Machine Learning

Python Basics – Foundations for Machine Learning

Unit I • Programming Fundamentals

Class Objective

By the end of this class, students will be able to:

- Write and run basic Python programs
- Understand variables and data types
- Use arithmetic, comparison, and logical operators
- Perform type conversion and string operations
- Use common built-in Python functions
- Practice coding using Jupyter Notebook / Google Colab

Why Python for Machine Learning?

- Simple and readable syntax
- Beginner-friendly language
- Large ecosystem of libraries

Popular Python Libraries for DS

- NumPy
- Pandas
- Matplotlib / Seaborn
- Scikit-learn
- TensorFlow / PyTorch

Python Basics: First Program

A Python program can be written in just one line:

```
print("Hello, Machine Learning!")
```

Checking Your Python Version

```
import sys
print("Python version:", sys.version)
```

Why this is important:

- Libraries may require specific versions
- Helps debug compatibility issues

Variables in Python

Variables store data values:

```
name = "Sita"
age = 21
gpa = 3.75
is_student = True
```

- Python automatically assigns data types
- No need for explicit type declaration

Variable Naming Rules

✓ Start with a letter or underscore
✓ Case-sensitive (age ≠ Age)
✓ Cannot use Python keywords
✓ Use descriptive names

❌ Invalid:

```
2name = "Ram"
class = "DS"
```

✓ Valid:

```
student_name = "Ram"
total_marks = 85
```

Python Data Types (Primitive)

Data Type	Example
int	42
float	3.14
str	"Hello"
bool	True, False
NoneType	None

Checking Data Types

age = 25
print(type(age))

Output

```
<class 'int'>
```

Python Arithmetic Operators

Addition (+)
Subtraction (-)
Multiplication (*)
Division (/)
Floor Division (//)
Modulus (%)
Exponentiation (**)

Arithmetic Operator Examples

a = 10
b = 3

```
print(a + b)      # 13
print(a / b)      # 3.333...
print(a // b)     # 3
print(a % b)      # 1
print(a ** b)     # 1000
```

Comparison Operators

Used to compare values:

- == equal
- != not equal
- <, >, <=, >=

```
x = 5
y = 10

print(x == y)   # False
print(x < y)    # True
```

Logical Operators

Used to combine conditions:

- and
- or
- not

```
print((x != y) and (x < y))  # True
```

Type Conversion

Convert data from one type to another:

```
age = int("25")
price = float("19.99")
score = str(95)
```

Why needed:

- User input is usually string
- ML models need numeric data

String Operations

Concatenation

```
first_name = "Ram"
last_name = "Sharma"
full_name = first_name + " " + last_name
print(full_name)
```

f-Strings (Recommended)

```
message = f"{first_name} is a student"
print(message)
```

String Indexing

```
word = "Python"

print(word[0])    # P
print(word[-1])   # n
```

- Index starts at 0
- Negative index counts from end

Built-in Python Functions

Common functions:

- print()
- type()
- len()
- abs()
- round()
- max()
- min()

```
print(type(42))
print(len("Python"))
print(abs(-10))
print(round(3.7))
```

Hands-On Exercise #1

Tasks:

- Print a welcome message
- Create variables (name, age, GPA)
- Perform arithmetic operations
- Practice type conversion

Hands-On Exercise #2

Tasks:

- Write boolean expressions
- Perform string operations
- Use f-strings
- Apply built-in functions

Common Beginner Mistakes

❌ Indentation errors
❌ Mixing data types

"Age: " + 21 # Error

❌ Using keywords as variables
❌ Case sensitivity issues

Knowledge Check

1. List three stages of Machine Learning workflow
2. Name three Python libraries for Machine Learning
3. What is the data type of 3.14?
4. What does // operator do?
5. Convert "100" into an integer

Next Class

Python Programming Basics & Operators

(Control Flow, Conditions, Loops)