Chapter 1: Setting Up Your Environment

For Windows Users Only:

Set up a Linux-like environment using Windows Subsystem for Linux (WSL). Refer to the official documentation or instructional YouTube videos for guidance:

1. How to Install the Latest Ubuntu on Windows 11 (WSL)
2. How to Install Ubuntu on Windows 11 (WSL)

Official guide:
https://canonical-ubuntu-wsl.readthedocs-hosted.com/en/latest/guides/install-ubuntu-wsl2/

Note:

If you are using Linux or macOS, you do not need to set up WSL. Proceed directly with your native terminal.

GitHub Getting Started: Essential Commands and SSH Setup

Git is a powerful and widely-used version control system, essential for collaborative software development. Below is an improved and comprehensive guide for getting started with GitHub, including SSH key setup and important Git commands.

1. Step 1: Set Up SSH Keys for GitHub
    a. **Checking for existing SSH keys**
    b. **Generateing a new SSH key**
    c. **Adding a new SSH key to your GitHub account**
    d. **Testing your SSH connection - GitHub Docs**

2. Step 2: Clone and Work with a Repository
    a. Fork the Repository:
    Go to https://github.com/IBM-Cloud/get-started-python and click "Fork" to create your own copy.
    b. You can see the same repository in your profile's repository.
        i. https://github.com/shubhambagwari/get-started-python
    c. Clone Your Forked Repo:
        i. *git clone* https://github.com/shubhambagwari/get-started-python
        ii. Note: Replace **shubhambagwari** with **your_github_id**
3. Step 3: GitHub (Make Changes and Push)
    a. Change any file
    b. *git add <that changed file>* →Add a file to the staging area
    c. *git commit -m "some meaningful message"* → Commit changes
    d. *git push* → Push changes to remote repository (remembered branch)
    e. *git init* → Initialize a local Git repository
    f. *git status* → check status
    g. git branch → See all the branches
    h. *git branch <new_branch_name>* → To create new branch
    i. *git checkout <branch_name>* → To activate any branch
    j. *git push origin <branch_name>* → Push a branch to your remote repository
    k. *git push -u origin <branch name>* → Push changes to remote repository (and remember the branch)
    l. *git log* → View changes
    m. *git log --summary* → View changes (detailed)
    n. *git log --oneline* → View changes (in short)
    o. *git clone ssh://git@github.com/[username]/[repository-name].git* →Create a local copy of a remote repository
    p. *git merge <branch>* → Merge another branch into the current branch
    q. *git pull*→ Fetch and merge changes from the remote repository
    r. *git remote -v* → List remote repositories

Note: Use ***git push --force*** or ***git push -f*** only when you need to overwrite remote history, such as after a rebase. Use it with caution to avoid data loss. After pushing, check your GitHub repository online to confirm your commit appears as expected.

Chapter 2: Conda Instructions

Conda is a powerful package and environment management system that allows you to create isolated environments, each with its own files, packages, and dependencies. This isolation helps prevent conflicts between projects and ensures reproducibility

Why Use Conda Environments?

1. Each environment is independent, so packages and dependencies do not interfere with each other.
2. You can easily manage different Python versions and package sets for different projects.

Installing Miniconda: Miniconda is a lightweight version of Anaconda, providing only the essential packages and the Conda package manager. It is recommended for users who want a minimal setup and full control over installed packages. Installation Steps (Linux Example)

a. Update your system (optional but recommended)
   i. sudo apt update -y
   ii. sudo apt upgrade -y
b. mkdir -p ~/miniconda3
c. wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda3/miniconda.sh
d. bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
e. source miniconda3/bin/activate → To activate miniconda
f. conda create --name <env_name> → create conda environment:
g. conda create -n <env_name> python=3.5 →install specific python version
h. conda activate <env_name> → To activate environment
i. conda install <package_name> → Once the environment is activated, you can install any necessary packages using "**conda install** " or "**pip install**".
j. pip install -r requirement.txt → install dependencies [This file is already present in the repo that you forked.]
k. python hello.py → Run any file
l. conda deactivate → When you are done working in the environment, you can deactivate it using
m. conda env list → To see all your conda environments

Note:

1. https://docs.conda.io/en/latest/miniconda.html