

CS ELECTIVE

Malaluan, Nervin C.

BSCS-3C

Introduction to Python

Python is a widely used, high-level, general-purpose programming language known for its simplicity, readability, and versatility. It's a great choice for beginners and seasoned programmers alike due to its:

Features:

Easy to learn: Python's syntax is clear and concise, resembling spoken English, making it easier to pick up compared to other languages.

Interpreted: Code is executed line by line, allowing for quick testing and debugging.

Dynamically typed: No need to explicitly declare variable types, simplifying development.

Object-oriented: Supports object-oriented programming principles for modular and reusable code.

Large standard library: Comes with a rich set of built-in modules for various tasks, reducing the need for external libraries.

Extensive third-party libraries: Thousands of libraries available for diverse applications, from web development to scientific computing.

Advantages:

Abundance of Third-Party Modules: Python benefits from a rich ecosystem of third-party modules and libraries that enhance its capabilities across diverse tasks.

Comprehensive Support Libraries: Python features extensive support libraries such as NumPy for numerical computations and Pandas for data analytics, making it well-suited for scientific and data-centric applications.

Open Source with a Large Community: Being open source, Python enjoys contributions from a sizable and active community, fostering continual development and robust community support.

Versatility, Readability, and Learnability: Python stands out for its versatility, simplicity, and readability, catering to both beginners and experienced programmers alike.

User-Friendly Data Structures: Python provides intuitive and user-friendly data structures, simplifying tasks related to data manipulation and management.

High-Level Abstraction: Python operates as a high-level language, abstracting away low-level complexities, resulting in a more user-friendly programming experience.

Dynamic Typing for Flexibility: Python's dynamic typing eliminates the need to explicitly declare data types, offering flexibility without compromising reliability.

Support for Object-Oriented and Procedural Programming: Python accommodates both object-oriented and procedural programming paradigms, offering versatility in coding styles.

Portability and Interactivity: Python is portable across operating systems and interactive, facilitating real-time code execution and testing.

Ideal for Prototyping: The concise syntax of Python enables rapid prototyping, allowing developers to create applications quickly with minimal code.

Highly Efficient: Python's clean design enhances process control, and its robust text processing capabilities contribute to its efficiency across various applications.

Internet of Things (IoT) Opportunities: Python is utilized in IoT applications due to its simplicity and adaptability to diverse scenarios.

Interpreted Language for Debugging Ease: As an interpreted language, Python supports easier debugging and facilitates smoother code development processes.

Disadvantages:

Interpreted Nature and Performance: Python being an interpreted language can result in slower execution compared to compiled languages like C or Java, impacting performance for tasks with high computational demands.

Global Interpreter Lock (GIL): Python's Global Interpreter Lock restricts multiple threads from executing Python code concurrently, limiting parallelism and concurrency in certain applications.

Memory Consumption: Python can exhibit high memory consumption, especially when dealing with extensive datasets or running resource-intensive algorithms.

Dynamic Typing Challenges: Python's dynamic typing allows variable types to change at runtime, potentially making error detection more challenging and introducing bugs.

Packaging and Versioning Complexity: The abundance of Python packages and libraries may lead to complexities in versioning and potential conflicts among packages.

Lack of Strictness: Python's flexibility can be advantageous for rapid development but may result in code that is harder to comprehend and maintain due to a lack of strictness.

Steep Learning Curve for Beginners: Despite being considered relatively easy to learn, Python may pose a steep learning curve for beginners, particularly those without prior programming experience.

Applications:

Web development: Back-end and front-end frameworks like Django and Flask power popular websites and applications.

Data science and analytics: Libraries like NumPy, Pandas, and SciPy make Python a go-to choice for data analysis and machine learning.

System scripting and automation: Python scripts automate tasks, manage system resources, and interact with APIs.

Game development: Frameworks like Pygame enable game development, from simple 2D games to complex 3D projects.

Education and research: Used in educational tools and research projects due to its accessibility and ease of use.

Graphical User Interface (GUI) Applications Development: Python is utilized to create applications with a graphical interface.

Graphics Design, Image Processing, Games, and Scientific/Computational Applications: Python is employed in tasks such as graphic design, image processing, game development, and scientific or computational applications.

Web Frameworks and Applications: Python serves as the foundation for popular web frameworks like Django and Flask, which are used to build web applications.

Enterprise and Business Applications: Python finds application in various business tasks, including data analysis and automation within enterprise applications.

Operating Systems Development: Python is utilized in the development of operating systems and tools that manage system functions.

Education: Python is commonly used as a teaching language in programming and computer science courses.

Database Access: Python provides libraries that facilitate accessing and managing databases.

Language Development: Python is employed to create and evolve new programming languages.

Prototyping: Python is well-suited for quickly creating prototypes of software and applications.

Software Development: Python is widely used for general-purpose software development.

Data Science and Machine Learning: Python is a primary language for tasks related to data science and machine learning.

Scripting: Python is widely utilized for writing scripts that automate various tasks and processes.

What is Python? A high-level, general-purpose, interpreted programming language.

Why Python? Easy to learn, versatile, with extensive libraries and a large community.

How to install Python? Visit <https://www.python.org/downloads/> and follow the instructions for your operating system.

How to install PyCharm (IDE)?

Visit the Official Website:

Go to the PyCharm website at <https://www.jetbrains.com/pycharm/>.

Download PyCharm:

On the PyCharm website, locate the "Download" button.

Click on the "Download" button to start the download process.

Choose Edition:

PyCharm is available in two editions: Community (free) and Professional (paid).

Choose the edition that best suits your needs and click on the respective download link.

Download and Save the Installer:

Once you've selected the edition, the website will provide you with an installer file (usually a .exe file for Windows or a .dmg file for macOS).

Save the installer file to your computer.

Run the Installer:

Locate the downloaded installer file and run it.

On Windows, this might involve double-clicking the .exe file.

On macOS, double-click the .dmg file and drag the PyCharm icon to the Applications folder.

Follow the Installation Wizard:

The installation wizard will guide you through the installation process.

Follow the on-screen instructions, which typically involve accepting the license agreement, choosing the installation location, and configuring optional settings.

Launch PyCharm:

Once the installation is complete, you can launch PyCharm.

On Windows, you might find a shortcut on your desktop or in the Start menu.

On macOS, you can open PyCharm from the Applications folder.

Configure PyCharm (Optional):

Upon launching PyCharm for the first time, you may be prompted to configure settings such as your theme, keymap, and other preferences.

Activate or Evaluate:

If you are using the Professional edition, you may need to activate it with your JetBrains account or evaluate it for a trial period.

Start Coding:

Once the installation and configuration are complete, you can start using PyCharm for your Python development projects.

Basic Python Program Structure:

```
Python
print("Hello, World!")

# Define a function
def add(a, b):
    return a + b

# Call the function
result = add(5, 3)
print(f"5 + 3 = {result}")
```

Basic Python Program Explanation:

- The first line prints "Hello, World!" to the console.
- The add function takes two arguments (a and b) and returns their sum.
- The result variable stores the return value of calling add with 5 and 3.
- The last line prints a formatted string with the addition result.