Command Line & Git

This cheat sheet provides a quick reference for essential command-line operations and Git version control workflows. From setting environment variables and running Python scripts to managing files, directories, and permissions, the command-line section equips you with the tools to streamline everyday tasks. Whether you're redirecting output, piping commands, or working with CSV files, these commands simplify working with data and files efficiently.

The Git section covers foundational version control tasks, such as initializing repositories, committing changes, and managing branches. It also includes advanced workflows like stashing changes, merging branches, and interacting with remote repositories. These commands help you track, share, and collaborate on your projects with ease, ensuring your work stays organized and secure.

Designed for data professionals, developers, and anyone working in a terminal, this cheat sheet is your go-to resource for staying productive on the command line and mastering Git. Whether you're automating tasks, exploring data, or managing code, this handy resource ensures you can work effectively and confidently.

Table of Contents

Command Line

Setting Environment Variables
Running Python Scripts
Printing Text
Changing Directory
Listing Files
Creating Directories
Deleting Files
Copying Files
Moving and Renaming Files
Viewing File Contents
Searching Text
Redirecting Output
Appending Output
Piping Commands
Changing File Permissions
Checking Disk Usage
Finding Files
Running Python Modules
Searching for Patterns
Starting a Bash Shell
Redirecting Command Output
Printing Columns with AWK
Filtering CSV Rows
Starting IPython Shell
Appending Text to File
Extracting CSV Columns
Getting CSV Statistics

Git

Initializing a Repository
Cloning a Repository
Checking Repository Status
Adding Files to Staging
Committing Changes
Viewing Commit History
Creating a New Branch
Switching Branches
Merging Branches
Adding a Remote Repository
Pushing Changes to Remote
Pulling Changes from Remote
Aborting a Merge
Stashing Changes
Applying Stashed Changes
Deleting a Branch
Initializing Version Control
Adding Remote Repository
Creating Git Branches
Merging Branches
Cloning a Repository

Command Line

Syntax for	How to use	Explained
Setting Environment Variables	`export VAR=value`	Sets an environment variable `VAR` to `value`.
Running Python Scripts	`python script.py`	Runs a Python script from the command line.
Printing Text	`echo "Hello, World!"`	Prints "Hello, World!" to the terminal.
Changing Directory	`cd /path/to/directory`	Changes the current directory to the specified path.
Listing Files	`ls -l`	Lists files in the current directory with detailed information.
Creating Directories	`mkdir new_directory`	Creates a new directory named `new_directory`.
Deleting Files	`rm file.txt`	Deletes the file named `file.txt`.
Copying Files	`cp source.txt destination.txt`	Copies `source.txt` to `destination.txt`.
Moving and Renaming Files	`mv old_name.txt new_name.txt`	Renames or moves `old_name.txt` to `new_name.txt`.

Syntax for	How to use	Explained
Viewing File Contents	`cat file.txt`	Displays the contents of `file.txt`.
Searching Text	`grep "pattern" file.txt`	Searches for "pattern" in `file.txt`.
Redirecting Output	`command > file.txt`	Redirects command output to `file.txt`.
Appending Output	`command >> file.txt`	Appends command output to `file.txt`.
Changing File Permissions	`chmod 755 script.sh`	Sets the permissions of `script.sh` to `755`.
Checking Disk Usage	`du -h`	Displays disk usage in human-readable format.
Finding Files	`find /path -name "filename"`	Searches for files named "filename" in `/path`.
Running Python Modules	`python -m script`	Runs a Python module as a script.

Command Line

Syntax for	How to use	Explained
Starting a Bash Shell	`bash`	Starts a Bash interactive shell.
Printing Columns with AWK	`awk '{print $1}' file`	Prints the first column of a file.
Filtering CSV Rows	`csvgrep -c column -m value file.csv`	Filters rows in a CSV file by column value.
Starting IPython Shell	`ipython`	Starts an IPython interactive shell.
Appending Text to File	`echo "text" >> file.txt`	Appends "text" to the end of `file.txt`.
Extracting CSV Columns	`csvcut -c column file.csv`	Extracts a specific column from a CSV file.
Getting CSV Statistics	`csvstat file.csv`	Provides statistics about a CSV file.

Git

Syntax for	How to use	Explained
Initializing a Repository	`git init`	Initializes a new Git repository.
Cloning a Repository	`git clone URL`	Clones a repository from a remote URL.
Checking Repository Status	`git status`	Displays the status of the working directory and staging area.
Adding Files to Staging	`git add file.txt`	Adds `file.txt` to the staging area.
Committing Changes	`git commit -m "message"`	Commits staged changes with a message.
Viewing Commit History	`git log`	Shows the commit history.
Creating a New Branch	`git branch branch_name`	Creates a new branch named `branch_name`.
Switching Branches	`git checkout branch_name`	Switches to the branch named `branch_name`.

Git

Syntax for	How to use	Explained
Merging Branches	`git merge branch_name`	Merges `branch_name` into the current branch.
Adding a Remote Repository	`git remote add origin URL`	Adds a remote repository.
Pushing Changes to Remote	`git push origin branch_name`	Pushes changes to the remote repository.
Pulling Changes from Remote	`git pull origin branch_name`	Pulls changes from the remote repository.

Syntax for	How to use	Explained
Aborting a Merge	`git merge --abort`	Aborts a merge in progress.
Stashing Changes	`git stash`	Stashes changes in the working directory.
Applying Stashed Changes	`git stash apply`	Applies stashed changes.
Deleting a Branch	`git branch -d branch_name`	Deletes the branch named `branch_name`.