

Introduction to Python

- Python is a versatile, high-level programming language available on multiple platforms.
- Used in various fields, including web development, data analytics, and business forecasting.
- Syntax is intuitive and beginner-friendly, resembling English.
- Powerful and adaptable, appreciated by experienced programmers.
- Essential for developers due to its popularity in software development.

❓ Origin of Python

- Named after *Monty Python's Flying Circus*, a TV show, not the snake.
- The creator found the name short, cryptic, and appealing.

❓ Introduction by Layla Rizvi

- Background: Software engineer at Instagram, based in San Francisco.
- Over 10 years of experience coding in Python.
- First programming language she learned.
- Works at Meta, using Python daily, and considers it her favorite programming language.

❓ First Experience with Python

- Built a calculator as her first application.
- Faced challenges learning indentation, spacing, syntax, loops, and core computer science concepts.
- Perseverance led to success in mastering the language.

❓ Everyday Interaction with Python

- Applications that rely on Python: Instagram, Facebook, Google, and Spotify.
- Python is widely used in machine learning, including frameworks like TensorFlow.
 - Examples:
 - Airbnb uses TensorFlow to classify images.
 - Healthcare companies use it to classify MRI data.

❓ Python at Meta

- Used for Instagram's back end and ads machine learning algorithms.
- Supports production engineers in maintaining services.

❓ Why Learn Python?

- Easy to learn and beginner-friendly.
- Simple syntax and an abundance of libraries.
- Facilitates rapid development and feature building.
- Widely used and supported by a large engineering community.

❓ Encouragement

- Despite initial challenges, Python's simplicity makes it an excellent starting point.
- Offers quick results and a rewarding learning experience.
- A valuable skill for a software engineering career.

History of Programming

- **Charles Babbage (1822)**
 - Worked on improving calculating devices for navigation charts and astronomical tables.
 - Realized these devices were prone to human error and proposed the **Difference Engine**.
 - **Difference Engine:**
 - Used mechanical gears with numbers 0-9 etched onto them.
 - Operated manually using hand cranks to compute results.
- **Evolution of Babbage's Work**
 - Created improved designs, such as the **Difference Engine 2**.
 - Developed the concept of the **Analytical Engine**, considered the foundation of modern computing.
- **Ada Lovelace**
 - Documented how the Analytical Engine could perform a sequence of calculations, akin to a modern computer program.
- The **Analytical Engine** was never completed, partly due to lack of proper documentation.

Basics of How Computers Work

- Computers understand **binary code** (0 and 1):

- **0:** Represents "off" (low electrical state).
 - **1:** Represents "on" (high electrical state).
- **Binary Representation Examples:**
 - Decimal 1 → Binary 1.
 - Decimal 2 → Binary 10.
 - Decimal 3 → Binary 11.
- Binary code is represented using **transistors** housed inside the **CPU** (the computer's brain).

Introduction to Python

- Python is a high-level programming language that works on multiple platforms (Windows, Mac, Linux).
- Syntax is simple and similar to the English language, making it easy to learn and use.
- Created by **Guido van Rossum** and released in **1991**.

Key Features of Python

- Designed to be readable, with similarities to English and mathematics.
- Programs require less code compared to languages like C or Java.
- Abstracts complexity, allowing developers to focus on the core task.

Popularity and Use Cases

- Python is one of the most popular programming languages today.
- Widely used in areas such as:
 - **Web development.**
 - **Artificial intelligence** and **machine learning.**
 - **Data analytics.**
 - Various other programming applications.

Benefits of Learning Python

- Easy to learn, with a simpler learning curve than many other languages.
 - Adheres to the philosophy of "**write less, do more**".
 - Increases developer productivity, enabling faster project completion.
 - A great starting point for new programmers.
-

Career Opportunities

- High demand for Python developers in the job market.
- Learning Python can lead to a rewarding career in software development.

After Installation

My first python program

```
print("hello world")
```

Ways to Run Python Programs

1. Python Shell

- Useful for running and testing small code snippets.
- Allows direct code execution without creating .py files.
- To exit the Python shell, type `exit()` and press Enter.

2. Running Python Files from the Command Line/Terminal

- Files with the .py extension can be executed via command line/terminal.
- Syntax: `python filename.py`.

Python syntax

Spacing

Correct

#any ammount of whitespace on a single line is ok

```
x = 1 + 2
```

Incorrect

```
x = 1
```

```
+ 2
```

Indentation

Correct

```
def say_hello():  
    print("Hello there!")
```

```
print(say_hello())
```

Incorrect

```
def say_hello():  
print("Hello there!")
```

Commenting code

On completion of this reading, you will be able to explain why and where to use comments in coding.

Adding comments to code not only helps you as a developer but also helps other members of your team. Comments are great for refreshing your memory of code you may have written months ago and you may have forgotten what it was intended to do. There are multiple reasons why you need to add comments to a code file. They can range from the following:

- Explaining what the code is intended to do.
- Let developers know that code is deprecated.
- Add TODO comments for work to be completed at a later time.

Below are examples of the different types of comments that can be used.

Single-line comments

The use of a # symbol tells Python to ignore everything after this point until the end of the current line.

```
# Don't try to Run Me, I'm a comment
```

```
x = 10
```

```
# The function below accepts two integers and
```

```
# adds both numbers to return the value.
```

```
def add_two_numbers(a, b):
```

```
    return a + b
```

Inline comments

The # symbol tells Python to ignore everything after this point until the end of the current line, this allows the placement of comments within the code itself. Inline comments should be used to supply important information about the code they deal with.

```
x = 1 # Resetting buffer size
```