Git, Bash, and Programming Basics

Git Basics

Git is a distributed version control system commonly used for tracking changes in code.

1. Initialize a Git repository: `git init`
2. Clone a repository: `git clone <repository_url>`
3. Check repository status: `git status`
4. Stage changes: `git add <file>`
5. Commit changes: `git commit -m "Commit message"`
6. Push to a remote repository: `git push`
7. Pull changes from a remote: `git pull`
8. Create a new branch: `git branch <branch_name>`
9. Switch to a branch: `git checkout <branch_name>`
10. Merge branches: `git merge <branch_name>`
11. View commit history: `git log`
12. Discard local changes: `git reset --hard`
13. Create a Git tag: `git tag <tag_name>`
14. Remove a Git tag: `git tag -d <tag_name>`

Bash/Command Line Basics

Bash (Bourne Again Shell) is a command-line shell and scripting language.

1. Navigate to a directory: `cd <directory_path>`
2. List files and directories: `ls`
3. Create a new directory: `mkdir <directory_name>`
4. Remove a file: `rm <file_name>`
5. Copy a file: `cp <source_file> <destination_directory>`
6. Move/Rename a file: `mv <old_name> <new_name>`
7. Display file content: `cat <file>`
8. Execute a script: `./<script_name>`
9. View manual page for a command: `man <command>`
10. Find files by name: `find <directory> -name <filename>`
11. Search for text in files: `grep <pattern> <file>`
12. Redirect output to a file: `command > output.txt`

Common Programming Words and Symbols

- **Variable**: A symbolic name for a value.
- **Function**: A reusable block of code.
- **Conditional Statement**: Executes code based on a condition (e.g., `if`, `else`).
- **Loop**: Repeats a block of code until a condition is met (e.g., `for`, `while`).
- **Comment**: An annotation in code for documentation.
- **Array**: A collection of elements, indexed by numbers.
- **String**: A sequence of characters.
- **Integer**: A whole number (e.g., 123)
- **Float**: A number with a decimal point (e.g., 3.14)
- **Boolean**: Represents true or false.
- **Operator**: Performs operations (e.g., +, -, *, /).
- **Function Call**: Invoke a function to execute its code.
- **Variable Assignment**: Set a variable's value.

Glossary

Argument/Parameter: Information passed to a function or command to influence its behavior.

Branch: A separate line of development within a Git repository, often used for isolating features or bug fixes.

Clone: Creating a local copy of a remote repository on your computer.

Command: An instruction to the shell.

Commit: A snapshot of changes made to the code, accompanied by a message describing the changes.

Command Line Interface (CLI): A text-based interface for interacting with a computer's operating system and software.

Directory: A folder that can contain files or other directories.

Environment Variable: A variable that holds system or user-specific information used by applications and the operating system.

Executable: A file that can be run as a program or script.

File: A named collection of data or information.

Fork: Creating a personal copy of a repository, allowing independent development without affecting the original.

Function: A reusable block of code that performs a specific task.

Function Call: Invoking a function to execute its code, providing necessary arguments.

Git: A distributed version control system used for tracking changes in code.

IDE (Integrated Development Environment): A software suite that combines code editing, debugging, and build automation tools for software development.

Integer: A whole number, without a decimal point.

Loop: A programming construct that repeatedly executes a block of code until a specified condition is met.

Merge: Combining changes from one branch into another.

Merge Conflict: A situation where Git cannot automatically resolve differences between code changes made by multiple contributors.

Operator: A symbol or keyword that performs an operation on one or more values or variables in code.

Path: The address or location of a file or directory in the file system.

Git, Bash, and Programming Basics

Permission: Access rights that determine who can read, write, or execute files and directories.

Pipe: A method for redirecting the output of one command as input to another command.

Programming Languages: Languages used to write software, each with its own syntax and purpose.

Pull: Fetching and incorporating code changes from a remote repository into the local repository.

Pull Request (PR): A request to merge changes from one branch or fork of a repository into another, often used for code review.

Push: Sending local code changes to a remote repository.

Repository: A storage location for a project's files and version history.

Repository Hosting: Online platforms like GitHub or GitLab where Git repositories can be hosted and shared.

Remote: A copy of a Git repository hosted on a server or another location.

Script: A series of commands or instructions saved in a file for automated execution.

Standard Input/Output (stdin/stdout): Channels for input (keyboard) and output (screen) of commands in the command line.

Syntax: The set of rules that dictate how code must be structured in a programming language.

Text Editor: A software application for creating and editing text files, often used for code editing.

User Directory (Home Directory): The top-level directory associated with a user's account, often represented by ' '.

Variable: A symbolic name for a value or data storage location.

Version Control: The practice of tracking and managing changes to code using tools like Git.

Your Name
YourGitHubLink