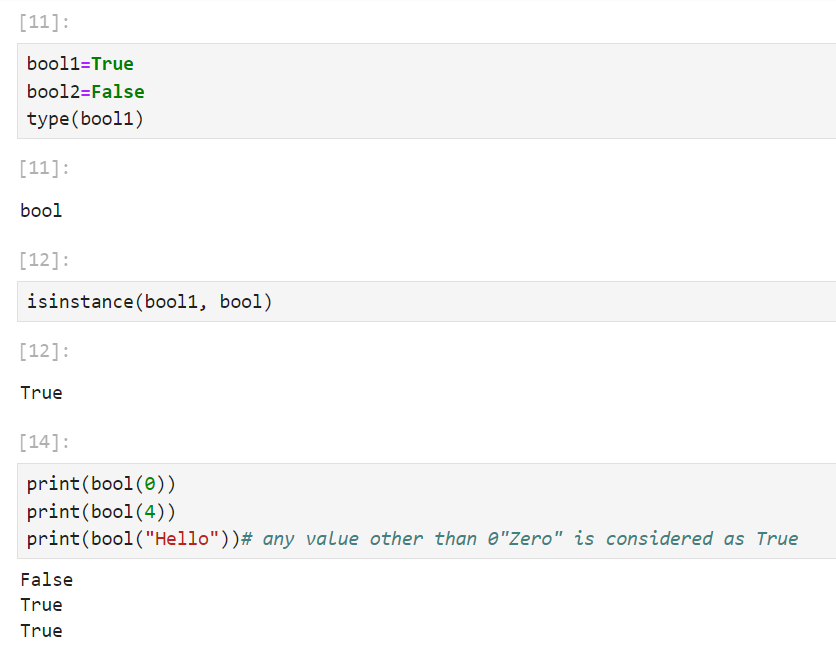
**Examples:**

1. **Numeric Type:**
2. Integer:
3. Float:
4. Complex Data:
5. **Boolean Type**

bool() is used to convert any value into boolean.

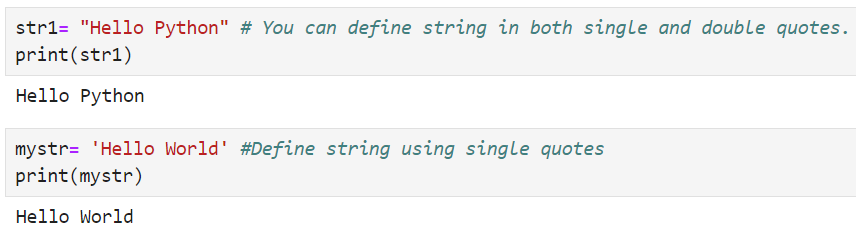
Int – bool

float – bool

****String – bool

1. **String Type:**

**String Creation:**

****

**String Indexing:**

Indexing means every character of string is given a number starting from 0 (zero). Using this numbers, we can access that particular character.

**Example:**

Mystr= “Helloworld”

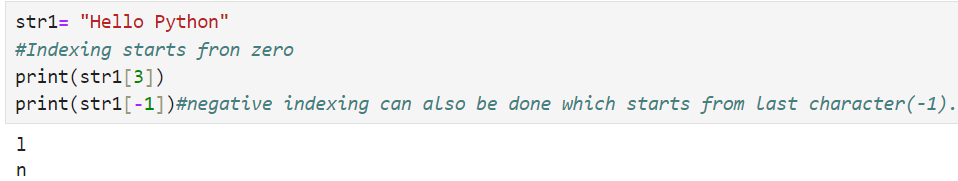
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| H | E | L | L | O | W | O | R | L | D |

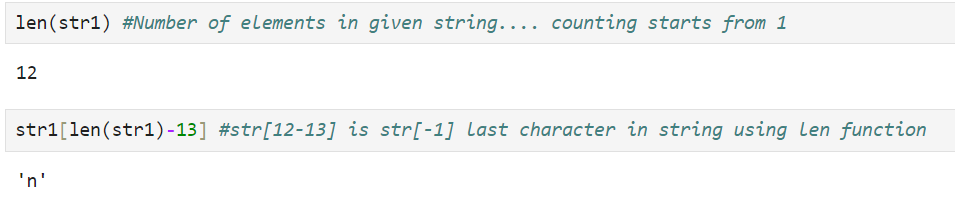
[0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

Print(mystr[3])

The output would be – ‘L’

By using string indexing, we can access any character based on the index.



****

**String Slicing:**

In Python, you can access a subset of a string by giving a range of indices by using string slicing. Slices let you take out a section of the string, called a "substring."

Syntax:

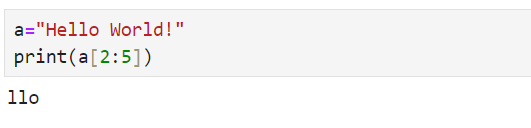
string[start:end:step]

start is the index where the slice begins (inclusive).

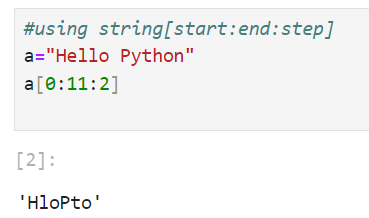
end is the index where the slice ends (exclusive).

step is an optional argument that determines the interval between characters (default is 1).

**Basic Slicing:**



This extracts the substring from index 2 to 4 (since the end index is exclusive).



**string[:]:**

It returns entire string as it is.

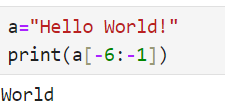


**Omitting start or end:**

****

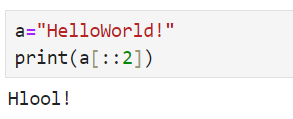
**Using negative Indices:**

Negative indices count from the end of the string

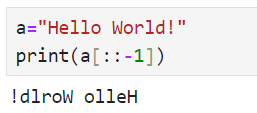
****

**Step in slicing:**

The step argument skips every second character.



**Reverse a string:**

****