

# Work Flow for Solo Developers and Small Teams

There's no audio yet.

We'll start at 2PM (Eastern Time).

[emma@emmajane.net](mailto:emma@emmajane.net)



# Today

1. Developer's Environment  
(The Command Line)
2. Version Control Basics
3. Developer Tools for Larger Teams

# “I’ve Tried, And I Can’t Learn This Stuff”

- It’s not your fault. Honest.
- The way we teach web stuff isn’t the way that you probably need to be exposed to the information in order to learn it.
- Blame the teachers, not yourself.
- Or maybe not **blame** but, be persistent when working to solve important and sticky problems.

# How Do We “Teach” Web Dev?

- RTFM: read the manual
- Here are all the commands, here are all the options. Memorize everything, and figure out later how to apply the knowledge.

# How Do We “Self-Teach” Web Dev?

- Guess at what the problem is.
- Search on the internet.
- Find someone else’s description of how they solved, what you hope is a similar problem.

# Adults Learn Best When They Can Be Selfish.

Andragogy assumes the following about the design of learning:

- Adults have the need to know why they are learning something.
- Adults learn through doing.
- Adults are problem-solvers.
- Adults learn best when the subject is of immediate use.

# Define the Problems You Truly Want to Solve.

- Your problem might sound like:  
My client keeps changing his mind, and but they don't want to pay me to redo the work.
- Your problem doesn't sound like:  
My client wants me to memorize all the parameters for using Git at the command line.

# Solve Real Problems

1. Define your *real* problem clearly.
2. Learn how to use a tool to get your problem solved.
3. Try solving the problem. Take notes about how smooth it was to solve your problem. Write recommendations to your future self on how you'd solve the problem in the future now that you know what you know.



# Disaster Recovery Planning

AKA Motivated Learning via Real Problems

Let's solve problems before they occur (or at least have a really awesome Plan B if something does go wrong).

Problem:

# What version of the file is on my server?

- “diff” -- what's the difference?
- never edit directly on the server, always “pull” from another location
- install only code which has a version associated with it (e.g. module.info)

Problem:

# What Was I Thinking When I Wrote This?

- add in-code documentation
- use frequent commit messages, but only share (or “push”) relatively stable code
- put radically different, or unrelated, ideas into different branches
- put related ideas, which are allowed to evolve, into the same branch, but add tags to show milestones
- for very significant milestones, you may want to have different (numbered) branches

Problem:

# Untested Code (Eventually) Breaks Stuff

- create a development environment on another machine.
- copy your data down (backup and migrate)
- keep everything versioned (have a central repository that is web-accessible which you can “pull” changes from for both your local env + dev server)
- upload your tested configuration changes.

Problem:

# My Client Changed Their Mind ... Again

- You need a giant undo button to be able to roll back your code to a previous state.
- build all three designs (get basics right, then modify only bits but tag between each)
- add lots of “milestones” to your code so that you can see **easily** where things changed

Problem:

# I Changed **Something**, And Broke Stuff.

- Make only very small changes before making commits.
- Commits in distributed version control systems are like an “undo” button you can apply after restarting your computer.
- Make only related changes within one commit. e.g. only font changes; vs. only colour changes

Problem:

# My Computer Died...Now What?

- having your configuration files saved (and perhaps versioned) means you can quickly recover an entire system

Problem:

# Two Clients Want Something Similar...

- one central repo
- create branches for each client
- where it makes sense, merge common functionality back into the main branch / trunk



Problem:

## My [Collaborator] Overwrote My Work.

- You're working with a very tiny team and you think you don't need a whole source control system.
- Except you keep overwriting each others' work.
- Create a centralized repository that you both check your work into and push from that centralized server to the "live" site.
- On a regular basis "pull" your partner's work from the centralized repo.

# Homework

Write a list of all the problems you'd like to solve.  
Make sure your “problem” includes:

- A description of all the people involved.
- A description of what makes the problem “bad”.
- A description of what the situation would look like once “fixed”.

# A Developer's Environment





# The command line is used in many free, online resources.

```
git add filename.txt
```

## Committing Files

The `Git add` command is normally followed immediately by the `Git commit` command. When we **commit** something, we are saying that we want this to be a snapshot of our work.

When we commit a file in Git, we need to give it a commit message. The **commit message** just explains what we are committing and why. Think of your commit message as sort of your notes on the particular snapshot of the file.

The command for committing is:

```
git commit -a -m "This is my commit message!"
```

The last bit within the quotes is your commit message.

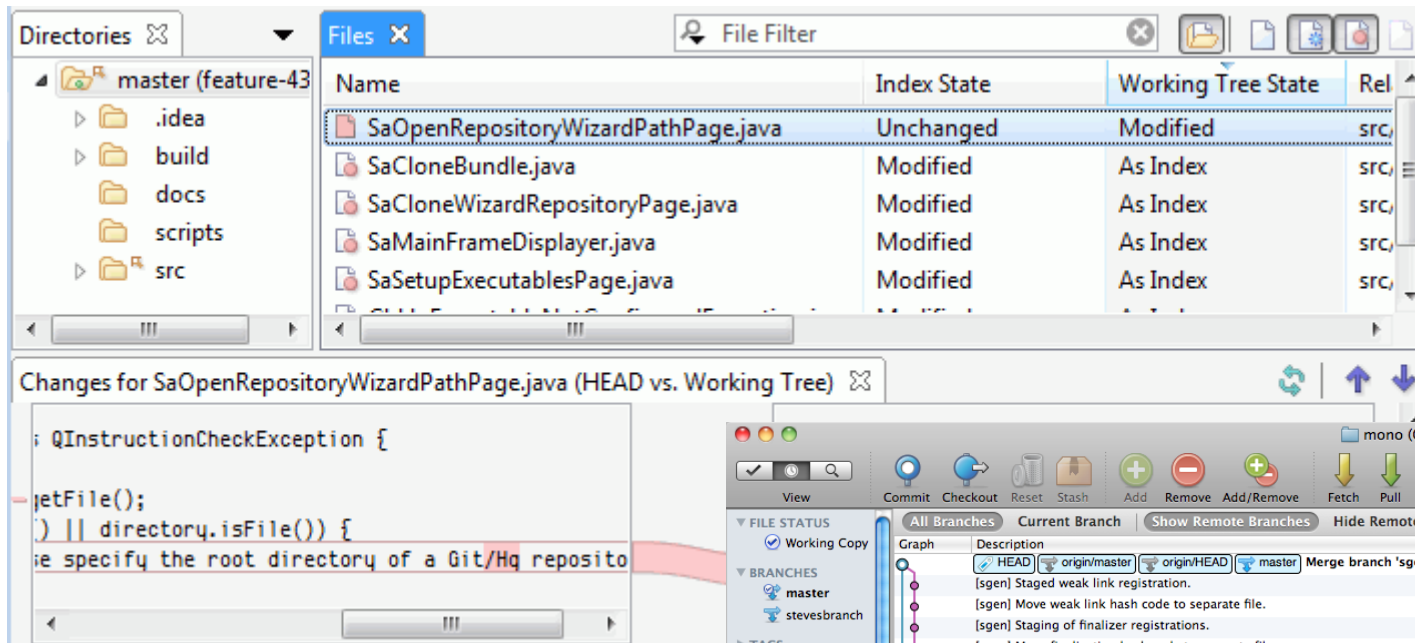
## Setup Your First Instance

Vagrant is so easy to use that you can get your first virtual machine up and running in just 3 easy steps:

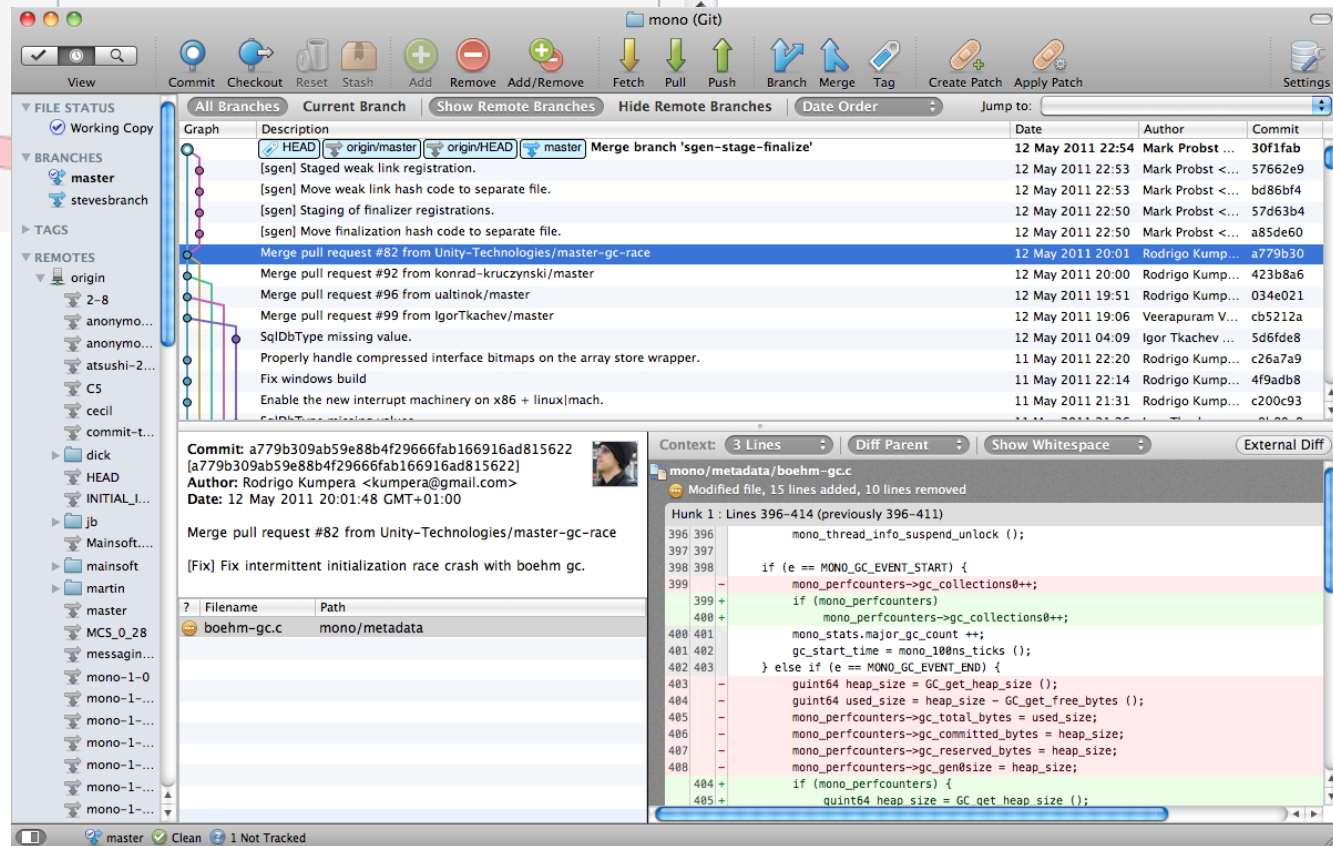
```
$ vagrant box add base http://files.vagrantup.com/lucid32.box  
$ vagrant init  
$ vagrant up
```

Ready to move on? Continue to the [getting started guide](#).

# GUIs reveal complexity in different ways.



SmartGit, cross-platform



SourceTree, OSX

\$ \_

# In the Beginning, Was the Command Line

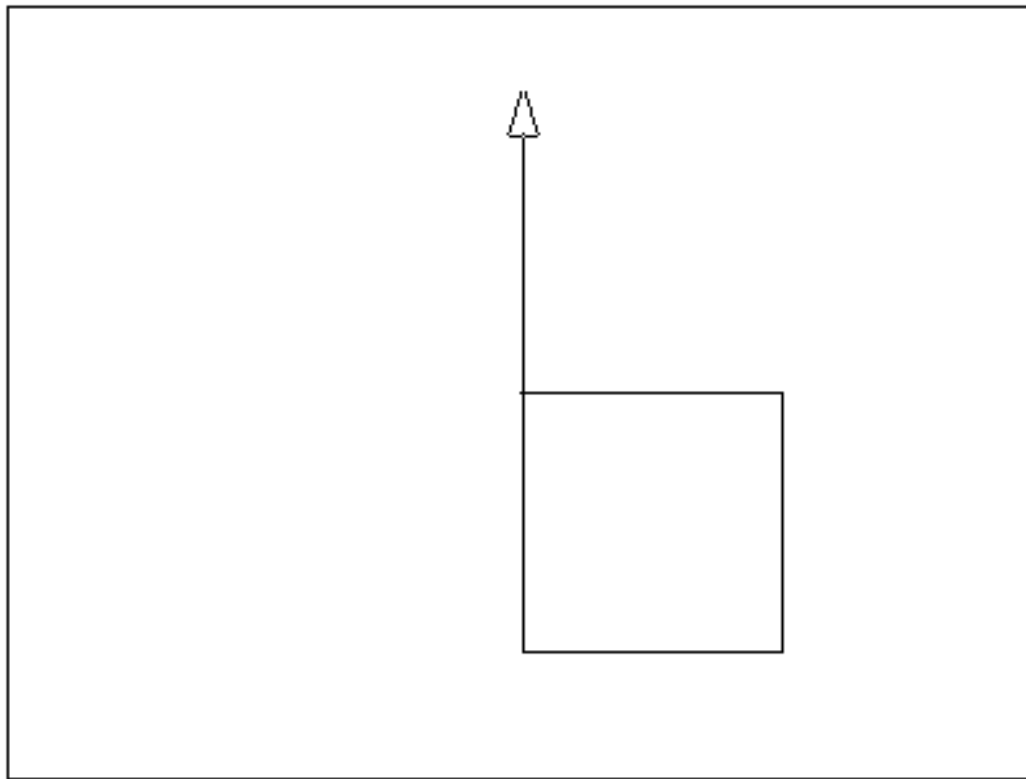
A command-line interface (CLI) is a means of interaction with a computer program where **the user (or client) issues commands to a program in the form of successive lines of text (command lines).**

The command-line interface evolved from a form of dialog once conducted by humans over teleprinter machines, in which human operators remotely exchanged information, usually one line of text at a time.

[http://en.wikipedia.org/wiki/Command-line\\_interface](http://en.wikipedia.org/wiki/Command-line_interface)



# Commands Affect Your Environment



```
Welcome to Berkeley Logo version 5.5
? right 90
? forward 100
? right 90
? forward 100
? right 90
? forward 100
? right 90
? forward 100
? forward 100
? □
```

# Commands Affect Your Environment

```
West of House                                     Score: 0      Moves: 1

ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House
You are standing in an open field west of a white house, with a boarded front
door.
There is a small mailbox here.

>open mailbox
Opening the small mailbox reveals a leaflet.

>
```

# define:prompt

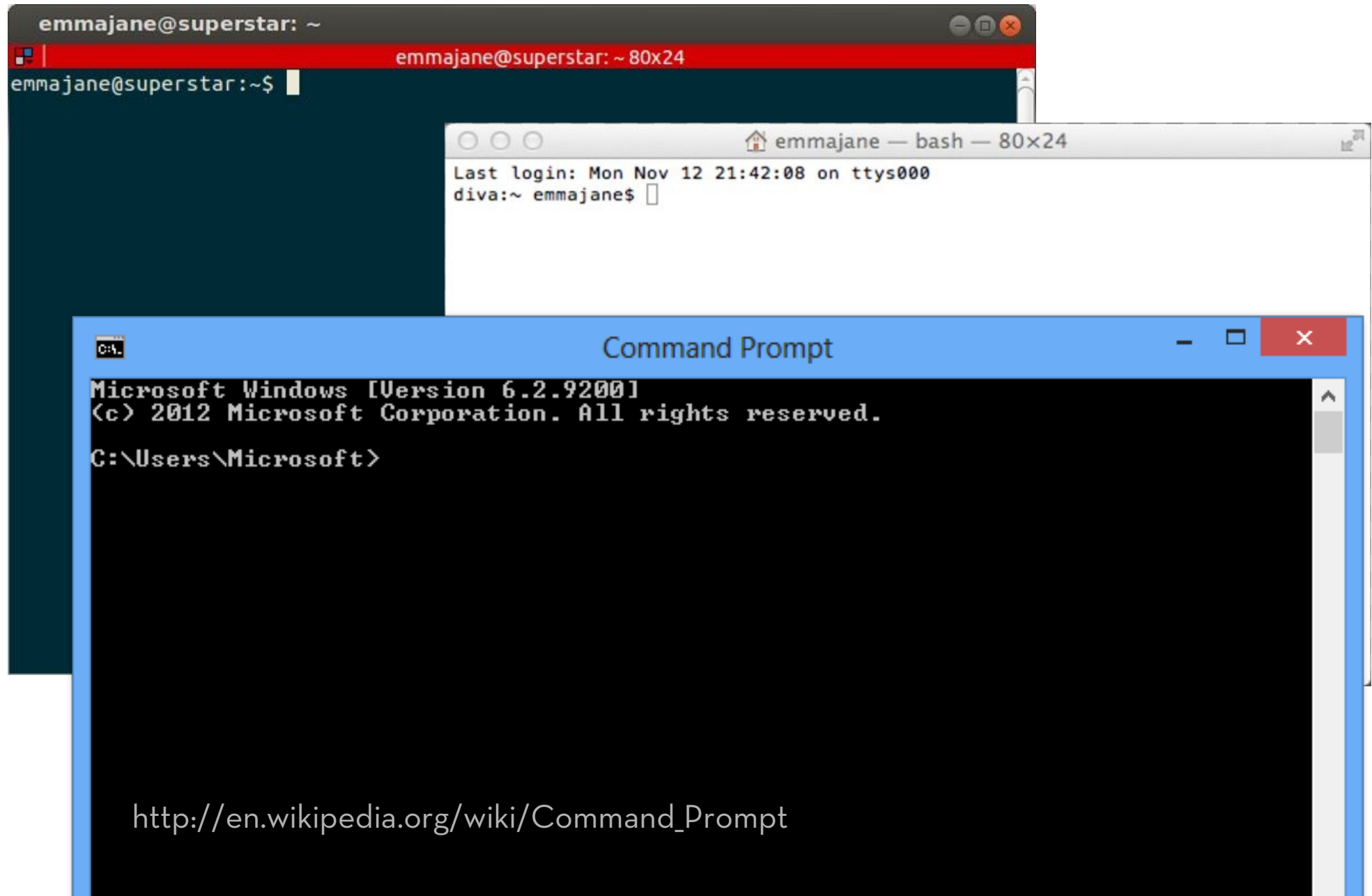
a transitive verb meaning *To move to action.*

? forward 90

> open mailbox

At the command line, you are constantly being asked by the computer, “What action would you like to take next?”

# Command Line Interface



[http://en.wikipedia.org/wiki/Command\\_Prompt](http://en.wikipedia.org/wiki/Command_Prompt)

# \$ bash

- OSX and Linux are offshoots of Unix.
- Both use the GNU Bourne Again Shell (Bash) for their command line environments.
- System commands should work the same (ish) on both systems.
- Windows uses an executable program, COMMAND.EXE. Its commands are different.

# Commands

(verb)

- One-word actions:
  - sing
  - jump
  - look
- One-word commands:
  - help
  - exit
  - history

# Commands With Parameters

(verb noun)

- An action, and a direct object:
  - take book
  - read pamphlet
  - open window
- Commands with parameters:
  - `cd /var/www`
  - `man <command_name>`
  - `mv file_current_name.jpg file_new_name.jpg`

# Commands With Modifiers + Parameters

(verb adjective noun)

- Refining the request:
  - go down stairs
  - take blue pill
  - open left door
- Adding a flag/switch/modifier:
  - `mkdir -p parent_directory/sub-directory`
  - `tar xvf file_to_extract.tar.gz`
  - `ls -lh`



# Ten Commands I Use All the Time

ls	list files
cd	change the current directory
mv	move a file to a new location (and/or rename)
cp	copy a file
pwd	print working directory (answers: where am I?)
rm	remove file
clear	refresh the screen
man	read the manual page for a specific command
tar	work with packages of files
chmod	change the “mode” (aka permissions) of a file or directory

<http://www.commandlinefu.com/commands/browse>

Google linux commands Emma Jane Hogbin + Share

About 2,550,000 results (0.42 seconds)



Related searches: [linux wallpaper](#) [linux desktop](#)

[illegible]

The screenshot shows a terminal window titled "root@kali: ~" with a prompt "root@kali:~#". The user has entered the command "df -h". The output shows the disk usage for the root filesystem (/) as 100% used (1.8G / 1.8G). Below this, the user has entered the command "df -hT", which displays a table of disk usage for all mounted filesystems.

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/sda1	ext4	1000000000	1000000000	0	100%	/
/dev/sda2	ext4	1000000000	1000000000	0	100%	/boot
/dev/sda3	ext4	1000000000	1000000000	0	100%	/var
/dev/sda4	ext4	1000000000	1000000000	0	100%	/usr
/dev/sda5	ext4	1000000000	1000000000	0	100%	/tmp
/dev/sda6	ext4	1000000000	1000000000	0	100%	/home

[illegible][illegible][illegible]

```

#
# Copyright 2010-2012, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 2681, 2682, 2683, 2684, 2685, 2686, 2687, 2688, 2689, 2690, 2691, 2692, 269
```

[illegible]

Conversations with the World: Jews, spirituality, mysticism and their culture

```

An A-Z index of the Linux lsattr command line

alias      Create an alias
cat        View and replace text, database port/passwords/info
cd         Exit from a long
cat        Run a shell builtin
ls         Display a calendar
date       Conditionally perform a command
ls         Display the contents of a file or
ls         Change directory
ls         Partition file and manipulate for lsattr
ls         Change group ownership
ls         Change access permissions
ls         Change file mode and group
ls         Run a command with a different root directory
ls         List C/C++ checkmark and byte counts
ls         User localized error
ls         Compare two files
ls         Compare two files line by line
ls         Run a command - ignoring shell functions
ls         Remove the start characters of a loop
ls         Copy use of new files to another location
ls         Because to execute scheduled commands
ls         Schedule a task to run at a later time
ls         Split a file into content/determined places
ls         Divide a file into parallel parts

```

[illegible]

### All results

# Learning the CLI tools makes your knowledge portable.

- OSX and Linux both use the same flavour of command line environment (“the shell”).
- Not all Web server applications have a graphical administrative interface, so learning to use the command line allows you to run server applications in your local developer environment.
- Sequences of commands can be captured as scripts, which you can run on any compatible machine without having to point-and-click.

# Homework

Using only the command line, complete the following tasks:

- Find the file *node.module* in Drupal core's module directory.

You will need to use the commands *ls* and *cd*.

- Download and unpack the module Backup and Migrate.

You will need to use the commands *wget* and *tar*.

# Standardizing the Developer Environment

- Drupal is deployed *usually* on a Linux server.
- Create a test environment for yourself that matches your “live” environment.
  - Acquia's Developer Desktop
  - WebEnabled
  - VirtualBox
  - Vagrant