Advance Python Programming

Trainer: Fawad Bahadur

Introduction to Python

What Is Python?

- ✓ Python is a high-level, interpreted, general-purpose programming language.
- ✓ Known for its **clear syntax**, **readability**, and **simplicity**.
- ✓ Used by **beginners** and **experts** alike due to its **versatility**.

Why Learn Python?

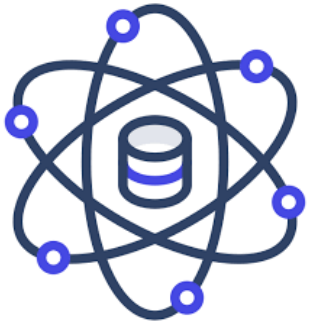
- **Beginner-friendly** yet **powerful** for complex applications.
- **Widely adopted** by **Google, NASA, Netflix, and more**.

Conti...

Key Features

- **Easy to Learn & Write** – Minimalistic syntax, resembles English.
- **Rich Standard Library** – Pre-built modules for faster develop
- **Strong Community Support** – Active forums, tutorials, and open-source contributions.

- **Versatile** – Used in:
  - ✓ **Web Development** (Django, Flask)
  - ✓ **Data Science & AI** (Pandas, TensorFlow)
  - ✓ **Automation & Scripting**
  - ✓ **Game Development** (Pygame)


Key characteristics

High-Level — Simple, abstracted coding

Object-Oriented — Classes, objects support

Interpreted — Runs without compilation

General-Purpose — Versatile, multi-domain use

Conti...

Extensible Integrates external code

Vast Ecosystem Rich libraries, tools

Dynamically Typed Flexible variable types

Cross-Platform Runs on all OS

Large Community Active, supportive network

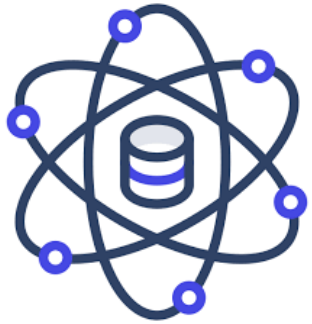
Python Programming

Applications

- Web Development
- Data Science
- Data Analytics
- Machine Learning & AI
- Automation
- Software Development and Prototyping
- Desktop GUI applications
- Scientific and Numeric Computing
- Game Development
- Education and Research





Download & Install Python

1

Download Python

Visit **python.org/downloads**
Choose the **latest stable version** (e.g., Python 3.12).

2

Run the Installer

Windows/macOS: Double-click the downloaded .exe (Windows) or .pkg (macOS).

3

Setup Options (Windows/macOS)

Check "Add Python to PATH" (Critical for command-line use).
Optional: Install pip (Python's package manager).

4

Verify Installation

Open **Command Prompt/Terminal** and type:
python --version

Trainer: Fawad Bahadur Marwat

Install Anaconda for Python

Why Anaconda?

- Includes Python, Jupyter Notebook, and key libraries.
- Simplifies setup for immediate coding.

1. Download Anaconda

- Visit anaconda.com/download
- Choose **Python 3.x** for your OS (Windows/macOS).

2. Run the Installer

- **Windows:** Double-click .exe → Follow prompts.
- **macOS:** Open .pkg → Drag to Applications.

3. Key Installation Options

- Add Anaconda to PATH (Recommended 'easy CLI access').
- **Register Anaconda as default Python** (Optional but useful).

4. Verify Installation

- Open **Command Prompt/Terminal**, run:
- conda --version

Trainer: Fawad Bahadur Marwat

Set Up Jupyter Notebook

1

Open Anaconda Navigator

Launch from your system after installation.

2

Start Jupyter Notebook

In Navigator, click the Jupyter Notebook icon.

3

Create a New Notebook

Opens in your web browser.
Click New > Python 3 in top-right.

4

Start Coding

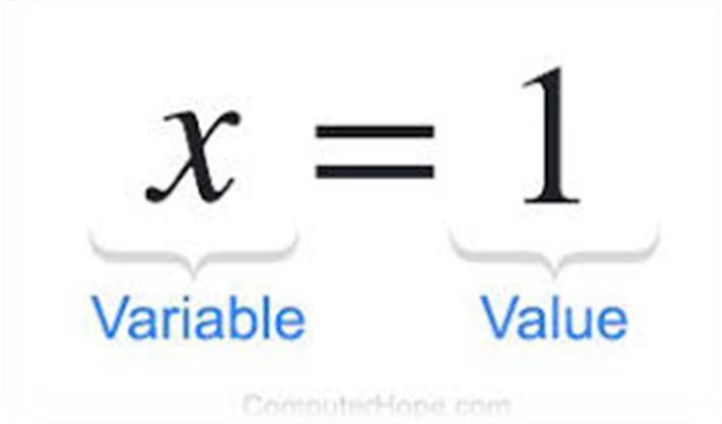
Write Python code in notebook cells.
Press Shift + Enter to execute.

Trainer: Fawad Bahadur Marwat

Variables in Python

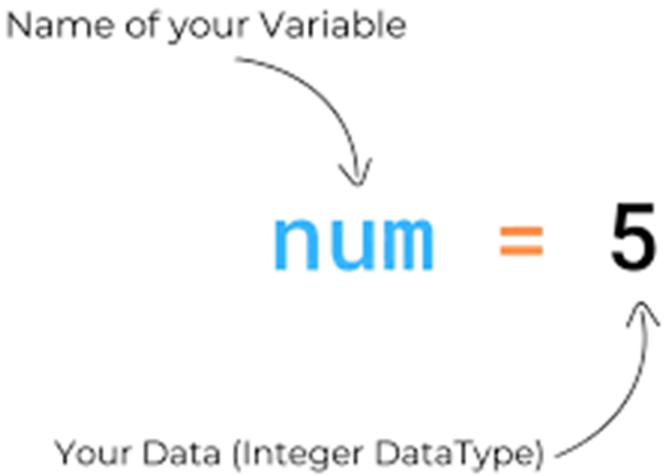
What is a Variable?

- A named container to store data in memory.
- Holds values that can be changed during program execution.
- No need to declare type



Key Rules

Names are case-sensitive (x ≠ X).
Can include letters, digits, _ (no spaces).
Cannot start with a digit.
Avoid Python keywords (if, for, etc.).



Syntax

```
variable_name = value
```

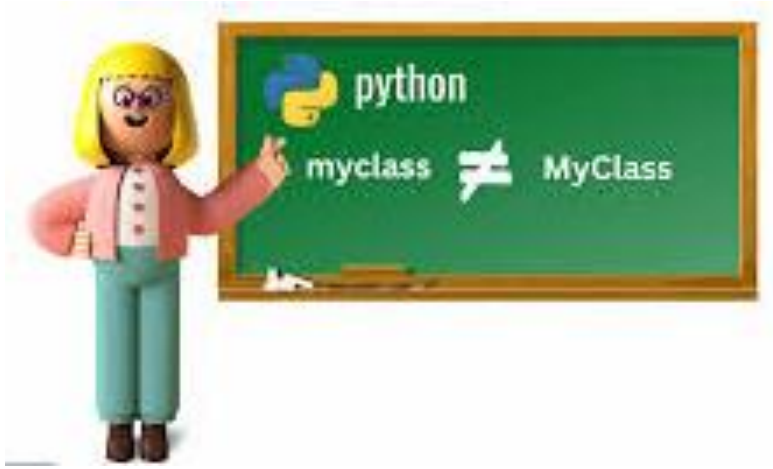
Example

```
age = 25
name = "Alice"
```

Trainer: Fawad Bahadur Marwat

Python Syntax & Structure

- ☑ Case-sensitive
- ☑ Indentation matters (no braces)
- ☑ Comments: # single-line, ''' multi-line '''
- ☑ Statement ends by newline (no semicolon needed)




```
def greet():
    print("Hello!")
    print("Welcome to Python.")
```

Data Types

What is Data type?

Building Blocks of Python Programs

1. **Numeric Types**
   - *Int, Float, Complex*
2. **Sequence Types**
   - *Str, list, tuple*
3. **Mapping Types**
   - *dict*
4. **Set Types**
   - *Set*
5. **Boolean Types**
   - *bool*
6. **None Types**
   - *None*

Integers in Python

What is a Integers?

- A whole number (positive, negative, or zero) without decimals.
- Used for counting, indexing, and mathematical operations.

Syntax

```
variable_name = integer_value
```

Example

```
age = 30
temperature = -10
count = 0
```

Key Properties

Immutable: Cannot be changed after creation (a new object is created on modification).
Unlimited Size: Python handles large numbers seamlessly.

Floating-Point Numbers (Floats) in Python

What is a Floats?

- Represents real numbers (with decimal points or scientific notation).
- Used for measurements, scientific calculations, and fractional values.

Syntax

```
variable_name = float_value
```

Example

```
pi = 3.14159
temperature = -12.5
scientific_notation =
2.5e3  # 2500.0
```

Key Properties

Precision: Limited by hardware (64-bit double-precision).
Immutable: Like integers, floats are immutable objects.

Strings in Python

What is a String?

- A sequence of characters enclosed in quotes (' ' or " ").
- Used to represent text data in Python.

Syntax

```
string_name = "text"  # or 'text'
```

Key Properties

Immutable: Cannot be changed after creation.
Indexed: Access characters using indices (starts at 0).
Iterable: Can loop through characters.

Example

```
name = "Alice"
greeting = 'Hello, World!'
multiline = """This is a multi-line string"""
```

Trainer: Fawad Bahadur Marwat

Boolean (bool) in Python

Definition

Represent True or False values.

Key Uses:

- Control flow in if and while statements.
- Result of logical operations (e.g., 5 > 3 → True).
- Represent states (e.g., on/off, open/closed).

Conversions:

bool(0) → False,

bool(1) → True.

Example

```
is_logged_in = True
if is_logged_in:
    print("User is logged in")
# Output: User is logged in
```

Working with boolean data type in python

Trainer: Fawad Bahadur Marwat

List in Python

Definition

Ordered, mutable collections of items.

Key Features

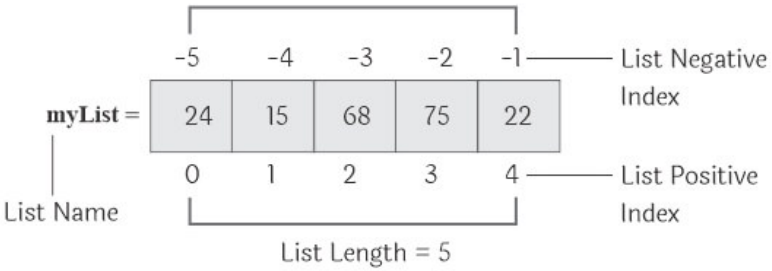
- Can store mixed data types (e.g., integers, strings, lists).
- Indexed from 0.
- Supports operations: .append(), .remove(), .sort().

Use Cases

Store sequences (e.g., names, numbers, objects).

Example

```
fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
print(fruits)
# Output: ['apple', 'banana', 'cherry', 'orange']
```



Trainer: Fawad Bahadur Marwat

Tuple in Python

Definition

Ordered, immutable collections of items.

Key Features

- Values cannot be changed after creation.
- Indexed from 0, like lists.
- Can store mixed data types (e.g., integers, strings, tuples).

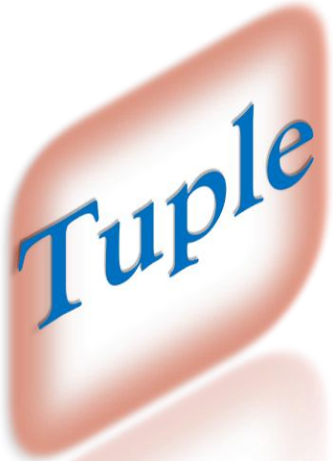
Use Cases

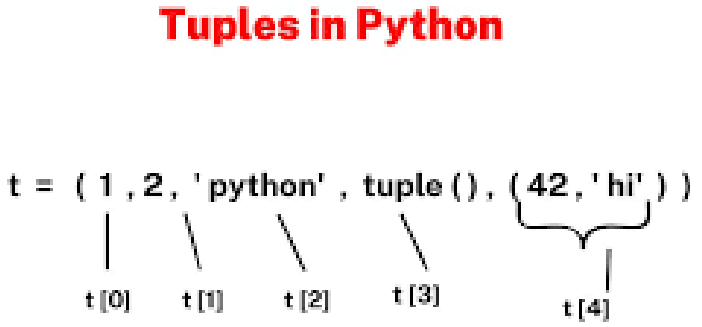
Fixed data sets (e.g., coordinates, color values).

Unpacking: x, y = (10, 20).

Example

```
coordinates = (34.0522, -118.2437)
print(coordinates[0])  # Output: 34.0522
```





Trainer: Fawad Bahadur Marwat

Dictionary in Python

Definition

Stores data in key-value pairs for fast lookups.

Keys: Unique, immutable (e.g., strings, numbers, tuples).

Values: Any data type, mutable.

Use Cases: Structured data (e.g., user profiles, settings, JSON).

Key Operations

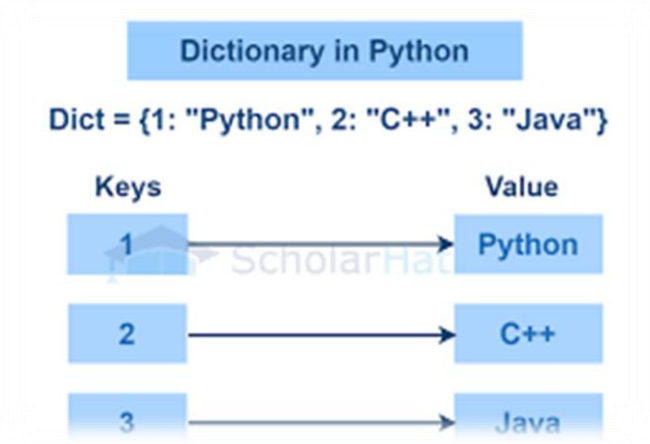
Add: dict['key'] = value

Remove: del dict['key']

Advantage: Fast key-based access via hashing.

Example

```
student = {"name": "Alice", "age": 22}
print(student["name"])  # Output: Alice
```



Set in Python

Definition

Stores unique, unordered elements..

Keys: Unique, immutable (e.g., strings, numbers, tuples).

Values: Elements must be immutable (e.g., strings, numbers, tuples).

Use Cases:

Membership testing, removing duplicates, mathematical set operations (union, intersection, difference).

Key Operations

Add: set.add(element)

Remove: set.remove(element)

Advantage: Fast membership testing via hashing; automatic duplicate removal.

Example

```
numbers = {1, 2, 3, 3, 4}
print(numbers)
# Output: {1, 2, 3, 4} (order may vary)
print(2 in numbers)
# Output: True
```

Trainer: Fawad Bahadur Marwat

Comparison of Mutability and Use Case

Data Type	Example	Mutable	Use Case
Int	x = 5	-	Whole numbers
Float	y = 3.24	-	Decimals
Str	s = ‘text’	❌	Text
Bool	Flag = True	-	True/False
List	[1,2,3]	☑	Modifiable collection
Tuple	(1,2)	❌	Fixed Collection
Dict	{“key” : “value”	☑	Key-Value Pairs
Set	{1,2,3}	☑	Unique elements