

A central graphic featuring a computer monitor with the Python logo on its screen. The logo consists of two interlocking snakes, one blue and one yellow. The background is dark blue with various abstract shapes and colors (yellow, red, green) and faint outlines of code editors and a smartphone.

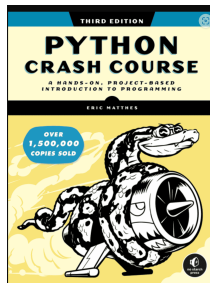
Introduction to **PYTHON**

Introduction to Python Programming

Kizito NKURIKIYEYEU, Ph.D.

Readings and activities

- Read Chap 1 —Getting started (page 3 through 13)
- Complete the installation of Python on your computer
 - Read the installation section in the textbook
 - Follow the instruction on Installing and Configuring Visual Studio Code for Python Development ^a
 - Watch the video on installing Visual Studio Code ^b
- Installing Jupyter notebook and jupyter lab ^c
- Read Chap 2 variable and simple data types
- Practice the exercises in the textbook (short in-text exercises)
- Complete the online quiz on the Google classroom



^a<https://realpython.com/python-development-visual-studio-code/>

^b<https://realpython.com/lessons/introduction-visual-studio-code/>

^c<https://jupyter.org/install>

What is Programming?

Definition

- The process of creating instructions for computers to follow
- A way to communicate with machines using specific languages

Key Components

- **Algorithms**—Step-by-step procedures for solving problems
- **Code**—Written instructions in a programming language
- **Syntax**—Rules for writing code correctly

Purpose

- Automate tasks
- Solve complex problems
- Create software applications
- Control hardware devices

Programming Languages

- High-level languages (e.g., Python, Java, C, C++, C#, Javascript, Rust, Kotlin, etc)
- Low-level languages (e.g., Assembly)
- Each with its own syntax and use cases

What is Programming?

Definition

- The process of creating instructions for computers to follow
- A way to communicate with machines using specific languages

Key Components

- **Algorithms**—Step-by-step procedures for solving problems
- **Code**—Written instructions in a programming language
- **Syntax**—Rules for writing code correctly

Purpose

- Automate tasks
- Solve complex problems
- Create software applications
- Control hardware devices

Programming Languages

- High-level languages (e.g., Python, Java, C, C++, C#, Javascript, Rust, Kotlin, etc)
- Low-level languages (e.g., Assembly)
- Each with its own syntax and use cases

What is Programming?

Definition

- The process of creating instructions for computers to follow
- A way to communicate with machines using specific languages

Key Components

- **Algorithms**—Step-by-step procedures for solving problems
- **Code**—Written instructions in a programming language
- **Syntax**—Rules for writing code correctly

Purpose

- Automate tasks
- Solve complex problems
- Create software applications
- Control hardware devices

Programming Languages

- High-level languages (e.g., Python, Java, C, C++, C#, Javascript, Rust, Kotlin, etc)
- Low-level languages (e.g., Assembly)
- Each with its own syntax and use cases

What is Programming?

Definition

- The process of creating instructions for computers to follow
- A way to communicate with machines using specific languages

Key Components

- **Algorithms**—Step-by-step procedures for solving problems
- **Code**—Written instructions in a programming language
- **Syntax**—Rules for writing code correctly

Purpose

- Automate tasks
- Solve complex problems
- Create software applications
- Control hardware devices

Programming Languages

- High-level languages (e.g., Python, Java, C, C++, C#, Javascript, Rust, Kotlin, etc)
- Low-level languages (e.g., Assembly)
- Each with its own syntax and use cases

What is Python?

- High-level programming language
- Created by Guido van Rossum
- First released in 1991
- Open-source and community-driven

Philosophy:

- Emphasizes code readability
- "There should be one– and preferably only one –obvious way to do it"
- "Simple is better than complex"
- Focus on programmer productivity

Key Features:

- Readability and clean syntax
- Extensive standard library
- Cross-platform compatibility
- Interpreted and dynamically typed
- Object-oriented and functional

Timeline:

- 1989: Development started
- 1991: Python 0.9.0 released
- 2000: Python 2.0 introduced
- 2008: Python 3.0 released
- 2020: Python 2 retired

Python Applications

Web Development:

- Frameworks: Django, Flask, FastAPI
- RESTful API development
- Web scraping and automation

Data Science & AI:

- Data analysis: Pandas, NumPy
- Machine Learning: Scikit-learn, TensorFlow
- Data visualization: Matplotlib

Scientific Computing:

- Scientific simulations
- Computational biology
- Physics and astronomy research

Other Areas:

- Game development (Pygame)
- Desktop applications (PyQt)
- System administration
- Education and teaching

Application of Python in Industry

Major Companies Using Python:

- Core component in web crawling at Google
- Library management, production engineering at Facebook
- Recommendation algorithms, security tools at Netflix
- Desktop client, backend services at Dropbox
- Data analysis, backend services at Spotify
- Backend web framework (Django) in Instagram
- Scientific computing tasks at NASA
- Production pipeline for films at Pixar

Python in Startups:

- Rapid prototyping capabilities
- Extensive libraries for various domains
- Cost-effective due to open-source nature
- Large talent pool of Python developers

Industries:

- Quantitative trading, risk management
- Medical imaging, genomics research
- Scripting in major game engines
- Penetration testing, threat modeling

Installing Python

Step 1: Downloading

- Go to python.org/downloads/
- Choose the latest stable version
- Select the appropriate installer for your OS

Step 2: Installation

- Run the installer
- Check "Add Python to PATH" (this is very important)

Note—If you get any issue in the installation:

- Carefully watch [this video](#)¹ on python installation
- Once done, Type: `python -version`
- This should display installed Python version

Step 3: Verification

- Open command prompt/terminal
- Type: `python -version`
- Should display installed Python version

Common Issues:

- PATH not set correctly
- Multiple Python versions
- Permission issues (Unix-based systems)

¹<https://www.youtube.com/watch?v=nU2Egc3Zx3Q>

Choosing a Python IDE

IDLE

- Comes bundled with Python
- Simple and lightweight
- Good for beginners
- Limited features

PyCharm^a

- Full-featured IDE
- Intelligent code completion
- Integrated debugger
- Available in free Community Edition

^a<https://www.jetbrains.com/pycharm/>

Visual Studio Code^a

- Lightweight but powerful
- Extensive plugin ecosystem
- Built-in Git integration
- Free and open-source

Other Options

- Jupyter Notebook^b: For data science
- Spyder^c: Scientific computing
- Thonny^d: Python IDE for beginners
- Google colab^e: A free cloud service to create interactive notebooks

^a<https://code.visualstudio.com/>

^b<https://jupyter.org/>

^c<https://www.spyder-ide.org/>

^d<https://thonny.org/>

Installing and Setting Up an IDE

We'll demonstrate with Visual Studio Code:

- 1 Download VS Code from code.visualstudio.com
- 2 Run the installer and follow the prompts
- 3 Open VS Code after installation
- 4 Install the Python extension:
 - Go to Extensions (Ctrl+Shift+X)
 - Search for "Python"
 - Install the official Microsoft Python extension
- 5 Create a new Python file: `hello.py`
- 6 Write a simple program: `print("Hello, World!")`
- 7 Run the program using the play button or terminal

NOTE: Follow the following online information (and video)

- [Introduction to Visual Studio Code](#)²
- [Python Development in Visual Studio Code](#)³

²<https://realpython.com/lessons/introduction-visual-studio-code/>

³<https://realpython.com/python-development-visual-studio-code/>

Next Steps

- Explore your chosen IDE's features
- Set up a virtual environment (we'll cover this later)
- Start writing and running simple Python programs
- Experiment with different IDEs to find your preference
- Don't hesitate to ask for help if you encounter issues
- Now we can start using python.
- Instructions will be provided using Jupiter notebook

Checklist of Tasks to Complete Before Next Lecture

- ☐ Ensure Python is installed on your computer
- ☐ Install Visual Studio Code on your computer
- ☐ Install Jupyter Notebook and JupyterLab on your computer
- ☐ Register for the course's Google Classroom
- ☐ Review the course lecture notes
- ☐ Complete Quiz #1 on the course's Google Classroom

Checklist of Tasks to Complete Before Next Lecture

- ☐ Ensure Python is installed on your computer
- ☐ Install Visual Studio Code on your computer
- ☐ Install Jupyter Notebook and JupyterLab on your computer
- ☐ Register for the course's Google Classroom
- ☐ Review the course lecture notes
- ☐ Complete Quiz #1 on the course's Google Classroom

Checklist of Tasks to Complete Before Next Lecture

- ☐ Ensure Python is installed on your computer
- ☐ Install Visual Studio Code on your computer
- ☐ Install Jupyter Notebook and JupyterLab on your computer
- ☐ Register for the course's Google Classroom
- ☐ Review the course lecture notes
- ☐ Complete Quiz #1 on the course's Google Classroom

Checklist of Tasks to Complete Before Next Lecture

- ☐ Ensure Python is installed on your computer
- ☐ Install Visual Studio Code on your computer
- ☐ Install Jupyter Notebook and JupyterLab on your computer
- ☐ Register for the course's [Google Classroom](#)
- ☐ Review the course [lecture notes](#)
- ☐ Complete Quiz #1 on the course's [Google Classroom](#)

Checklist of Tasks to Complete Before Next Lecture

- ☐ Ensure Python is installed on your computer
- ☐ Install Visual Studio Code on your computer
- ☐ Install Jupyter Notebook and JupyterLab on your computer
- ☐ Register for the course's [Google Classroom](#)
- ☐ Review the course [lecture notes](#)
- ☐ Complete Quiz #1 on the course's [Google Classroom](#)

Checklist of Tasks to Complete Before Next Lecture

- ☐ Ensure Python is installed on your computer
- ☐ Install Visual Studio Code on your computer
- ☐ Install Jupyter Notebook and JupyterLab on your computer
- ☐ Register for the course's [Google Classroom](#)
- ☐ Review the course [lecture notes](#)
- ☐ Complete Quiz #1 on the course's [Google Classroom](#)

The end