



Shell | Bash
Scripting

Python

Introduction to Python

- Python is a popular high-level programming language known for its simplicity and readability.
- It was created by Guido van Rossum and first released in 1991.
- Python emphasizes code readability and a clear syntax, making it an excellent choice for beginners and experienced programmers alike.
- It supports multiple programming paradigms, including procedural, object-oriented, and functional programming styles.
- Python has a vast standard library and a thriving ecosystem of third-party packages, making it versatile for a wide range of applications, such as web development, data analysis, artificial intelligence, and automation.

Python Installation

- Installation on **Windows** : <https://www.digitalocean.com/community/tutorials/install-python-windows-10>
- Installation on **Ubuntu/Debian** : <https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-programming-environment-on-an-ubuntu-18-04-server>
- Installation on **Fedora/CentOS/RHEL** : <https://www.digitalocean.com/community/tutorials/how-to-install-python-3-and-set-up-a-local-programming-environment-on-centos-7>
- Installation on **Mac OS** : <https://www.dataquest.io/blog/installing-python-on-mac/>
- Installation using **Shell/Bash Script** : <https://blog.knoldus.com/how-to-install-python-in-target-host-using-ansible/>
- Installation using **Ansible** : <https://relativkreativ.at/articles/how-to-install-python-with-ansible>

Check Python version using command : **python --version**



Python

Use Cases of Python

- Python has a wide range of use cases due to its versatility and extensive libraries. Here are some popular applications of Python:
- 1) **Web Development:** Python frameworks like Django and Flask enable rapid development of web applications, providing tools for database management, routing, and user authentication.
 - 2) **Data Analysis and Visualization:** Python libraries like NumPy, Pandas, and Matplotlib offer powerful tools for data manipulation, analysis, and visualization, making it ideal for tasks such as data cleaning, exploration, and generating visual representations.
 - 3) **Machine Learning and Artificial Intelligence:** Python's libraries, such as TensorFlow, Keras, and scikit-learn, provide comprehensive support for machine learning and AI development. Python's simplicity and readability make it easier to implement complex algorithms and train models.
 - 4) **Scripting and Automation:** Python's scripting capabilities make it an excellent choice for automating repetitive tasks, system administration, and scripting in various applications.
 - 5) **Scientific Computing:** Python, along with libraries like SciPy and SymPy, is extensively used in scientific computing, numerical simulations, and solving complex mathematical equations.
 - 6) **Game Development:** Python's libraries, such as Pygame and Panda3D, offer frameworks for game development, making it easier to create interactive games and simulations.
 - 7) **Internet of Things (IoT):** Python is used for programming IoT devices, enabling communication, data processing, and control of connected devices.
 - 8) **Prototyping:** Python's quick development cycle and easy integration with other languages make it an excellent choice for prototyping and proof of concept projects.
 - 9) **Web Scraping:** Python libraries like BeautifulSoup and Scrapy provide tools for extracting data from websites, enabling tasks such as web scraping, data mining, and automation.
 - 10) **Desktop Applications:** Python can be used to develop cross-platform desktop applications using frameworks such as PyQt and Tkinter.



Shell | Bash

Introduction to Shell | Bash

- **Shell** is a program that provides an interface for users to interact with an operating system. It allows users to execute commands and run programs by typing text-based instructions.
- **Bash** (Bourne Again Shell) is one of the most commonly used Unix shells, and it is the default shell on many Linux distributions.
- Bash provides a command-line environment where users can type commands and receive output. It supports various features, including command execution, input/output redirection, piping, scripting, and automation. Users can navigate the file system, manage files and directories, manipulate text, and perform system administration tasks using Bash commands.
- In addition to executing commands interactively, Bash allows users to write scripts—a series of commands saved in a file—which can be executed as a program. Bash scripts are useful for automating repetitive tasks, performing complex operations, and creating custom command-line tools.
- Shell scripting with Bash involves writing scripts using a combination of built-in shell commands, external programs, control structures, variables, and various programming constructs.
- Bash scripts can take inputs, perform calculations, make decisions based on conditions, iterate over lists of data, and perform other operations to achieve specific goals.
- **Shell is not a standalone program that needs to be installed.** It is a fundamental component of an operating system.
- On Unix-based systems, such as Linux and macOS, the shell is an integral part of the system, and there are various shells available, including **Bash**, **Zsh**, **Csh**, **Ksh**, and more. These shells are usually pre-installed, and you can access them through the terminal or command line interface.



Shell | Bash

Bash Installation

- Bash (Bourne Again Shell) is **typically pre-installed on most Unix-based systems, including Linux and macOS**. However, if you need to install Bash on a system where it is not already available or if you want to ensure you have the latest version
- **Installation on Ubuntu/Debian** : <https://zoomadmin.com/HowToInstall/UbuntuPackage/bash>
- **Installation on Fedora/CentOS/RHEL** : **sudo yum install bash**
 - It is good to update the package repository using - `sudo yum update`
- **Installation on Mac OS** using **brew** : **brew install bash**
 - If **brew** is not installation then install it using - `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`

- **Installation on Windows :**

Bash is not natively available on Windows. However, you can install a Bash environment using Windows Subsystem for Linux (WSL) or Git Bash.

- 1) WSL: If you are using Windows 10, you can enable the Windows Subsystem for Linux feature and install a Linux distribution from the Microsoft Store (e.g., Ubuntu, Debian, etc.). These distributions come with Bash pre-installed.
- 2) Git Bash: Another option is to install Git for Windows, which includes a Bash emulation environment called Git Bash. You can download the Git for Windows installer from the official website (<https://gitforwindows.org/>) and follow the installation steps.
- 3) Else you can also check out this -<https://itsfoss.com/install-bash-on-windows/>

Check Bash version using command : **bash --version**



Shell | Bash

Use Cases of Bash | Shell

- Bash and shell have numerous use cases that range from interactive command-line usage to complex automation and scripting tasks. Here are some common use cases for Bash and shell:
- 1) **Interactive Command-Line Usage:** Bash and other shells provide an interactive command-line interface where users can execute commands and interact with the operating system. This allows users to navigate the file system, manage files and directories, perform system administration tasks, and execute various utility programs.
 - 2) **Shell Scripting:** One of the primary use cases for Bash and shell is scripting. Shell scripts are a series of commands and instructions written in a shell language (such as Bash) that can be executed as a program. Shell scripting is useful for automating repetitive tasks, performing system administration tasks, creating custom command-line tools, and executing complex workflows.
 - 3) **System Administration:** Shell scripting is widely used for system administration tasks. System administrators leverage shell scripts to automate routine maintenance tasks, configure and manage servers, perform backups, monitor system resources, and manage user accounts.
 - 4) **File Manipulation and Text Processing:** Bash and shell provide powerful tools for manipulating files and processing text. Shell commands can be used to search, filter, and manipulate text files, extract specific information, perform text transformations, and generate reports. Utilities like grep, sed, awk, and cut are commonly used for these purposes.
 - 5) **Automation and Task Scheduling:** Bash and shell scripting allow for the automation of various tasks. This includes scheduling and running tasks at specific times using tools like cron on Unix-based systems. Automation with shell scripts enables batch processing, data processing pipelines, and the execution of multiple tasks in sequence.
 - 6) **Software Development:** Bash and shell scripting can be utilized in software development workflows. Shell scripts can be used to build, package, and deploy software, run tests, manage version control systems, and facilitate continuous integration and deployment processes.
 - 7) **Data Processing and Analysis:** Bash and shell provide capabilities for data processing and analysis tasks. Shell scripts can be used to process large datasets, extract relevant information, transform data formats, and generate reports. By combining shell commands with other tools like grep, sed, awk, and sort, data processing tasks can be automated and customized.





If you like this post and find this post helpful, then please share.



Like



Comment



Save



<https://www.linkedin.com/in/themr255/>

Reach me at : maheshrm255@gmail.com

GitHub : <https://github.com/themr255>

TWS Community Builder : <https://www.linkedin.com/company/trainwithshubham/>

TWS YouTube Channel: <https://www.youtube.com/@TrainWithShubham>