

Developer Tools Cheat Sheet

Command Line

Command Line

- A text-based interface.
- *Synonyms*: command line interface (CLI) console

Terminal

- An OS X application that provides text-based access to a computer's operating system.
- Any device or application used for data entry and display in a computer system.
- *Synonyms*: client, computer terminal, terminal emulator

File System

- A file system is a systematic way to control how information is stored and retrieved on a computer. It describes where one piece of information stops and the next one begins. Each file system has its own structure and logic.
- *Synonyms*: NTFS (Windows' File System), HFS+ (Apple's File System), file allocation table, GFS (Global File System)

Directory

- A unit, or container, used to organize computer files into a hierarchical structure.
- *Synonyms*: folder, catalog, drawer

Path

- A sequence of symbols and names that identifies a file or directory. The path always starts from your working directory or from the root directory, and each sub-directory is followed by a forward slash.
- An *absolute*, or full, path begins with the root directory and specifies every directory above the terminating file or directory name.
- A *relative* path is a way to specify the location of a directory or file relative to another directory. It does not include the root or parent directory names.
- *Synonyms*: path name

Command

- The action we want the computer to take; always a single word.

- *Synonyms*: utility

Option

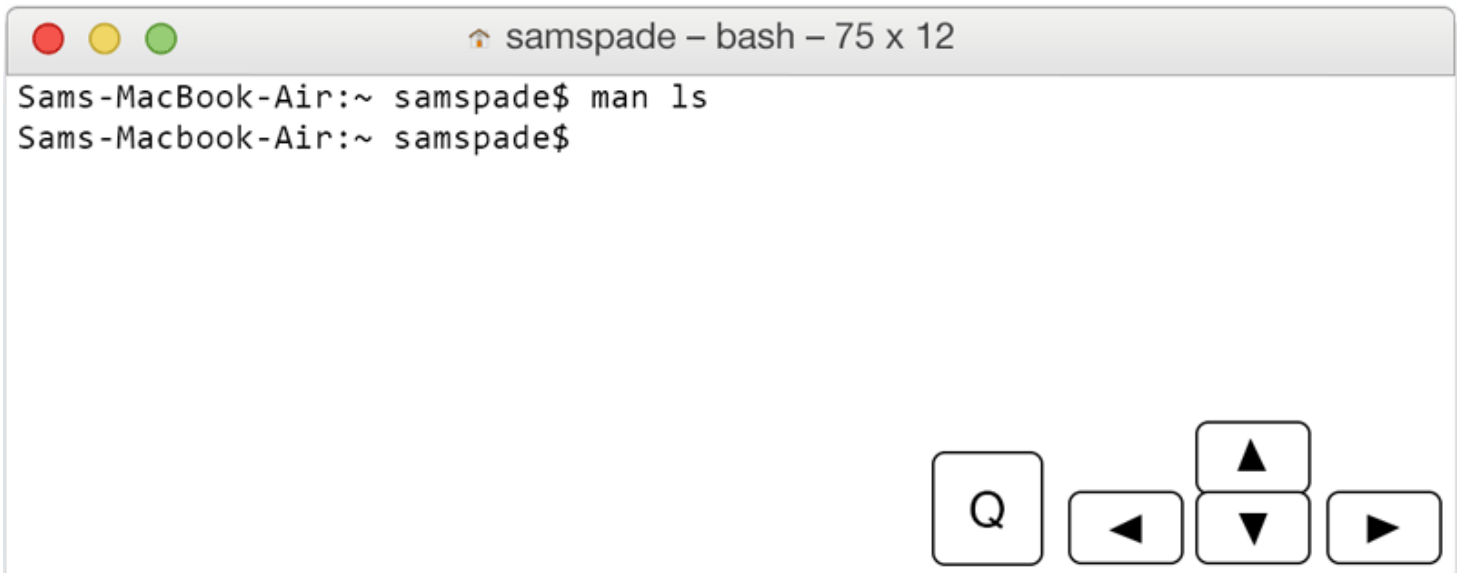
- Follows the "command" in a command line; used to modify the behavior of the command in some way.
- *Synonyms*: flag

Argument

- Follows the "command" and "options" (if any) in a command line and is used to explain what we want the command to act upon.
- The number of arguments used generally depends on the command; some don't need arguments, some require exactly one argument, some require many arguments, and some are flexible in the number they can take.

Command	Description
<code>pwd -options</code>	Prints the working directory; returns the absolute path name of the current directory
<code>ls [-options] [path/to/directory]</code>	Lists directory contents
<code>cd [-options] [path/to/directory]</code>	Changes the current working directory to the specified directory
<code>mkdir [-options] [path/to/directory]</code>	Makes a new directory
<code>rm -r [path/to/file] [path/to/file] ...</code>	Removes directories or files permanently
<code>mv [-options] [path/to/file] [path/to/directory]</code>	Moves directories or files to a new locale
<code>mv [-options] [path/to/file] [NEW_FILE_NAME]</code>	Renames a file or directory

On Mac, your terminal comes with a manual. To access more (*a lot more*) information about any command, type `man`, followed by the command name, and press `Enter`:



A terminal window titled "samspace - bash - 75 x 12" with three colored window control buttons (red, yellow, green) in the top-left corner. The terminal text shows a prompt "Sams-MacBook-Air:~ samspace\$" followed by the command "man ls" and the prompt "Sams-Macbook-Air:~ samspace\$". In the bottom-right corner of the terminal area, there are five navigation icons: a square with the letter 'Q', a square with a left-pointing arrow, a square with up and down arrows, and a square with a right-pointing arrow.

You can scroll through a manual entry using the arrow keys or space bar. To quit this view and return to your prompt, type `q`.

Git

Git

- An open-source program for tracking changes in text files. It was written by the author of the Linux operating system, and is the core technology that GitHub, the social and user interface, is built on top of.

Commit

- An individual change to a file (or set of files).
- It is similar to saving a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who.
- Commits usually contain a commit message which is a brief description of what changes were made.
- *Synonyms*: a revision

Diff

- A diff is the difference in changes between two commits, or saved changes.
- The diff will visually describe what was added or removed from a file since its last commit.

Remote

- The version of something that is hosted on a server, most likely GitHub.com. It can be connected to local clones so that changes can be synced.

Repository or Repo

- The most basic element of Git.
- A repository is a project's folder, containing all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

Fork

- A personal copy of another user's repository that lives on your account.
- Forks allow you to freely make changes to a project without affecting the original.
- Forks remain attached to the original, allowing you to submit a pull request to the original author to update with your changes.

Clone

- A copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy.
- With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online.
- It is, however, connected to the remote version so that changes can be synced between the two.
- You can push your local changes to the remote to keep them synced when you're online.

Push

- Pushing refers to sending your committed changes to a remote repository such as GitHub.com.
- For instance, if you change something locally, you'd want to then push those changes so that others may access them.

Command	Description
<code>git init</code>	Tells Git to start monitoring the current folder you're in. In other words, for my working directory, create a new "timeline" where we can manage our source code.
<code>git status</code>	Get the current status of Git. Files that are "staged" (about to be committed), and files that are unstaged (files that have changed since the last commit, but are not about to be committed) will both show up here
<code>git add</code> <code>path/to/directory/or/file</code>	Add a file to the "stage". The stage can be thought of as an in-between state between the last commit and what is ready to be committed. Once the command <code>git commit</code> is run, all staged files will be committed to the timeline.

Command	Description
<code>git commit -m "Commit message"</code>	Commit all staged files to the timeline. If -m "Commit message" is omitted, Git will open your default text editor (typically Vim) to enter a longer message. If for any reason Vim is opened, you can close it by typing :q.
<code>git log</code>	Visualize the timeline. You can scroll with the arrow keys or j, k, or can exit by typing q.
<code>git diff path/to/directory/or/file</code>	Show the tracked but unstaged changes of the given file or directory.
<code>git clone http://path/to/repo</code>	Create a new local git repo copied from a remote one
<code>git push origin master</code>	Send local changes to tracked remote repository

Navigating the Command Line

1) \$ cd Downloads 2) \$ cd World 3) \$ ls 4) \$ ls -a Europe

Controlling Files with Command Line

Part 1:

1) Command + Space. Inside search bar, type "Terminal." 2) \$ cd Downloads 3) \$ mv World .. will move to parent directory, in this case the desktop. 4) \$../ to move back to the parent directory, then \$ ls.

Part 2:

1) \$ cd World 2) \$ mkdir Antarctica 3) \$ ls 4) \$ cd World (only if pwd is not World) 5) \$ cd South America \$ ls 6) \$ mv Argentina ../North_America \$ mv Chile ../North_America \$ mv Brazil ../North_America 7) \$../ (back to World directory) \$ rm -r South_America 8) \$ mkdir Americas mv North_America Americas