

Artificial Intelligence/Machine Learning UpSkills Notebook

From Basics to Real-World — Starts Your ML Journey

Python Essentials: Next Steps Beyond the Basics

Welcome! In this notebook, you will learn the basics of using Python.

We will cover:

- How to use Python as a calculator
- Working with numbers
- Working with text (strings)
- Using lists
- First steps towards programming (variables, input, conditions)

Let's get started!

Using Python as a Calculator

Python can do simple math like a normal calculator.

```
In [56... # Addition  
2 + 3
```

```
Out[56]: 5
```

```
In [57... # Subtraction  
7 - 2
```

```
Out[57]: 5
```

```
In [58... # Multiplication  
4 * 5
```

```
Out[58]: 20
```

```
In [59... # Division  
10 / 2
```

```
Out[59]: 5.0
```

```
In [60... # Exponent (Power)  
2 ** 3
```

```
Out[60]: 8
```

```
In [61... # Floor Division  
17 // 3
```

```
Out[61]: 5
```

```
In [62... # Modulus (Remainder)  
17 % 3
```

```
Out[62]: 2
```

Use parentheses to control the order:

- **Without parentheses**

```
In [63]: 2 + 3 * 4
```

```
Out[63]: 14
```

- **With parentheses**

```
In [64]: (2 + 3) * 4
```

```
Out[64]: 20
```

Working with Numbers

In Python, numbers are a basic data type used to store numeric values.

Python can work with different kinds of numbers to perform calculations, just like a calculator.

The main types of numbers are:

- **Integers (int):** Whole numbers without decimals, like 5, -12, 0.

Example:

```
In [65]: # Integer
a = 10
```

- **Floating Point Numbers (float):** Numbers with decimal points, like 3.14, -2.5.

Example:

```
In [66]: # Float
b = 3.14
```

- **Complex Numbers (complex):** Numbers with a real and imaginary part, like 2 + 3j (used in advanced math and engineering).

Example:

```
In [67]: # Complex
c = 2 + 3j
```

```
In [68]: print(type(a), type(b), type(c))
```

```
<class 'int'> <class 'float'> <class 'complex'>
```

Python makes it easy to do arithmetic, rounding, and conversions with these numbers. They're the building blocks for all calculations in Python.

Converting Numbers

Converting numbers in Python means changing a number from one type to another.

For example, you might convert:

a float to an integer (removing the decimal part)

an integer (int) to a floating point number (float)

a number to a complex number

Example:

```
In [69]: x = 5.8
print(int(x)) # Float to int
```

```
5
```

```
In [70]: print(float(3)) # Int to float
```

```
3.0
```

```
In [71]: print(complex(4)) # Int to complex
```

```
(4+0j)
```

Built-in Math Functions

A built-in function is a function that comes ready-made with Python — you don't need to create it yourself.

These functions help you do common tasks quickly, like:

Doing math (`abs()`, `round()`, `pow()`)

Working with text (`len()`, `print()`)

Converting data types (`int()`, `float()`)

You just call the function by its name and pass it the value you want to work with.

Example:

```
In [72... import math

print("Square root:", math.sqrt(16))
print("Absolute value:", abs(-7))
print("Round:", round(3.14159, 2))
print("Power:", pow(2, 4))
```

```
Square root: 4.0
Absolute value: 7
Round: 3.14
Power: 16
```

Text

Text is called a **string** in Python.

You can store text in variables and use it.

Example:

```
In [73... name = "Alice"
print(name)

# Concatenation (joining text)
greeting = "Hello, " + name
print(greeting)

# Repeating text
echo = "Hi! " * 3
print(echo)
```

```
Alice
Hello, Alice
Hi! Hi! Hi!
```

Some examples of common String Functions

```
In [74... text = "Python is fun!"

# Length of string
print(len(text))
```

```
14
```

```
In [75... # Lowercase and Uppercase
print(text.lower())
```

```
python is fun!
```

```
In [76... print(text.upper())
```

```
PYTHON IS FUN!
```

```
In [77... # Replace text
print(text.replace("fun", "awesome"))
```

```
Python is awesome!
```

Lists

- A list stores multiple items in one variable.
- Lists can hold numbers, text, or mixed data.
- Items in a list are ordered by index (starting at 0).
- Lists are written inside square brackets [].
- Lists can be changed (items can be added, removed, or updated).
- A list can contain other lists (nested lists).
- Lists have many built-in methods to work with data.
- Lists make it easy to organize and process groups of data.

Examples:

```
In [78... # Create a list
fruits = ["apple", "banana", "cherry"]
print(fruits)
```

```
['apple', 'banana', 'cherry']
```

```
In [79... # Access by index
print(fruits[0]) # First item
print(fruits[-1]) # Last item
```

```
apple
cherry
```

```
In [80... # Change a value
fruits[1] = "mango"
print(fruits)
```

```
['apple', 'mango', 'cherry']
```

```
In [81... # Add an item
fruits.append("orange")
print(fruits)
```

```
['apple', 'mango', 'cherry', 'orange']
```

```
In [82... # Remove an item
fruits.remove("apple")
print(fruits)
```

```
['mango', 'cherry', 'orange']
```

```
In [83... # Length of list
print(len(fruits))
```

```
3
```

First steps towards programming

Combine what you learned!

- Use variables
- Take input
- Make decisions with `if`

Example:

```
In [84... # Variables
age = 18

# Input (try it in your notebook!)
# name = input("What is your name? ")
# print("Hello,", name)

# Conditional
if age >= 18:
    print("You are an adult.")
else:
    print("You are not an adult.")
```

```
You are an adult.
```

Summary

You learned:

- How to use Python as a calculator
- How to work with numbers, text, and lists

- How to write basic code with variables and conditions

Connect @

Mail (Isha Dhiman): dhimanisha177@gmail.com

Contact: [+91-9253165167](tel:+91-9253165167)

Isha Dhiman

AI/ML Enthusiast | Intern @CodroidHub | First-Year Engineering Student

Isha Dhiman