Arithmetic Operators in Python:

Operators are special symbols in Python that carry out arithmetic or logical computation.

Arithmetic operators are used to perform mathematical operations like *addition, subtraction, multiplication and division*.

Operands: The value that the operator *operates on* is called the operand.

For e.g. 2 + 3, Here "+" is an operator, while 2 & 3 are operands.

Assume, a = 2

b = 3

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	a + b = 5
- Subtraction	Subtracts right hand operand from left hand operand.	a – b = -1
* Multiplication	Multiplies values on either side of the operator	a * b = 6
/ Division	Divides left hand operand by right hand operand	b / a = 1.5
% Modulus	Divides left hand operand by right hand operand and returns remainder	b % a = 1
** Exponent	Performs exponential calculation on operators	a**b =2 to the power 3 = 8
// Floor Division	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed.	a // b = 0 b // a = 1

These Arithmetic Operators was the one of the type of operators in Python,

It also does have other **types of operators** as follows:

- Arithmetic Operators
- Assignment Operators
- Bitwise Operators
- Comparison (Relational) Operators
- Identity Operators
- Logical Operators
- Membership Operators

We are going to learn these operators on the go, as we proceed through the course in upcoming sessions.

Functions in Python:

Functions are a convenient way to divide your code into useful blocks, allowing us to order our code, make it more readable, reuse it and save some time. Functions provide better modularity for your application and a high degree of code reusing. It is a block of organized, reusable code that is used to perform a single, related action.

We do have many built-in functions like print() in Python, but we can also create our own functions. These functions are called user-defined functions.

Syntax:

```
def funName():        ←          Function Definition

        --------------------------  ⎫
        --------------------------  ⎬   Your code goes here (Body of function)
        --------------------------  ⎭

funName()             ←          Function Calling
```

Example:

```
        def greet():
                print("Hello there, greetings of the day!")
        greet()
```

Output:

```
Hello there, greetings of the day!
```

Addition Function:

```
        a = 10
        b = 20
        def add():
                c = a + b
                print(c)
        add()  //Calling add() function
```

Output:

```
                30
```

This kind of functions are called as non-parameterized function, since we're not passing any kind of parameters (arguments) to the function. In the live session we'll cover about the Parameterized functions, which can take arguments.

Note:

Function Calling is must for its execution. Function will execute if and only if, it is getting called. Until and unless we call the function, it is never going to execute on its own.

The code block of function definition is skipped or ignored during execution by Python Interpreter, and it gets executed once the function is called.

Getting User Input in Python:

Till now we've used hardcoded values for variables such as a = 10, b = 20 etc. Now we'll learn about getting these values from a user at runtime.

Python's built-in function input() is used to take user input in Python. We have to store the value got from user into a variable for further use. We will look at one example below:

Example:

```
name = input(“Please Enter Your Name:”)

print(“Hello”, name)
```

Output:

```
Hello Vaibhav
```

(Assuming if we enter “Vaibhav” as input)

Getting Integer Input in Python:

By default, the input function gets the user input in string format, so we've to typecast them into appropriate data type according to our need.

Lot of technical jargon here! Don't worry, we are going to explore it in depth, for the time being we'll just focus on how to get a integer (number) from user at run time.

Syntax:

```
var = int(input(“Your Prompt-Message Here”))
```

Example:

```
num1 = int(input(“Please Enter First Number:”))

num2 = int(input(“Please Enter Second Number:”))

print(“You have entered”, num1, “as first number and”, num2, “ as second.”)

tot = num1 + num2

print(“Addition of ”, num1, “ and”, num2, “is =”, tot)
```

Output:

```
Please Enter First Number:5
Please Enter Second Number:3
You have entered 5 as first number and 3  as second.
Addition of  5  and 3 is = 8
```