Terminal Basic Command

Clear the Window

```
$ clear
```

Simply put, this clears the current window. (Within Terminal in OS X, you can still scroll up to see what was there. This command simply clears the current view).

Navigating to Folders

You can navigate to a folder, such as **Utilities**, by typing the following command:

```
$ cd /Applications/Utilities
```

If you want to quickly jump back to your home folder, you can simply enter:

```
$ cd
```

With folders that contain spaces, there are two ways you can do it:

```
$ cd /Volumes/Macintosh\ HD/
$ cd "/Volumes/Macintosh HD/"
```

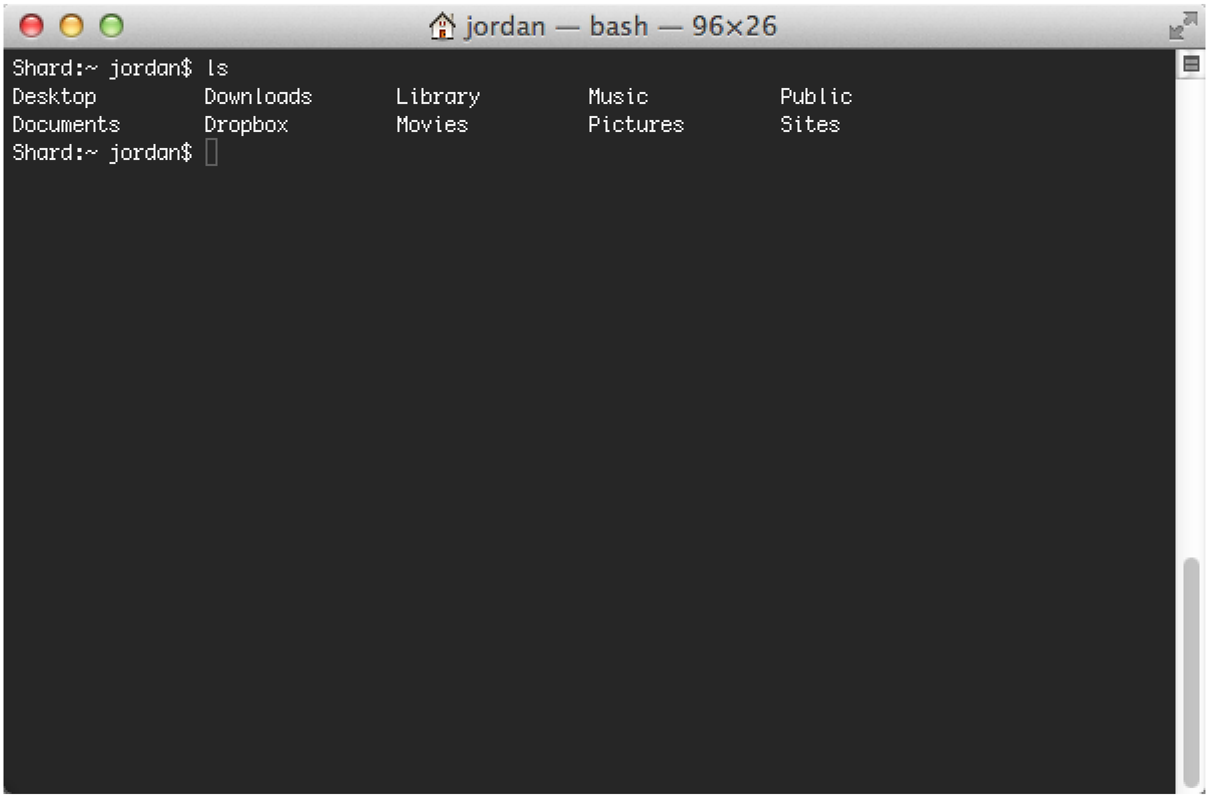
The first method uses the \ backslash to “escape” space. Since the command line treats spaces as part of commands, the \ character tells Terminal to ignore the space that immediately follows. Alternatively, you can simply wrap the folder’s path in quotation marks and any spaces it contains will be ignored.

Navigate back to your home folder before continuing.

List the Contents of a Directory

You can view the contents of a folder by entering:

```
$ ls
```

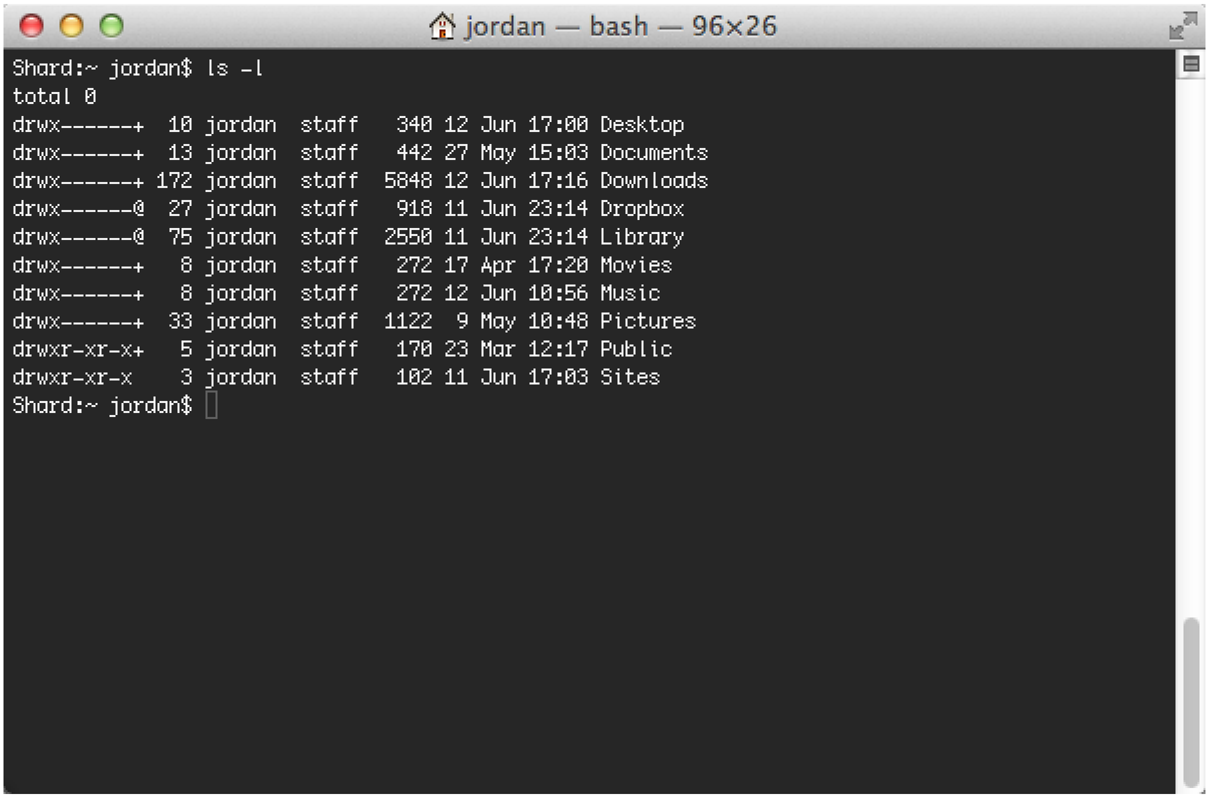


```
Shard:~ jordan$ ls
Desktop         Downloads       Library         Music           Public
Documents       Dropbox         Movies          Pictures        Sites
Shard:~ jordan$ 
```

The default output of 'ls'

To view it in a list format, enter:

```
$ ls -l
```



```
Shard:~ jordan$ ls -l
total 0
drwx------+  10 jordan  staff   340 12 Jun 17:00 Desktop
drwx------+  13 jordan  staff   442 27 May 15:03 Documents
drwx------+ 172 jordan  staff  5848 12 Jun 17:16 Downloads
drwx------@  27 jordan  staff   918 11 Jun 23:14 Dropbox
drwx------@  75 jordan  staff  2550 11 Jun 23:14 Library
drwx------+   8 jordan  staff   272 17 Apr 17:20 Movies
drwx------+   8 jordan  staff   272 12 Jun 10:56 Music
drwx------+  33 jordan  staff  1122  9 May 10:48 Pictures
drwxr-xr-x+   5 jordan  staff   170 23 Mar 12:17 Public
drwxr-xr-x    3 jordan  staff   102 11 Jun 17:03 Sites
Shard:~ jordan$ 
```

The output of 'ls' with the '-l' flag

If you want to view *all* files, including hidden ones:

```
$ ls -a
```

The above command will display all files, including those hidden ones that start with a period. What if we want to view the folder as a list but *also* view all the hidden files?

```
$ ls -la
```

Yep, it’s as simple as combining them together! I think you’re suitably warmed up, so let’s kick it up a notch!

Navigate back to your home folder before continuing.

Ditto

A hidden gem in OS X is Apple’s **ditto** command. I’ve found this incredibly useful in copying large amounts of data as it can run within a Terminal window that contains more useful information about its progress than the more traditional Finder copy window.

```
$ ditto -V /old/work/ /new/work/
```

The above command will copy our “work” folder to a new location. Actually, what happens is the command copies the *contents* of our “work” folder to a new “work” folder.

Adding **-V**, meaning *verbose* prints a line to the Terminal window for every file that’s being copied. It’s extremely useful as you can see exactly what file is being copied at any time.

Ditto is a command I use frequently and have often copied many gigabytes of data between hard drives using it because it seems to be so reliable.

Disable Screenshot Drop Shadows

If we ever take a screenshot of a window in OS X, by default it will always show a drop shadow, adding wasted pixels. If you’d prefer to have your screenshots drop shadow-free, you can use the following command:

```
$ defaults write com.apple.screencapture disable-shadow -bool TRUE
```

You can either restart your Mac to see the changes or, using another command, see them instantly:

```
$ killall SystemUIServer
```

This command stops the graphical front-end to OS X for a split second. Don’t worry, it *has* to run all the time so as soon as it is stopped, it restarts instantly - but updated with our new option. Take a screenshot and you’ll now notice that the shadow is no longer included.

To reverse the changes, you can enter the following command:

```
$ defaults write com.apple.screencapture disable-shadow -bool FALSE
```

```
$ killall SystemUIServer
```

To see the changes immediately.

Change Screenshot File Format

PNG is probably the best format to use for screenshots, but it can certainly take up a bit of space. If you’d prefer to use PDF format, you can use the following:

```
$ defaults write com.apple.screencapture type PDF
$ killall SystemUIServer
```

To revert the changes, enter the following:

```
$ defaults write com.apple.screencapture type png
```

```
$ killall SystemUIServer
```

Change Screenshot Location

Whilst taking screenshots is great, it can quickly clutter up the Desktop. To change this, we can use:

```
$ defaults write com.apple.screencapture location /drag/location/here
```

```
$ killall SystemUIServer
```

Rather than deal with pesky long path names, you can just type the first part of the command and, after the word “location”, drag a folder you’d like your screenshots to save to. To undo the changes, you can use:

```
$ defaults write com.apple.screencapture location ~/Desktop
```

```
$ killall SystemUIServer
```

Tip: The squiggle, ~, actually represents your home folder. In the example above, this would actually mean /Users/jordan/Desktop. A useful timesaver!

Enable AirDrop on Older Macs

AirDrop is a great feature to send files between Macs quickly and easily without dealing with pesky file sharing apps. Trouble is, it’s only on WiFi and on more recent Macs.

If your Mac doesn’t currently support AirDrop, you can enter the following command to use it, as well as to use it over ethernet:

```
$ defaults write com.apple.NetworkBrowser BrowseAllInterfaces -bool TRUE
```

You’ll need to restart the Finder, the quickest way is to use the **killall** command which will restart it:

```
$ killall Finder
```

To revert the changes, enter:

```
$ defaults write com.apple.NetworkBrowser BrowseAllInterfaces -bool FALSE
```

If you'd like more information on enabling AirDrop, we've covered it in more detail in our previous tutorial "*How to Enable AirDrop on Ethernet Connected and Unsupported Macs*".

Show Hidden Files and Folders

To view hidden files and folders in OS X:

```
$ defaults write com.apple.finder AppleShowAllFiles -bool TRUE
```

```
$ killall Finder
```

Once the Finder restarts, you'll see a lot of files you never saw before, usually starting with a period (which denotes a hidden file in OS X).

To revert the changes:

```
$ defaults write com.apple.finder AppleShowAllFiles -bool FALSE
```

```
$ killall Finder
```

We've also covered hidden file visibility in our previous tutorial "*Revealing Hidden Files in OS X*".

Stress Test Your Mac

If you're wanting to run some stress tests on your Mac, there's a simple command we can enter that will instantly utilize 100% of your CPU:

```
$ yes
```

Apple technicians use this frequently when dealing with troublesome Macs that might be crashing under load and it's a very quick way of stressing a Mac. To cancel the command, press **Ctrl-C**.

View File System Usage

Sometimes we want to see what our Mac is doing when it comes to reading and writing to disk, especially if we're trying to identify something that's preventing us from ejecting a drive. To do this, we use:

```
$ sudo fs_usage
```

By entering **sudo**, we are requesting to run this as a super user, which requires entering our Mac's login password. Once done, you're presented with a constant stream of information as processes are accessing your disk.

Some processes you are likely to see are Time Machine (backupd) and Spotlight (mds).

To cancel, you can press **Ctrl-C**.

View the Contents of Any File

Sometimes we have a file and we just don't know what format it is. If it's likely to be a text format, we can attempt to open it in **TextEdit**, but it doesn't always work. We can investigate *any* file from the Terminal by using:

```
$ cat /path/to/file
```

No matter what the file is, this will open it. On files like MP3s or videos, it'll likely be gibberish. But if you're trying to recover a corrupt document, this might help.

Rebuild Spotlight

Spotlight can occasionally not work as well as we’d like. Sometimes, we just need to tell Spotlight to create a new index so it can find files as fast and efficiently as possible. To do this:

```
$ sudo mdutil -E /Volumes/DriveName
```

The above command will delete Spotlight’s index, forcing it to start a new one. An easier way of completing the command is to enter the first part of it and, for the location, just drag a hard drive from your desktop to the Terminal window.

Remove Duplicate “Open With…” Entries

A common bug in OS X is seeing duplicate apps within the “*Open With…*” menu which can get quite full! To fix it:

```
$ /System/Library/Frameworks/CoreServices.framework/Frameworks/LaunchServices.framework/Support/lsregister -kill -r -domain local -domain system -domain user
```

The above command will work in OS X Leopard and above. Unlike the other commands, I’d recommend restarting your Mac.

Enable Text Selection in Quick Look

Quick Look is great but wouldn’t it be even better if we could select text from within Quick Look without having to open the document? Well with this command, you can:

```
$ defaults write com.apple.finder QLEnableTextSelection -bool TRUE
```

```
$ killall Finder
```

To revert the changes:

```
$ defaults write com.apple.finder QLEnableTextSelection -bool FALSE
```

```
$ killall Finder
```

Open Files In Any Application

You can open documents directly from the Terminal very easily, but what about opening them in another application that isn’t the default one for its file type?

```
$ open -a /Applications/AppName.app /path/to/file.txt
```

Doing the above will open our text file in the app *AppName*. Simply change the app and file to whatever you need.

Check the Uptime Of Your Mac

It can be days, weeks or even months before we reboot our Mac and it can sometimes be so long that we can't even remember when we last did. To see how long our Mac has gone without a restart, we can use:

```
$ uptime
```

This will display the current time and how long our Mac has been running.

Install OS X Software Updates

Despite Software Updates moving to the **App Store** in Mountain Lion, we're able to use the command line to install system updates without having to launch it. To see available software updates for your Mac:

```
$ sudo softwareupdate -l
```

After a few minutes, you'll be given a list of available updates.

If you'd like to install all available updates, enter:

```
$ sudo softwareupdate -ia
```

Display A Custom Message At The Login Window

When using a multi-user Mac, it's often good to have a message display to provide some useful information, such as "*Hey, sticky hands! Others have to use this keyboard, y'know.*"

```
$ sudo defaults write /Library/Preferences/com.apple.loginwindow LoginwindowText "Hey, Sticky Hands!"
```

Next time you log out or restart your Mac, your message will appear. To remove the message entirely:

```
$ sudo defaults delete /Library/Preferences/com.apple.loginwindow LoginwindowText
```

Start a Simple HTTP Server in Any Folder

If you're needing to quickly test some HTML that you're working on, start a simple web server within *any* folder on your Mac. Navigate to the folder to use and enter:

```
$ python -m SimpleHTTPServer 8000
```

The number at the end is the port to use, open your browser and visit *http://localhost:8000* You can use the default of port 80 if you wish and remove the port number entirely.

When you're finished, simply press **Ctrl-C**.

Run the Same Command Again

You will have seen a few of these commands have required us to enter *sudo* first, to run them as the root super user. Sometimes, we can enter a long command, only to find out we forgot to put *sudo* first. Thankfully, we don't need to re-write the entire command again (or copy and paste). Instead, we can use a very simple shortcut that stands in place of your previously written command:

```
$ !!
```

If we needed to enter the same command again but with *sudo* in front, we can just type:

```
$ sudo !!
```

Download a File Without a Browser

If you'd like to download a file without using a browser, we can use the following command:

```
$ curl -O http://appldnld.apple.com/iTunes11/091-6058.20130605.Cw321/iTunes11.0.4.dmg
```

This will download any URL you enter to the folder you're currently in. Some browsers try to automatically open files when they're downloaded or add a pesky file extension when you don't want them (especially when dealing with text files that have something else besides the *.txt* extension).

Shutdown Your Mac, With or Without a Delay

To shut down your Mac immediately:

```
$ sudo shutdown -h now
```

To restart your Mac immediately:

```
$ sudo shutdown -r now
```

We can even add a time delay (in minutes) if we wish:

```
$ sudo shutdown -r +60
```

Whilst this might not be very useful on your own Mac, if you spend any time remotely logged in to Macs via the command line, these can become very useful at restarting remote servers.

Disable Mail’s Reply Animation in Mountain Lion

Mail has a great looking animation whereby if you hit reply to any message, a compose window animates into view. It’s not for everyone and sometimes just having the window appear instantly. To turn off the animation:

```
$ defaults delete com.apple.mail DisableReplyAnimations -bool TRUE
```

Quit and relaunch Mail for the changes to take effect. To revert the changes:

```
$ defaults delete com.apple.mail DisableReplyAnimations -bool FALSE
```

Prevent Your Mac From Sleeping

We’ve all left our Mac doing something that will take some time to do, only to come back and find it went to sleep. We can prevent this from happening by using a simple, and humorously named, command:

```
$ caffeinate
```

Entering the command on its own will keep the Mac awake until you stop it by pressing **Ctrl-C**. You can add a time parameter with:

```
$ caffeinate -u -t 600
```

The number represents the number of seconds, so our command above will stop the Mac from going to sleep in the next ten minutes.

Create a File Of Any Size

There are occasions when we just need to a file of a particular size for testing. I often find it useful to copy a 1GB file over a network to get an idea of speeds. Finding a file that is the size we want is tricky, but with the command line, we can create an empty file of any size that we want.

```
$ mkfile 1g test.abc
```

You can specify the file size in bytes (b), kilobytes (k), megabytes (m) or gigabytes (g). The above example creates a test file of 1GB called *test.abc* but you can name it whatever you wish and it doesn’t need to have a file extension.

Continually Monitor the Output of a File

This one is for all the budding system administrators out there. If you'd like to keep monitoring a text file and view any changes to it as they're made, there's a suitable command that will constantly monitor your chosen file and display any new lines as they're added, perfect for monitoring system log files.

```
$ tail -f /var/log/system.log
```

Your Terminal window will constantly watch your specified file (in this case, the *system.log* and every time another line is added, it will print it on the screen.

To cancel, press **Ctrl-C**.

Get Your Network IP Address

Sure, we could open *System Preferences*, select *Network* and then view our IP address information, but doing it through the command line is so much cooler!

```
$ ipconfig getifaddr en0
```

The term **en0** represents the network interface to use. Similar to some programming languages, the first network interface starts at zero and then counts up, so if you have two interfaces (such as ethernet and WiFi) then they would be **en0** and **en1**, respectively.

By default, your Mac will always designate a wired network connection as **en0**, unless your Mac doesn't have built-in ethernet. You can substitute **en0** with **en1** if you're not using your wired connection.

Get Your External IP Address

The most common way of finding out your public IP address (the one that we're provided by our ISP) by visiting a site such as What Is My IP or even typing "what is my IP address" into Google.

We can actually use the Terminal to discover our public IP address using a similar service as above called IP Echo.

```
$ curl ipecho.net/plain; echo
```

As soon as you run the command, you're presented with your external IP address. Whilst it might not be as quick to type as visiting one of the aforementioned sites, we can do a lot with the Terminal output, especially if we're wanting to include it in some sort of script.

Test Network Connectivity

You're likely to have heard of the term "ping" at some point. Ping sends very small bits of information over a network to a remote computer, timing how long it takes for a response to be received. It's useful to see if an IP address is working or if a website might be down.

```
$ ping -c 10 www.apple.com
```

Running the above command will send 10 packets of information and provide detailed information about the response time. You can substitute *www.apple.com* for any other domain name or IP address.

Disable Google Chrome’s Two-Finger Swipe Navigation

For Macs with a built-in multitouch trackpad or used with the Magic Trackpad, Google Chrome offers its own version of two-finger swiping to go forward and back. Whilst it’s easy to change this behavior in Safari, doing so in Google Chrome requires a little Terminal trickery:

```
$ defaults write com.google.Chrome.plist AppleEnableSwipeNavigateWithScrolls -bool FALSE
```

Quit and restart Google Chrome to see the changes and you’ll notice you can’t two-finger swipe anymore. To revert the changes:

```
$ defaults write com.google.Chrome.plist AppleEnableSwipeNavigateWithScrolls -bool TRUE
```

Prevent Apps From Saving to iCloud by Default

iCloud is a great service but if you prefer to use Dropbox, or just save files to your Mac, then you’ll no doubt have been frustrated that many iCloud-compatible apps offer iCloud as the default *Save…* destination. Thankfully, we can change this:

```
$ defaults write NSGlobalDomain NSDocumentSaveNewDocumentsToCloud -bool FALSE
```

Quit and relaunch any app that uses iCloud and now any new documents will default to your Mac in the save dialog rather than iCloud.

To revert the changes:

```
$ defaults write NSGlobalDomain NSDocumentSaveNewDocumentsToCloud -bool TRUE
```

Make the Icon of Any Hidden App in the Dock Translucent

Hiding apps when you're not using them is a great way to free up screen space but there's not really any way to distinguish between which app is hidden and which isn't. We can alter the Dock so any apps that are hidden will have a translucent icon:

```
$ defaults write com.apple.Dock showhidden -bool TRUE
```

```
$ killall Dock
```

To revert the changes:

```
$ defaults write com.apple.Dock showhidden -bool FALSE
```

```
$ killall Dock
```

Autocomplete Paths

When entering a long path name such as:

```
/Volumes/Macintosh\ HD/Users/admin/Library/Application\ Support/
```

It can be very tedious to enter it all out. What we can do instead is use the **Tab** key to autocomplete. To use it, simply start entering any command where you're needing to type a path and once you've entered the first or second letter, press *Tab*. If there's only one folder that the first few characters could autocomplete to, it will do so. If not, you can press *Tab* twice and it will jump as far as it can, then display all potential files and folders you might want to get to.

To try this, enter the following but instead of pressing *Enter*, press *Tab* instead:

```
$ cd /Us
```

You'll find it then autocompletes to:

```
$ cd /User
```

You'll likely hear an alert tone as well. That's because there are two folders you might want access to: */Users* and */User Information*. You can then add the last character to the path and press *Enter*.

It works very similar to autocomplete of words in iOS.

Make Your Mac Talk

Mac OS X has some great speech functionality built-in and we can even have it say anything we want:

```
$ say "This Mac runs OS X, not OS ex"
```

We can even go one better and have it read any text file we like:

```
$say -f /path/to/file.txt
```

Our Mac can then say whatever was in the text file.

Restore a Disk Image to an External Drive

You can mimic the behavior of *Disk Utility* and restore a disk image file to a volume connected to your Mac:

```
$ sudo asr -restore -noverify -source /path/to/diskimage/dmg -target /Volumes/VolumeToRestoreTo
```

It will skip verification (you can remove that part if you want to verify it but it can take some time) and you can restore an image, such as a copy of OS X, directly to a volume or partition.

Turn Off Dashboard

The dashboard was once the future of quick-to-access apps such as a calculator and sticky notes. Despite being quite popular for a few years, it’s quickly faded into obscurity. It’s still around and usually opened accidentally.

I use **Mission Control** extensively and have it positioned on the far left but, honestly, I prefer it gone completely. Thankfully, *Dashboard* can be permanently silenced:

```
$ defaults write com.apple.dashboard mcx-disabled -boolean TRUE
```

```
$ killall Dock
```

You’ll find that *Dashboard* is no longer running, along with any widgets you might have had inside. Don’t worry, you can bring it back if necessary:

```
$ defaults write com.apple.dashboard mcx-disabled -boolean FALSE
```

```
$ killall Dock
```

View All Active Processes

The app **Activity Monitor** can give us a detailed view of what our Mac is currently doing. We can replicate much of its functionality in Terminal:

```
$ top
```

Once running, we can easily see all the current processes, listed by processor usage with the processes using the most CPU moving to the top of the list.

Additionally, there’s plenty of other information at the top of the window, from how many processes are running, how much RAM is currently in use to the amount of network traffic we have generated.

See A List of All The Commands You’ve Entered

For our final one, we can enter a command to view all the commands we’ve entered at the command line:

```
$ history
```

All of the previous commands that you’ve run within Terminal are listed here, giving you the ability to go back and see what you’ve already done.