Day: 8

9th November, 2023

**Python Operators:**

Operators are the special symbols used to perform operations on values/variables.

Types of Operators in Python:

1. **Arithmetic Operators:**

To perform mathematic operations.

**Basic Operators:**

* Addition Operator (+) eg:2+3 #Output=>5
* Subtraction Operator (-) eg:2-3 #Output=>-1
* Multiplication Operator (\*) eg:2\*3 #Output=>6
* Division Operator (/) eg:2/3 #Output=>0.6

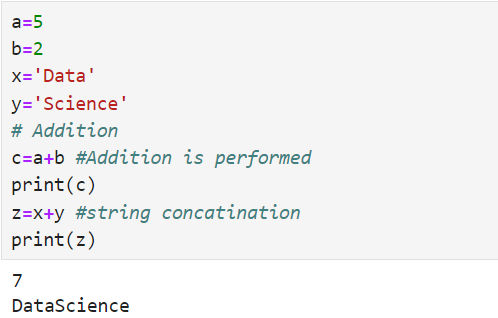
**Advanced Operators:**

* Modulus (%) eg:12%2 #Output=>0
* Floor Division (//) eg:4//3 #Output=>3
* Power Operator (\*\*) eg:4\*\*2 #Output=>16

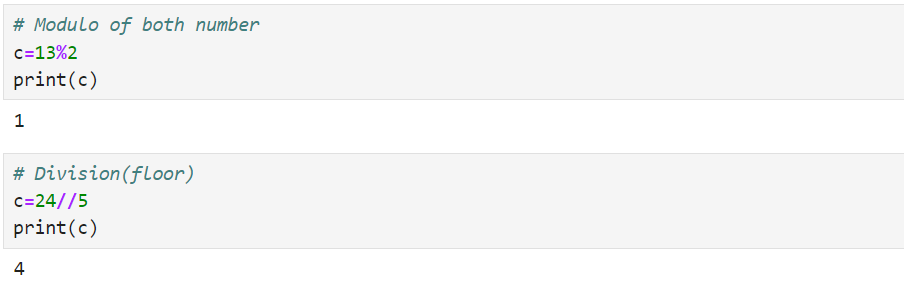
***Note: Modulus division returns remainder and Floor division returns quotient.***

**Floor division is also called integer division as it gives output in a rounded format.**

When you use + operator on string, it concatenates the string. And when you use + operator on numbers, addition is performed.

**Example:**



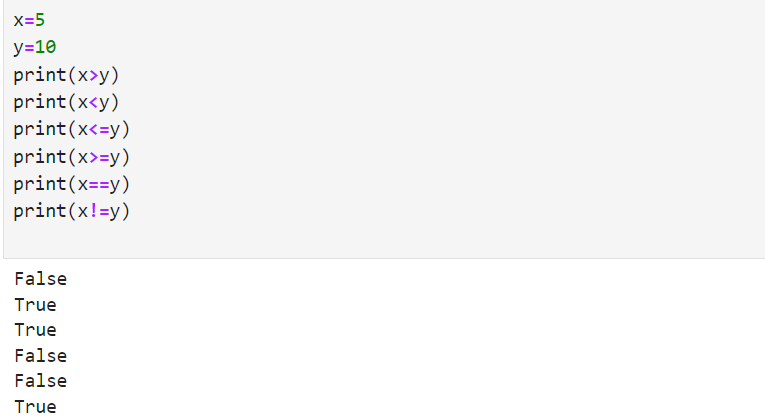


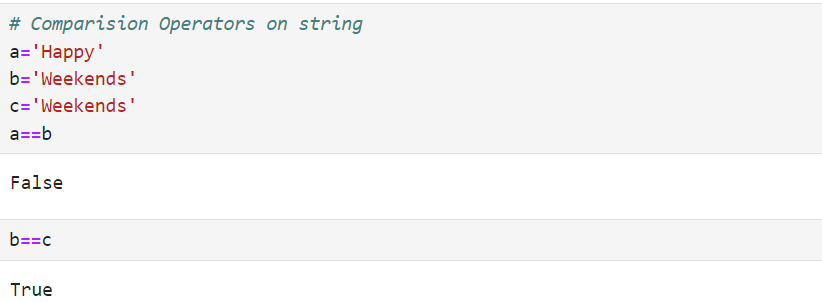
1. **Comparison Operators:**

These operators are used to perform comparison between 2 values and returns result in terms of True or False.

Types of comparison operators:

* < (Less than) eg:2<3 #Output=>True
* > (Greater than) eg:5>7 #Output=>False
* <= (Less than or equal to) eg:5<=3 #Output=>False
* >= (Greater than or equal to) eg:7>=5 #Output=>True
* == (Equals to) eg:3==3 #Output=>True
* != (Not equals to) eg:2!=1 #Output=>True

**** **Example:**

****

1. **Logical Operators:**

There are 3 logical operators in Python:

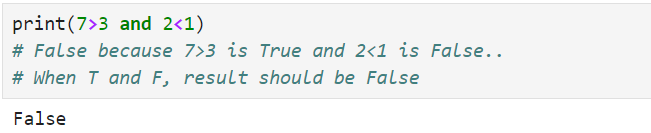
**and Operator:**

returns True of both values are True.

Truth Table:

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| **x** | **y** | **x and y** |
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |

**Example:**

****

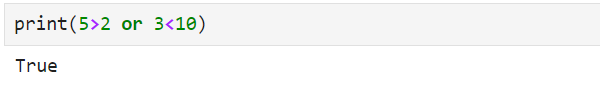
**OR Operator:**

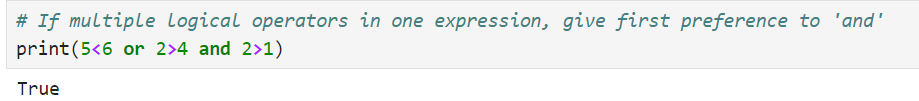
returns True if at least one input is True. Otherwise, returns False.

Truth Table:

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| **x** | **y** | **x and y** |
| True | True | True |
| True | False | True |
| False | True | True |
| False | False | False |

**Example:**

****

****

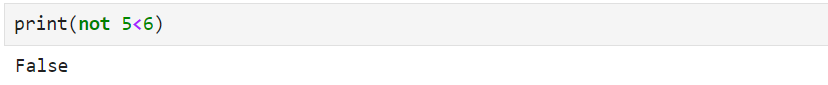
**NOT Operator:**

True if operand is False.

Truth Table:

|  |  |
| --- | --- |
| **Input** | **Output** |
| A | not A |
| False | True |
| True | False |

**Example:**

****

**Control Flow Statements:**

1. **If Conditional Statements:**

They are used to evaluate a condition whether it is True or False.

If it is True, then the statements under if will be executed. If it is False, then statements after if will be immediately executed.

**Syntax:**

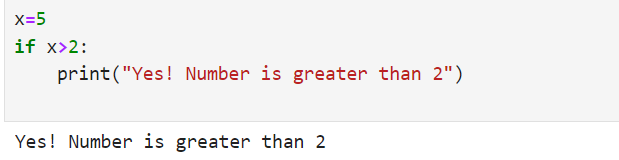
if(condition):

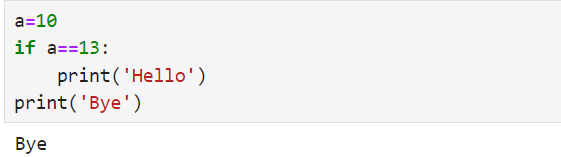
//statements

*Note: Space called Indentation is very important in Python.*

Indentation refers to the spaces at the beginning of a code line. It is very important as Python uses indentation to indicate a block of code.

**Example:**

****

The variable ‘a’ is set to 10 in the Python ****code. The if statement determines whether ‘a’ equals 13. The if statement's print('Hello') block is ignored as this condition is false (since an is 10, not 13). As a result, only the print('Bye') line is executed, producing the output "Bye".

1. **if-else Statement:**

The Python if statement comprises of a piece of code that only executes when the if statement’s condition is TRUE. If the condition is FALSE, the else statement, which includes code for the else section is executed.

**Syntax:**

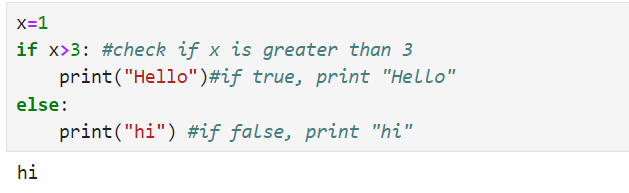
if(condition):

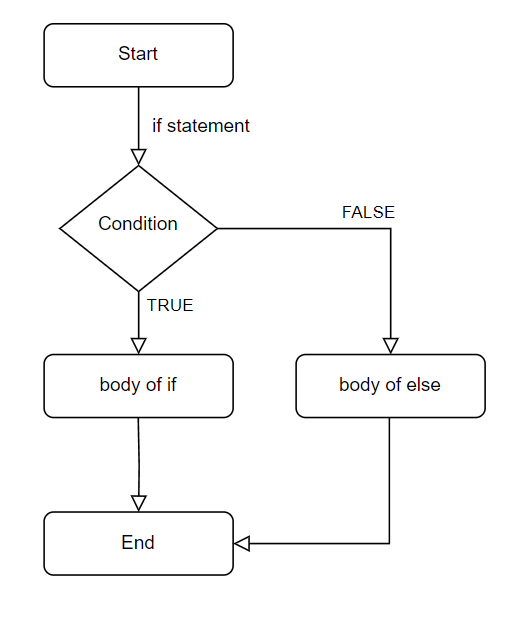
//statements

else:

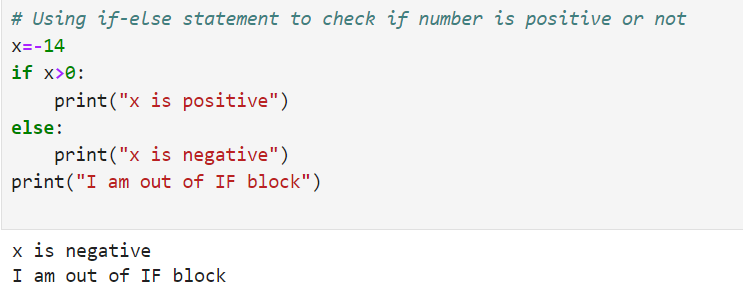
//statements

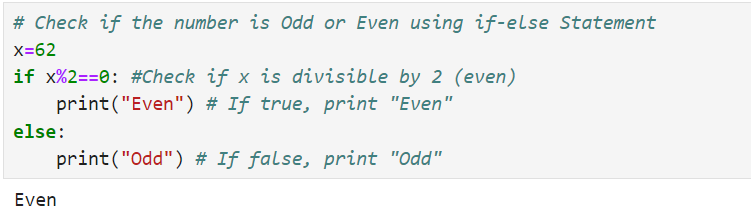
**Example:**

****



**Fig: Flowchart of if-else Statement**

****

****

1. **if-elif Statement:**

This is used to check multiple conditions. If one condition is not True, it will check for another condition, if that condition is also not True, then statements under else will execute.

**Syntax:**

if(condition):

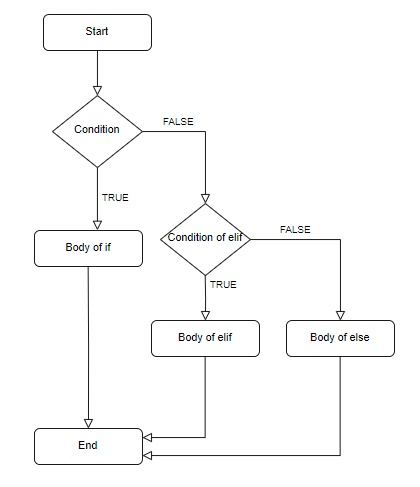
//statements

elif(condition):

//statements

else:

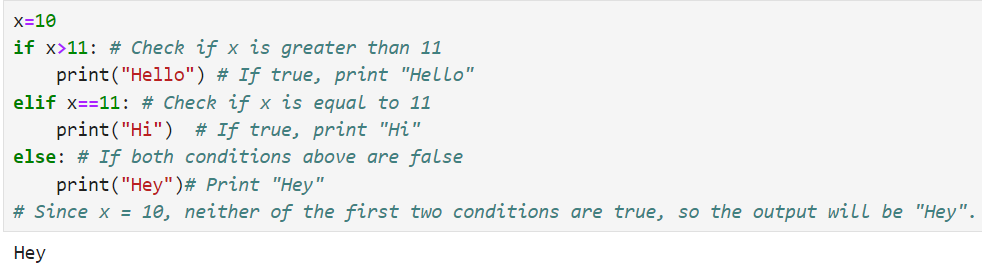
//statements

****

**Fig: Flowchart of if-elif-else Statement**

**Example:**

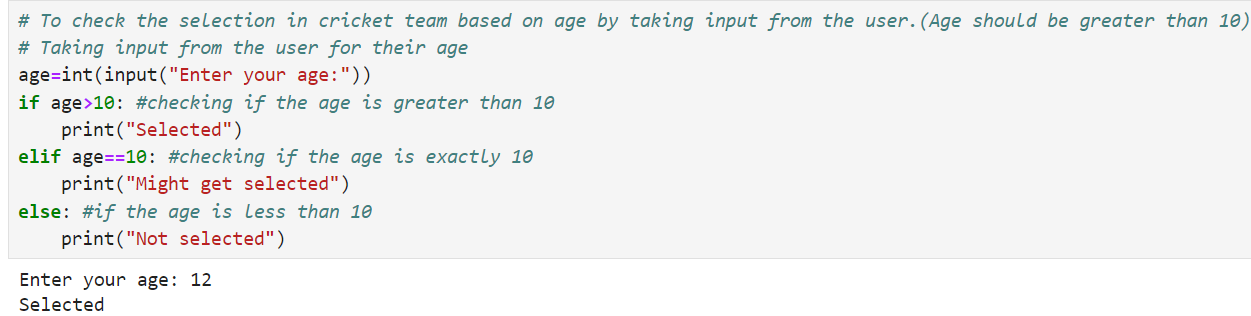
1. Simple use of if-elif-else statement.

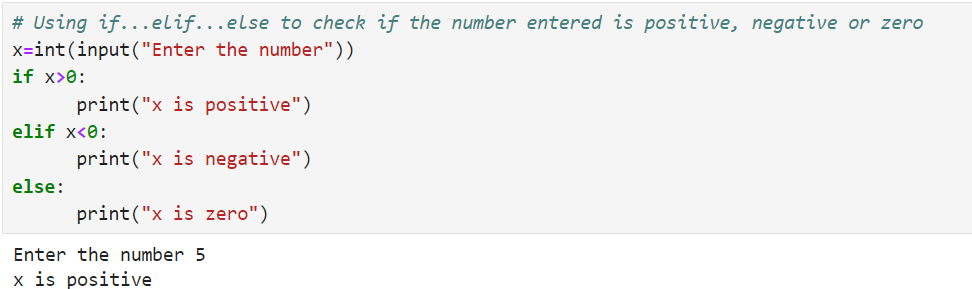
****

1. Taking input from the user.

input() is used to take user input.

Whenever we take input using input method, that value will be taken as string value. If you want to treat the input as a number, you need to convert it from a string to integer using int() in front of a input as shown in an example below:

****

****