

Python Programming

1. **Python 3 (v3.8.1)**
2. **Visual Studio Code (ext. Python, Code Runner)**
3. **Anaconda Distribution**
4. **PyCharm (The Python IDE for Professional Developers)**

Overview

Python is an object-oriented programming language created by **Guido Rossum** in **February 1991**. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++.

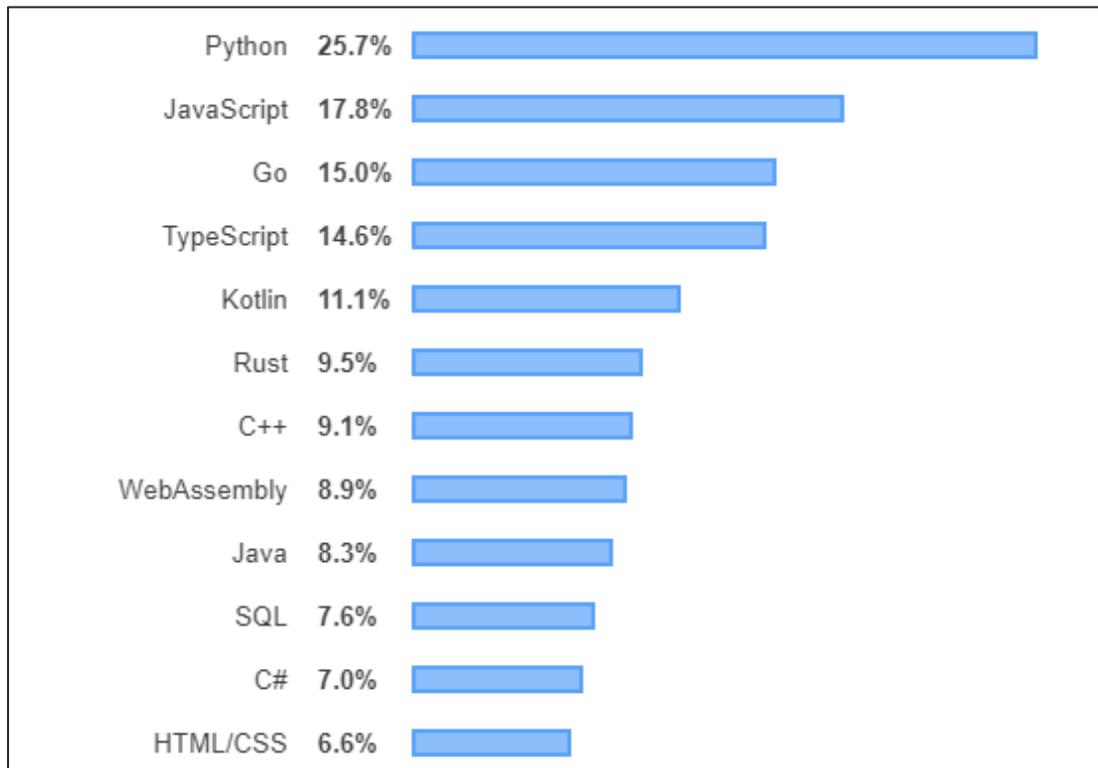
Many large companies use the Python programming language include **NASA, Google, YouTube, BitTorrent, etc.**

Python is a general-purpose language. It has wide range of applications from **Web Development** (like: Django and Bottle), **Scientific And Mathematical Computing** (Orange, SymPy, NumPy) to **Desktop Graphical User Interfaces** (Pygame, Panda3D).

The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

Python is the fastest growing programming language

Most Wanted Programming Languages



PYPL Popularity of Programming Language

Most Popular Programming Languages 1965 - 2019

Audience

This lecture has been prepared for the beginners to help them understand the basic to advanced concepts related to Python Programming language.

Class Prerequisites

Before you start doing practice with various types of examples given in this reference, I'm making an assumption that you are already aware about what is a **computer program** and what is a **computer programming language**.

Course Goals

Python Programming course will prepare you for jobs and careers connected with widely understood software development, which includes not only creating the code itself as a junior developer, but also computer systems design and software testing.

The main goal of the courses is to guide you from a state of complete programming illiteracy to a level of programming knowledge which allows you to design, write, debug, and run programs encoded in the Python language, and to understand the basic concepts of software development technology.

Materials

1. **Python Crash Course** by *Eric Matthes* (A Hands-On, Project-Based Introduction to Programming)
2. **Learn Python The Hard Way** by *Zed A. Shaw*
3. **Python for Data Analysis** by *Wes McKinney, O'Reilly*

*Students will receive books in digital format

Table Of Evaluation

| | |
|-----------------------------------|-------------|
| Attendance | 30% |
| Classroom Exercises and Practices | 70% |
| Total | 100% |

Certification

After completion of the full training package, upon successful completion, students shall receive the certificate if they have received a final **score of 75%** and delivered the training assignments/projects.

Training Method

Training includes lecture-based learning methods with live demonstrations.

Course Content

Introduction

- **History**
- **Features**
- **Why Choose Python?**

Python Setup

- **Python Installation**
- **Environment Selection**
- **Basic Syntax**

Object and Data Structure Basic

- **Numbers**
- **Strings**
- **Boolean**
- **Lists**
- **Dictionaries**
- **Tuples**
- **Files**
- **Sets**

Comparison Operators

- **Basic Operators**
- **Chained Comparison Operator**

Python Statements

- **If, elif, and else**
- **For loops**
- **While loops (range())**
- **List Comprehensions**

Method and Functions

- **Methods**
- **Functions**
- **Lambda expressions**
- **Nested Statements**
- **Scope**

Object Oriented Programming (OOP)

- **Objects**
- **Classes**
- **Methods**
- **Inheritance**
- **Special Methods**

Errors and Exception Handling

- **Errors**
- **Exceptions**
- **Try**
- **Except**
- **Finally**

Modules and Packages

- **Creating Modules**
- **Installing Modules**
- **Exploring the Python Ecosystem**

Buil-in Functions

- **Map**
- **Reduce**
- **Filter**
- **Zip**
- **Enumerate**
- **All and any**

Lab Exercises

Live Project