

Development Environment

Simple Bash Commands

- **cd (change directory):** This command allows you to navigate through the different directories in the Linux file system. You can use it to move into a specific folder, go back to the previous folder, or jump to your home directory.
 - Example: `cd development/java/Assignment1` (move into the Assignment1 directory)
 - Example: `cd ..` (go back to the parent directory)
 - Example: `cd ~` (go to your home directory)
- **mkdir (make directory):** This command is used to create new directories (folders) in your current location.
 - Example: `mkdir new_folder` (create a new folder named "new_folder")
- **rm (remove):** This command is used to delete files and directories. Be careful when using this command, as deleted files are usually not recoverable from the command line. You can use the `-r` flag to recursively delete directories and their contents.
 - Example: `rm important_file.txt` (delete a file named "important_file.txt")
 - Example: `rm -r old_folder` (delete a folder named "old_folder" and everything inside it)
- **ll (list files and directories):** This command provides a detailed listing of the files and directories in your current location. It shows information like permissions, ownership, size, and modification date.
 - Example: `ll` (lists files and directories in the current location)
- **.(current directory):** This is a special shorthand to refer to the current directory you are in.
 - Example: `code .` (opens VS Code in the current directory)
- **.. (parent directory):** This is a special shorthand to refer to the directory one level up from your current location (the parent directory).
 - Example: `cd ../..` (go up two levels in the directory structure)
- **~ (home directory):** This is a shorthand to refer to your personal home directory in the Linux file system.
 - Example: `cd ~` (navigates you to your home directory)
 - Example: `ll ~/Documents` (lists the contents of your Documents folder in your home directory)
- *** (wildcard):** This special character is used to represent one or more other characters. It's often used to select multiple files or directories at once.
 - Example: `ll *.java` (lists all files ending with ".java" in the current directory)
 - Example: `rm data*` (removes all files or directories starting with "data")

Simple Git Commands

- **git status:** This command shows you the current status of your Git repository, including any changes that have been made but not yet committed.
 - Example: `git status`
- **git add:** This command stages changes, preparing them to be included in the next commit.
 - Example: `git add *` (stages all the changes in your current directory)
 - Example: `git add HelloWorld.java` (stages a specific file)

- **git commit -m "{message}"**: This command saves your staged changes along with a descriptive message explaining what you've done. Replace `{message}` with your actual commit message.
 - Example: `git commit -m "Added initial code for HelloWorld"`
- **git push**: This command uploads your local commits to a remote repository (like on GitHub), allowing others to see your changes.
 - Example: `git push` (pushes to the repository)

Linux Directory System (Simplified View of /home)

Here's a simplified view of how your personal files and folders might be organized within the `/home` directory for development:

```
/home/  
├── development/  
│   ├── java/  
│   │   ├── Assignment1/  
│   │   │   └── HelloWorld.java  
│   │   ├── Assignment2/  
│   │   └── Assignment3/  
│   └── python/  
├── data/  
└── notes/
```

In this structure:

- `/home/` is the main directory.
 - `development/` is a subdirectory within `/home/`.
 - `java/` is a subdirectory within `development/`.
 - `Assignment1/` is a subdirectory within `java/`, and contains the file `HelloWorld.java`.
 - `Assignment2/` is a subdirectory within `java/`.
 - `Assignment3/` is a subdirectory within `java/`.
 - `python/` is another subdirectory within `development/`.
 - `data/` is a subdirectory within `/home/`.
 - `notes/` is also a subdirectory within `/home/`.

This structure helps keep your different types of files organized.

Getting Started with Your Development Environment

Here's how to start working in your development environment:

1. **Open up the terminal.** Press the terminal button (this might be an icon on your taskbar, or you might need to search for "Terminal" or "Ubuntu" in your start menu). This terminal is running a special environment called WSL (Windows Subsystem for Linux), which allows you to run a Linux operating system (in this case, Ubuntu) directly on your Windows computer. This helps keep your coding environment separate from your regular computer activities.



2. **Navigate to your assignment folder.** Use the `cd` command to move into the correct directory. For example, to get to your first Java assignment, you would type: `cd development/java/Assignment1`.
 - **Tip for Autocompletion:** When typing directory or file names in the terminal, you can often press the `Tab` key. If the name you've typed is unique enough, the terminal will automatically complete the rest of the name for you. This can save you a lot of typing!
3. **Open Visual Studio Code.** Once you are in the correct assignment directory, type `code .` and press Enter. The `code` command will open the Visual Studio Code editor. The `.` (period) at the end of the command tells VS Code to open the editor with the current directory as the workspace.

Compiling and Running Java

Here's how you can compile and run your Java code:

1. **Navigate to your assignment directory in the terminal.** For example: `cd /home/development/java/Assignment1`.
2. **Compile your Java code.** Use the `javac` command followed by the name of your Java file (including the `.java` extension). For example, if your file is named `HelloWorld.java`, you would type: `javac HelloWorld.java`.
3. **Run your compiled Java code.** Once the code is compiled successfully, you can run it using the `java` command followed by the name of your Java file (without the `.java` extension). For example: `java HelloWorld`.