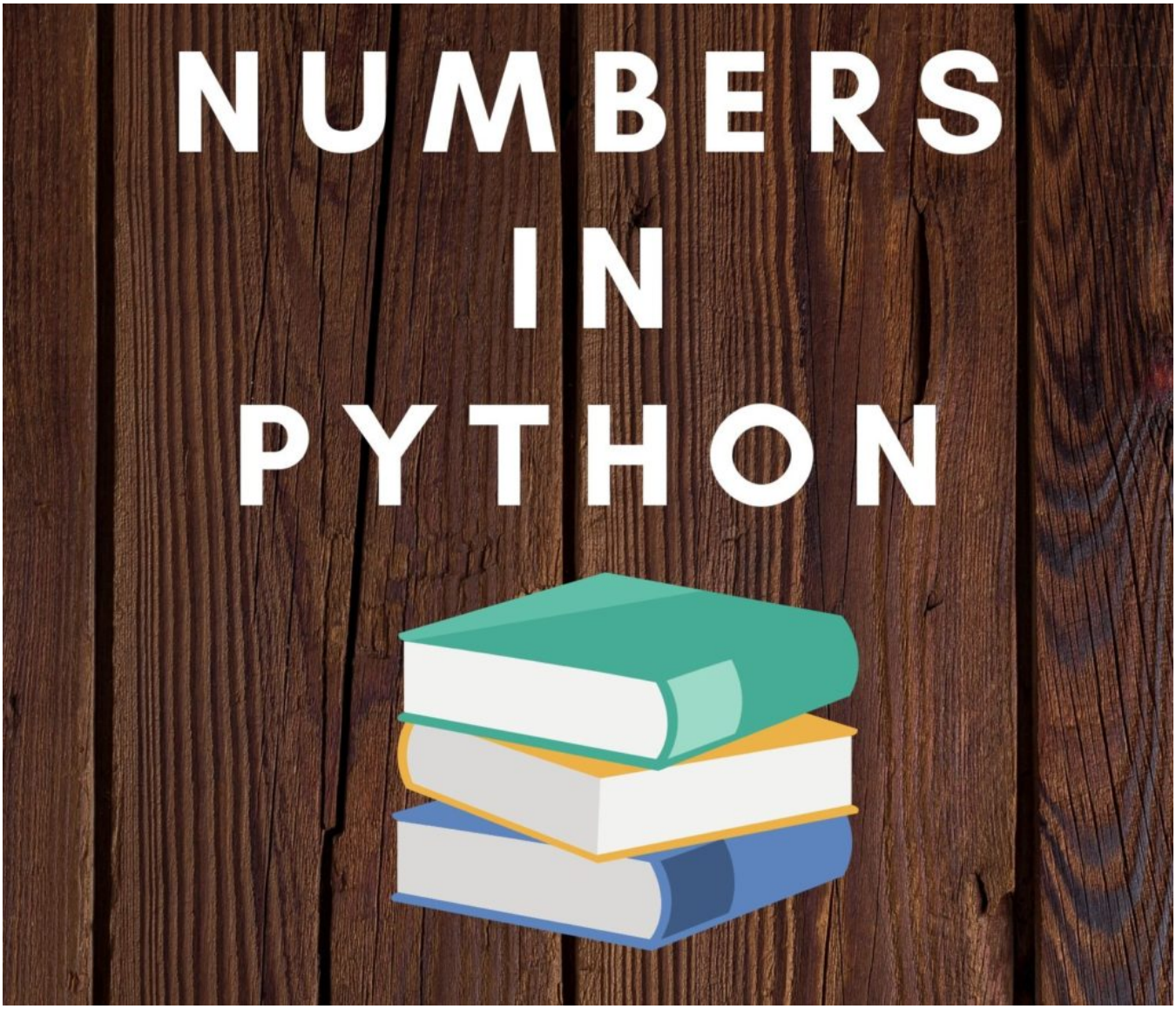
Python Tutorial: Numbers in Python

Python / March 9, 2020



This is Python Tutorial: Numbers in Python. In this Python Tutorial, you will learn the numbers in python language. At the end of this tutorial, you will have full knowledge of

numbers.

Hello, & Welcome!

Part 2- Python Tutorial: Numbers in Python

In this article of Python Tutorial, you will learn following-

1. Types of Numbers.
2. Type Conversion
3. Basic Calculations using Python

Types of Numbers-

In Python, there are basically three types of numbers-

- **Integer Type-** The Integer type is a whole number. It may be positive or negative. It doesn't contain a decimal.

For Example-

```
a=5 #int
b= 1079876509 #int
c= -673409 #int
```

- **Float Type-** It contain a decimal, and it may be positive or negative.

For Example-

```
a= 5.20 #float
b=5.0 #float
c= -67.83 #float
```

- **Complex Type-** This type of numbers are in the form of a+bJ, where J is the square root.

```
a= 9+8j # complex
b= 9j #complex
c= -4j #complex
```

To check the type of any number, you just need to type-

```
print(type(a))
```

and it will return the type of number. For example if you want to check the type of number of a=5 so, you can check it like that-

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

OUTPUT-> <class 'int'>
```

Type Conversion-

If you want to convert your number from one type to another, for example, int to float, float to complex, etc. you can simply do it.

Have a look How you can convert a type of number-

```
a= 6 #int
b=7.9 #float
c= 7+5j #complex

#from int to float-
x= float(a)

#from float to int-
y= int(b)
```

```
#from float to complex-
z= complex(b)
```

See, it is as simple as that, just by writing one line of code, you can convert from one type to another.

Note- You can't convert complex numbers to any other type.

Basic Calculations using Python-

After getting proper knowledge of types of numbers and conversion, its time to practice some basic calculations by python.

So, are you ready? Let's get started by simple maths-

- For simple **addition**, **subtraction**, **multiplication** and **division**, you just need to type as you type in the calculator. Very easy...right? Let's see, how to do it-

Write this code in your cmd, to practice with me-

```
>>> 5+5   #addition
10
>>>5-5    #substraction
0
>>>5*5    #multiplication
25
>>>5/5   #division
1.0 # float type
```

-> As you see in the division it is returning the result in a float as **1.0**. To get the result in int, you need to type it as-

```
>>>5//5   #division
1   #int type
```

-> For the **square root** of number-

```
>>>5**5
3125
```

-> For **remainder/modulus**–

```
>>>10%3
1
```

-> **BIDMAS Rule**–To perform longer calculation, you need to take care of BIDMAS Rule. Let's see an example, how it affects the whole calculation.

```
>>>5+5*3
20
```

- Here it is returning 20 as a result, just because * has more priority than + according to BIDMAS Rule.
- If you want to do addition first, put it into a bracket, because in BIDMAS Rule a bracket has more priority than *
- Let's see in this example, how result is changed just by using ().

```
>>>(5+5)*3
30
```

Store number in variables-

-> You can store numbers on variables and perform a calculation on it. Let's see how you can do it-

```
>>>age=25   #store number in variable "age"
>>>age      #print "age"
```

```
25          #OUTPUT

#perform addition on "age"
>>>age+5
30
```

->"age+5", simply add 5 to your age variable whose value is 25, but it doesn't alter the variable "age" value. If you enter "age", it returns 25 as a result. I'll show you here-

```
>>>age
5         #print 25 again
```

->If you want to update the value of "age", then you can simply do it by writing-

```
>>>age= age+5    #add 5 in variable "age", not direct.
>>>age           #print age
30              #OUTPUT
```

-> There is **one more way to update the value of variable "age"**. Have a look at it too-

```
>>>age+=5       #add 5 in variable "age", not direct.
>>>age          #print age
30              #OUTPUT
```

-> Just like addition, you can perform other calculations on a variable, by just writing a few lines of code. Let's have a look-

```
>>>age=25
>>>age-=5  #subtraction
>>>age
20
```

```
>>>age/=2   #division
>>>age
15.0
```

I hope, now you have a clear understanding of numbers in python. Now it's time to perform some high-level calculations.

Let's see one example-

```
>>>wages=1000
>>>bills=200
>>>rent=500
>>>food=200
>>>savings= wages-bills-rent-food    #formula to find saving
>>>savings
100
```

->In this example, we are calculating the total saving of month, and look it is how much simple in python, just put all values in variables, apply a formula, and get the results.

And here we go, congratulations! You successfully learned numbers in python.

In the next tutorial, we will start learning **Python Strings**.

You can simply go to the next Tutorial from Here.

Enjoy Learning Python!

All the Best!

Are you ML Beginner and confused, from where to start ML, then read my BLOG – How do I learn Machine Learning?

If you are looking for Machine Learning Algorithms, then read my Blog – Top 5 Machine Learning Algorithm.

If you are wondering about Machine Learning, read this Blog- What is Machine Learning?

```
Thank YOU!
```

Though of the Day...

> *'I am always ready to learn although I do not always like being taught."*
>
> *– Winston Churchill*

← Previous Post Next Post →

1 thought on "Python Tutorial: Numbers in Python"

Pingback: Python Tutorial: Complete Guide for Beginners (2020)

Leave a Comment

Your email address will not be published. Required fields are marked *

Type here..

Name*

Email*

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment »

Follow Us on Twitter-

Follow us on Pinterest-

Our Facebook Group-