Ollscoil na Gaillimhe
University of Galway

Introduction to CLI & Git

Oksana Dereza
Digital Library Developer
oksana.dereza@universityofgalway.ie

University *of* Galway.ie

Ollscoil na Gaillimhe
University of Galway

Command Line Interface

University *of* Galway.ie

GUI vs CLI

GUI = Graphical User Interface

CLI = Command Line Interface

Shell — a program providing direct access to the OS (most often through a CLI)

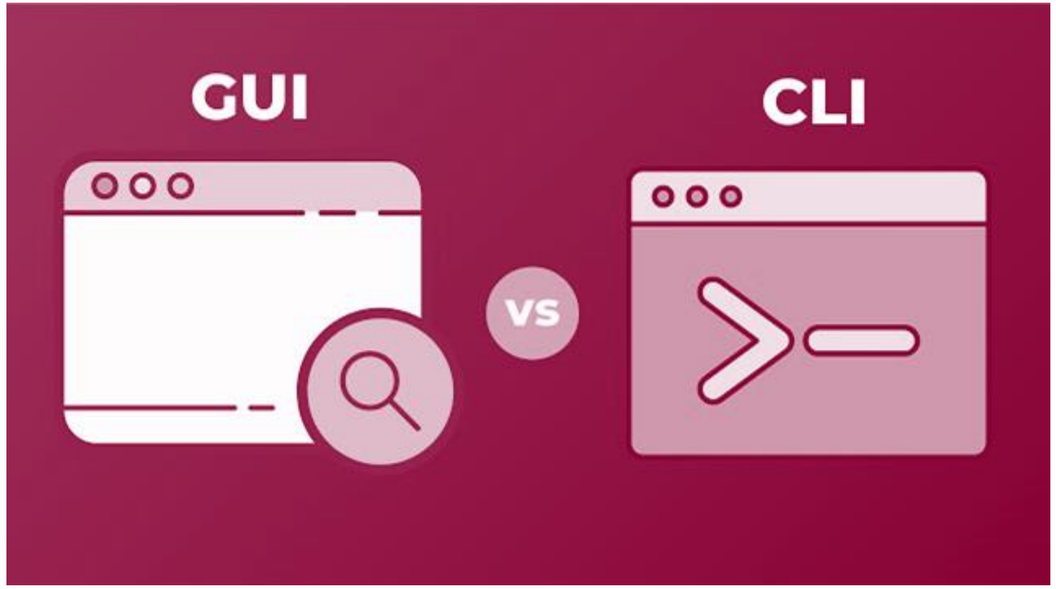


Image source: YouTube

CLI	GUI
Users interact with the system using various commands in the command prompt window.	Users interact with the system with graphical elements such as icons, menus, images, etc.
CLI requires only a keyboard to enter commands.	GUI requires various input devices to interact with the system, such as a keyboard, mouse, etc.
CLI is not very user-friendly because the user needs to memorize a lot of commands.	GUI is relatively simpler to use and is more user-friendly.
CLI is relatively faster and excels in efficiency for professional users.	GUI is slower and tends to prioritize ease of use over speed.
CLI consumes less RAM and processing power.	GUI consumes more RAM and processing power.

DB Difference Between.net

What can you do with it?

Ollscoil na Gaillimhe
University of Galway

- Batch process files (rename, delete, move)
- Quick search and filter file names and file contents
- Run programs with CLI (e.g. Cantaloupe Image Server, Tesseract OCR, kraken)
- Run scripts and pipelines on your machine (Python, R etc.)
- Run jobs on remote servers and HPC facilities
- Transfer data to/from remote servers
- Automate workflows with shell scripts
- Install and manage software
- Version control and collaboration (e.g. Git, Mercurial)

Curated list of CLI tools:
https://github.com/toolleeo/awesome-cli-apps-in-a-csv

Unix vs Windows

Ollscoil na Gaillimhe
University of Galway

Unix & Unix-like

- **MacOS, Linux, BSD**
- Share common features, standards and design principles, including the CLI
- Built-in CLI is called **Terminal** and uses shells like **Bash** or **Zsh**
- Support both basic commands and advanced scripting
- Most commands for MacOS and Linux users will be the same.

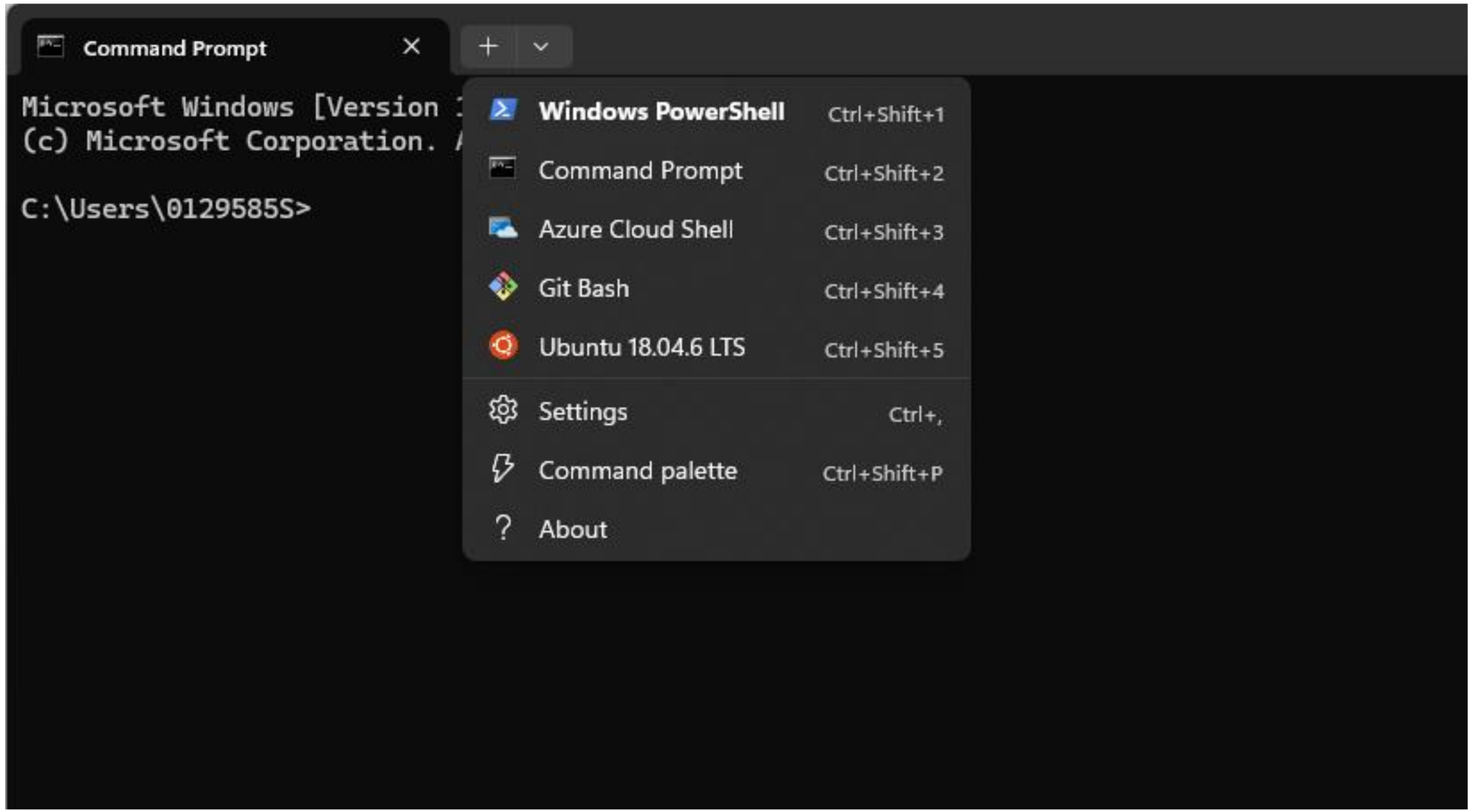
Windows

- Windows has two different built-in CLIs/shells, **Command Prompt** (cmd.exe) and **PowerShell**
- They have a different syntax and focus
- cmd is your go-to CLI for everyday tasks
- PowerShell is more object-oriented and provides advanced scripting capabilities

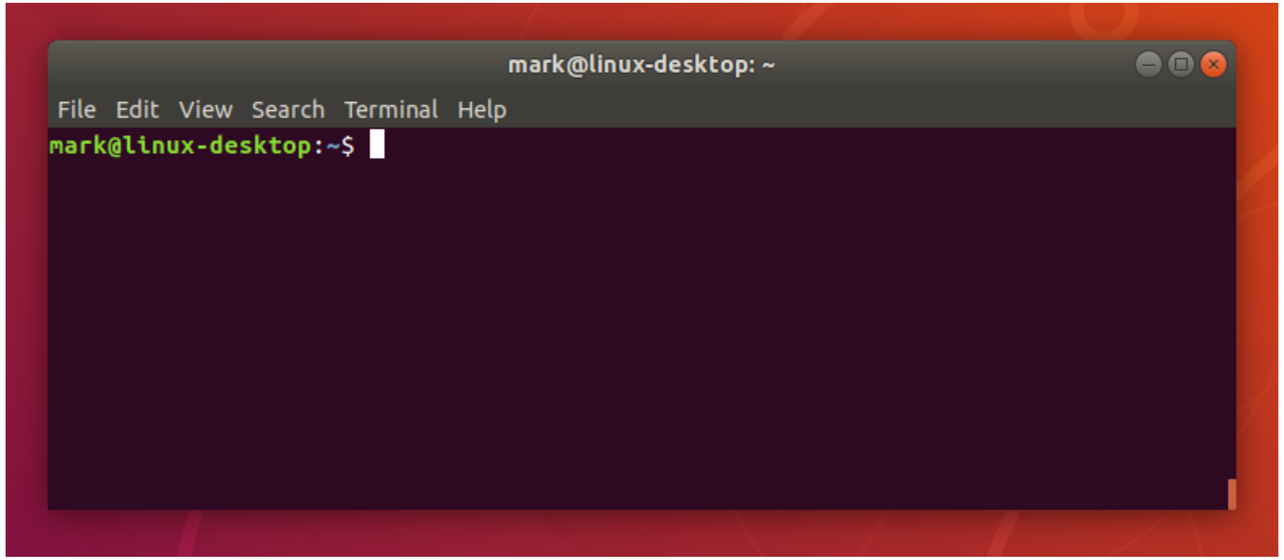
Windows Command Prompt

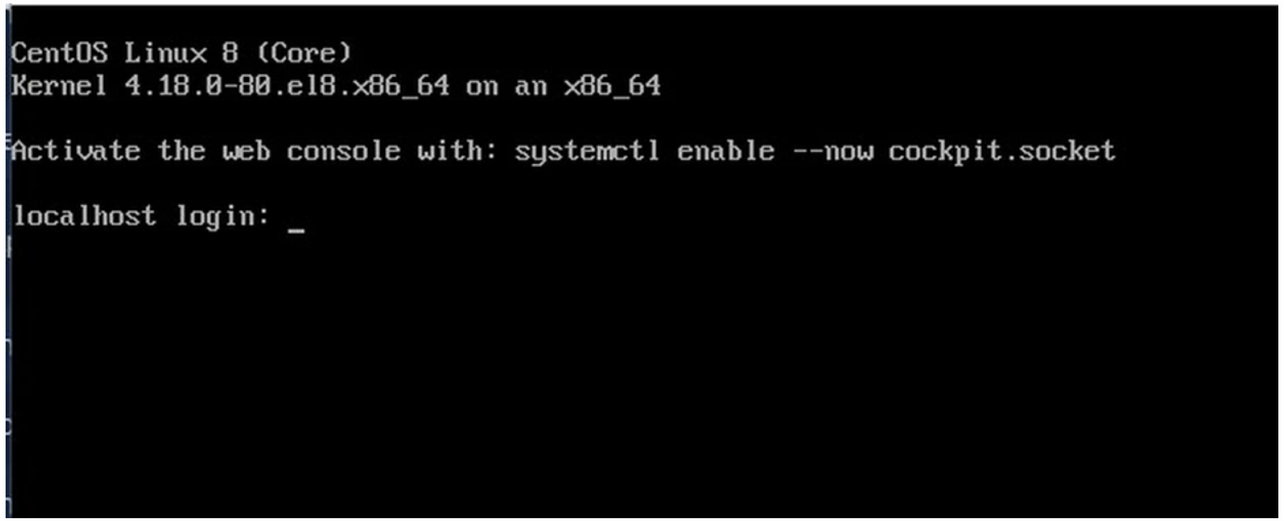
Ollscoil na Gaillimhe
University of Galway

- The latest version of cmd available in Windows 11
- Supports tabs
- Allows switching between cmd, PowerShell, Git bash or any other shell in a single app



Unix Terminal





CLI Basics

Ollscoil na Gaillimhe
University of Galway

When you open a CLI, it shows you your **current location in the file system** and a special symbol, or **prompt**, which means that the CLI is ready to take a command. This symbol is **>** on Windows and **$** on UNIX and Unix-like systems

Within a command, **>** symbol redirects the output of a command to a file, and **|** redirects the output of a command to another command as input

You can separate commands with **;** in Unix and with **&** in Windows

Some CLIs support copying and pasting, some don’t

CLIs "remember" a certain number of commands that you run during a session, and you can navigate between them using ⇅ arrows: ↑ will get you the previous command, and ↓ goes to the next one

Modern CLIs support **autocomplete** that helps you avoid typing long paths and filenames: type the first few letters of a file/directory name you need, and then press **Tab**

CLI Basics

Ollscoil na Gaillimhe
University of Galway

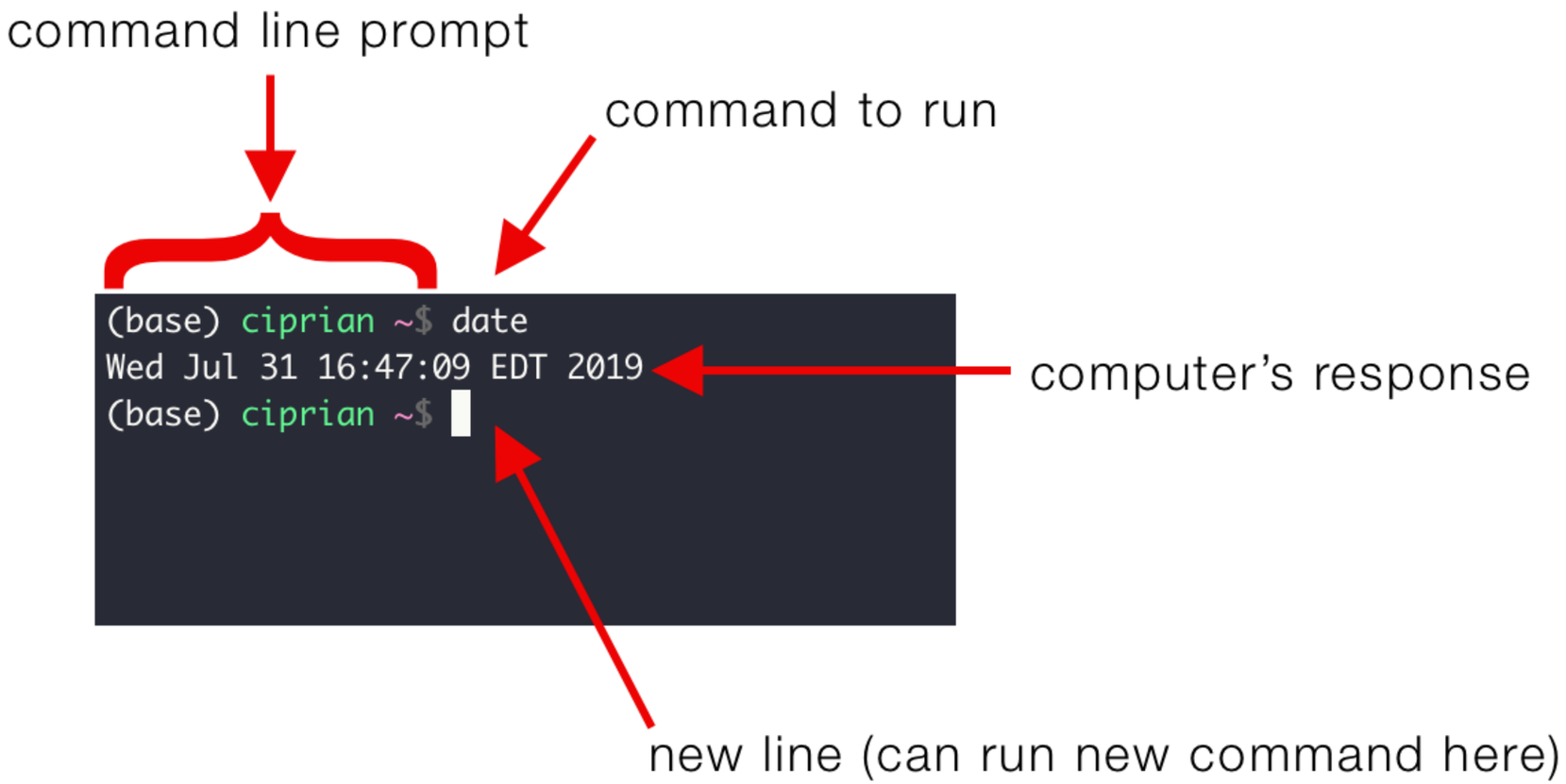
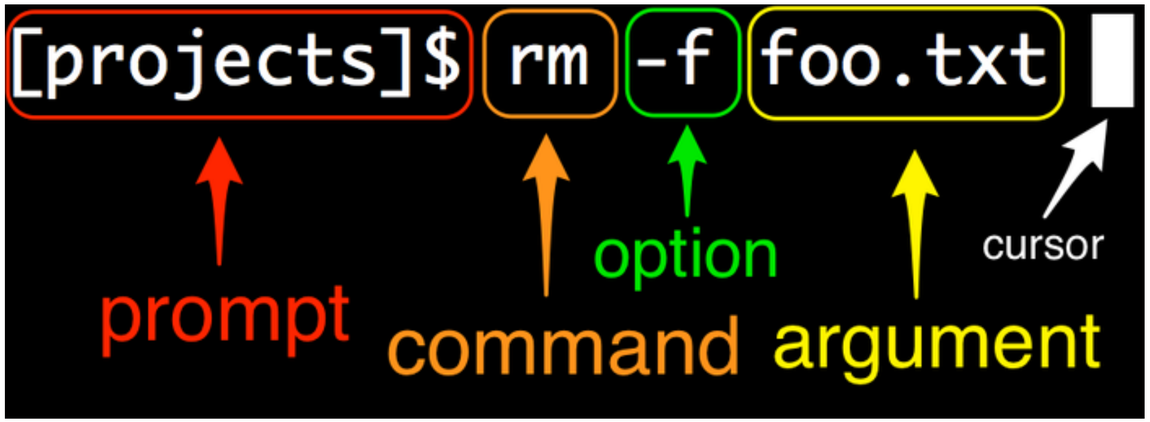


Image source: Edlitera

Anatomy of a Command



Unix

Image source: Learn Enough

Windows

Navigation

Ollscoil na Gaillimhe
University of Galway

CLIs support both relative and absolute paths to files and directories.

- `.` refers to your current working directory
- `..` refers to a directory one level up from your current working directory
- `../..` refers to a directory two levels up from your current working directory (you can go further up the tree in the same way)

cd <path> — change directory to the one specified in <path>; works for all systems

To change a drive on Windows, type the drive name first (e.g. **E:**), press **Enter**, and use **cd** to navigate further

```
0129585S@OKSANA C:\Users\0129585S\Code\digital-tools-workshops
$ cd ..

0129585S@OKSANA C:\Users\0129585S\Code
$
```

```
0129585S@OKSANA C:\Users\0129585S\Code\digital-tools-workshops
$ cd ../..

0129585S@OKSANA C:\Users\0129585S
$
```

Action	Windows	Linux / MacOS
Change directory	cd	cd
Create empty file	type nul > file.txt copy con file.txt, then Ctrl+Z	touch file.txt
Create directory	mkdir	mkdir
Delete file	del, erase	rm
Delete empty directory	rmdir	rmdir
Delete directory tree	rmdir /s	rm -rf
Print message	echo	echo
Print file contents	type	cat
Copy file	copy, xcopy	cp
Rename file	ren, rename	mv
Move file	move	mv
Search for a file by name	where	find, locate
Search for a string in a file	find	grep
Compare contents of files	fc	diff
Print list of files and directories	dir	ls
View your current directory location	chdir	pwd
Display free space	free	mem
Set environment variables	set	export
Send ICMP ECHO_REQUEST to network hosts	ping	ping
Clear screen	cls	clear
Help	help	apropos, man, whatis

Exercise

1. Open Terminal / cmd
2. Move to Desktop
3. Print a list of all files in your Desktop
4. Create a folder called “Test”
5. Print a list of all files in your Desktop
6. Move to the “Test” folder
7. Create an empty file called “test.txt”
8. Create an empty file called “new.txt”
9. Print a list of all files in your “Test” folder
10. Delete the “new.txt” file
11. Move one level up
12. Delete “Test” folder with all its contents

Image source: iStock

Find files that have a specific extension

Windows

In this command, `\b` matches the text pattern if it is at the beginning of a line and `/s` searches recursively (including subdirectories). The `/a-d` switch excludes directories, giving you only files. More about findstr tool.

```
dir /b /s /a-d | findstr ".mp3"
```

Unix

```
find . -type f -name "*.mp3"
```

Ollscoil na Gaillimhe
University of Galway

Find files that don’t have a specific extension

Windows

In this command, `\b` matches the text pattern if it is at the beginning of a line and `/s` searches recursively (including subdirectories). The `/a-d` switch excludes directories, giving you only files. The `/vi` switch indicates to exclude files containing the next parameter, the search being case insensitive.

```
dir /b /s /a-d | findstr /vi ".jpg”
```

Unix

```
find . -type f ! -name "*.jpg"
```

Ollscoil na Gaillimhe
University of Galway

Ollscoil na Gaillimhe
University of Galway

Git & GitHub

University *of* Galway.ie

Ollscoil na Gaillimhe
University of Galway

Git is a **distributed version control system (DVCS)** that allows you to

- track changes over time
- restore specific files to a previous state
- revert the entire project to an earlier version
- identify who last modified a file that might be causing an issue

Official website

Documentation

Textbook

GitHub

- A platform to share, store and manage code that uses Git as a VCS
- File size limit: 100 Mb
- One of many (cf. GitLab, BitBucket)
- Has a Desktop client
- Provides a web hosting service called GitHub Pages

Getting Started

Ollscoil na Gaillimhe
University of Galway

Registration

1. Download and install git from https://git-scm.com/downloads
2. Register on GitHub: https://github.com/

GitHub Authorisation

After you install git on your machine, there are two important things to do.

1. Tell git your name and email.

```
git config --global user.name "your name / github login"

git config --global user.email "the email you registered with on github.com"
```

2. Generate an SSH key on your machine and upload it to GitHub. Using the SSH protocol, you can connect and authenticate to remote servers without your username and personal access. A step-by-step guide from GitHub: https://docs.github.com/en/authentication/connecting-to-github-with-ssh/about-ssh

Git

In Git, you keep a copy of your files and change history on a **remote** server, from which you or your collaborators can download, or **clone** it.

Then, every collaborator with a **local** copy can contribute to the project by uploading, or **pushing** changes to the server, and get the latest changes by **pulling** them from the remote.

A VCS tracked folder is called a **repository**

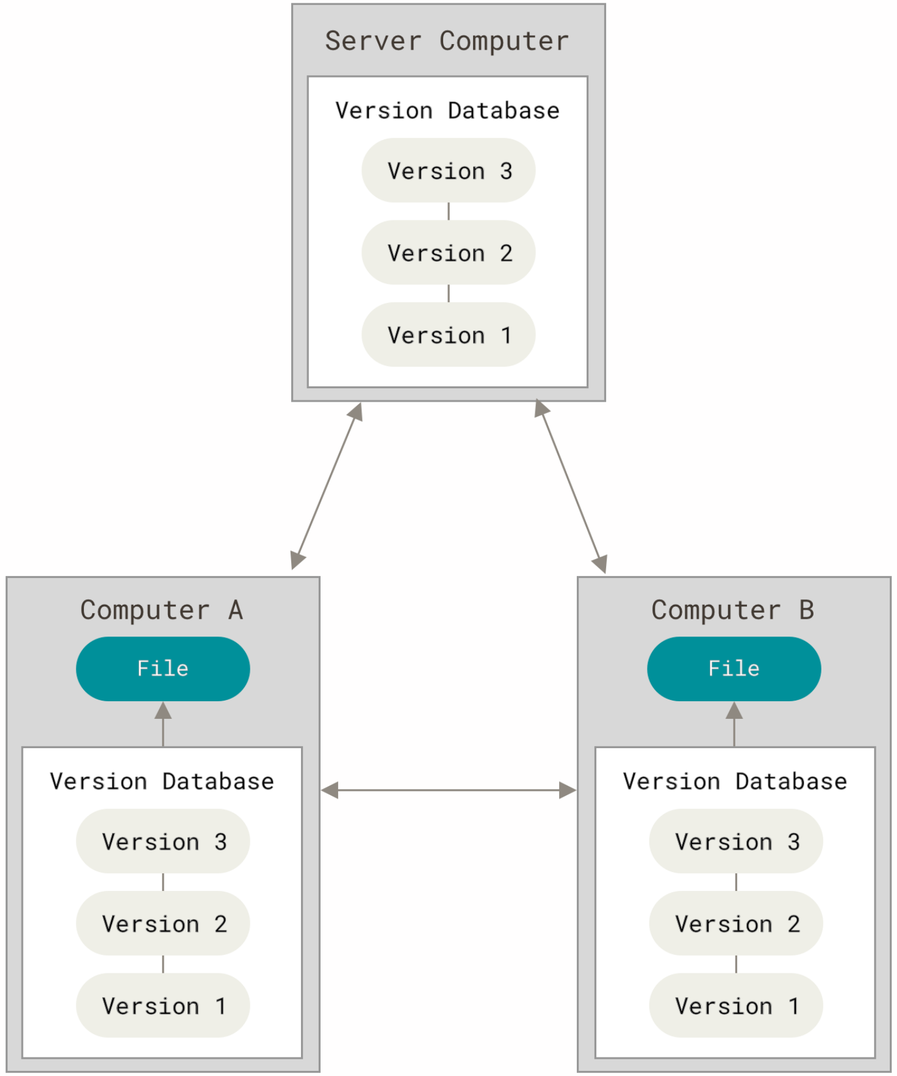


Image source: Pro Git

Git Working Areas

Ollscoil na Gaillimhe
University of Galway

Git has three main states that your files can reside in: **modified**, **staged**, and **committed**.

- **Modified** means that you have changed the file but have not committed it to your database yet.
- **Staged** means that you have marked a modified file in its current version to go into your next commit snapshot.
- **Committed** means that the data is safely stored in your local database.

You don’t have to track all files in your local git repository!

- Don’t stage them for commit (= don’t list them after `git add`)
- Use `.gitignore` file to always ignore certain files and folders

Basic Workflow

1. Copy the latest changes from the remote: **git pull**
2. Modify files in your local working tree
3. Selectively stage those changes to be recorded: **git add**
4. Record a snapshot of these changes: **git commit**
5. Send these changes to the remote: **git push**

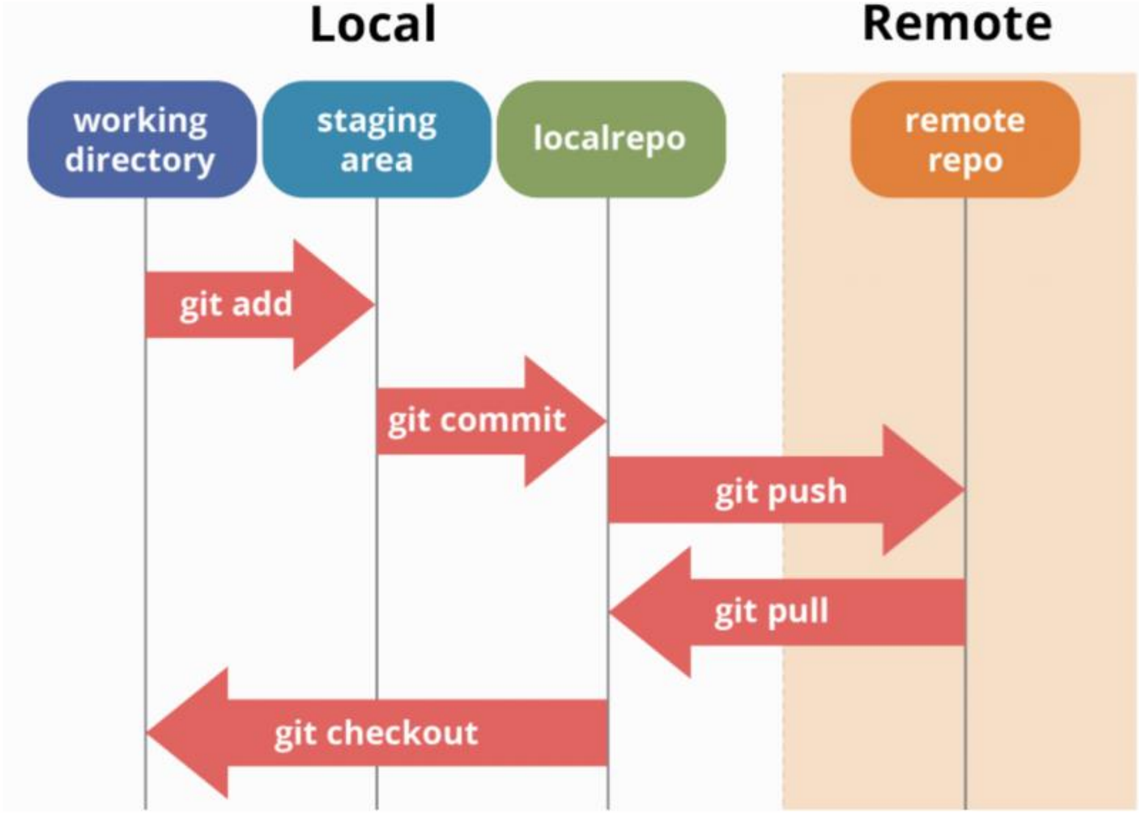


Image source: DEV

Basic commands

Ollscoil na Gaillimhe
University of Galway

Command	Action
git init	Create a local repository
git clone	Clone a repository from server
git pull	Pull changes from a remote repository
git status	Check the status of local changes
git add <PATH>	Add selected files/folders to be tracked
git add *	Add all files to be tracked
git rm	Delete files
git commit -m "Commit message"	Commit changes; -m is for commit message
git push	Push local changes to remote
git log	Check log

Correcting Mistakes

Ollscoil na Gaillimhe
University of Galway

One of the common undos takes place when you commit too early and possibly forget to add some files, or you mess up your commit message. If you want to redo that commit, make the additional changes you forgot, stage them, and commit again using the `–amend` option.

```
git commit --amend -m "Commit message"
```

Create a commit, opposite to the last one. This is safe.

```
git revert HEAD
```

Reset commit history to a specific commit. This is dangerous!

```
git reset --hard <commit id>
```

Correcting Mistakes

Ollscoil na Gaillimhe
University of Galway

Untracking files without deleting them

```
git rm --cached <FILE_NAME>
```

Example: `git rm --cached siteconfig.json`

Unstaging staged files

These two commands do the same.

```
git reset HEAD <filename>
```

```
git restore --staged <filename>
```

Unmodifying a modified file

These two commands do the same.

```
git checkout -- <filename>
```

```
git restore <filename>
```

Renaming files

```
echo "Hello world\!" > test.txt
```

```
git mv test.txt hello.txt
```

Branching

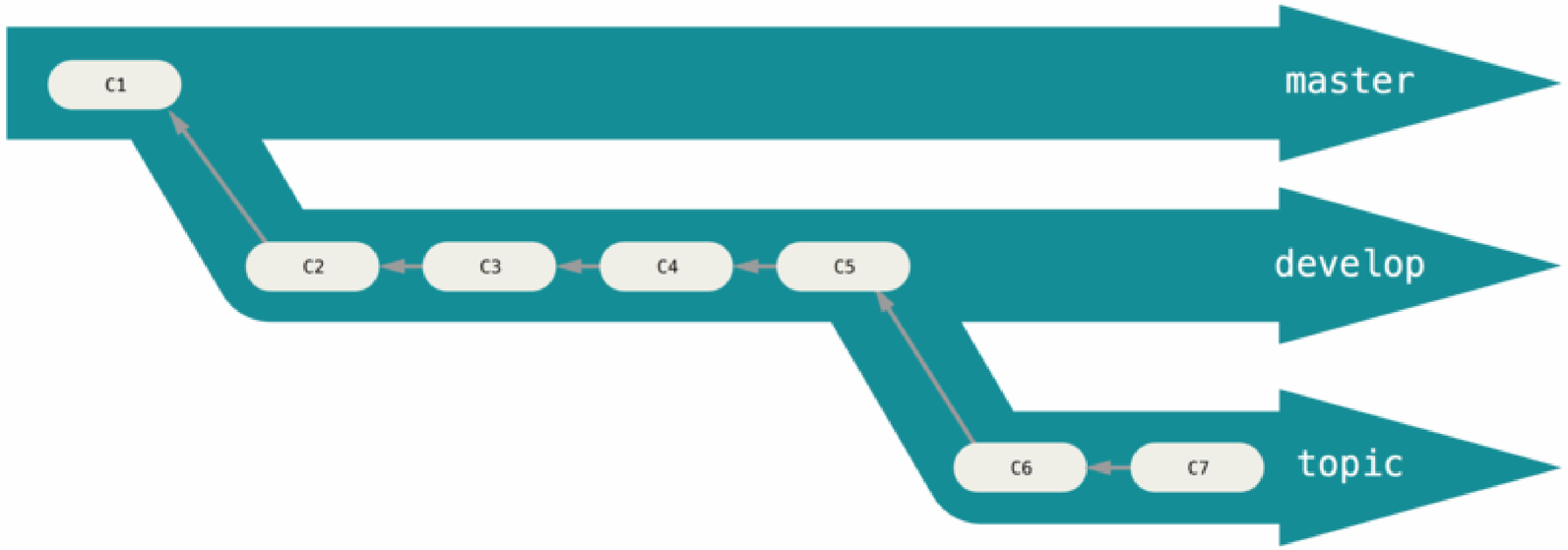
Ollscoil na Gaillimhe
University of Galway

- A **branch** in Git is simply a lightweight movable pointer to one of the commits.
- The default branch name in Git is **master**, or **main**.
- A **development** branch is used for long-term work before it is ready to be merged into master.
- A **topic** branch is a short-lived branch that you create and use for a single particular feature or related work.
- You can have as many branches as you like and name them as you like.

Action	Command
Checking branches	git branch -vv
Create a new branch	git branch <BRANCH_NAME>
Move to a branch	git checkout <BRANCH_NAME>
Add new branch to remote	git push origin <BRANCH_NAME>

Branching

Ollscoil na Gaillimhe
University of Galway



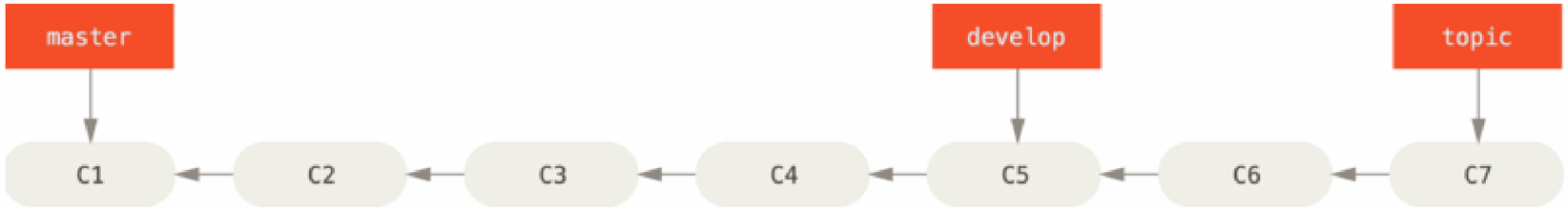


Image source: Pro Git

LibGuides

Ollscoil na Gaillimhe
University of Galway

Command Line Interface
https://libguides.library.universityofgalway.ie/DigitalTools/CLI

Git & GitHub
https://libguides.library.universityofgalway.ie/DigitalTools/GitHub