General Assembly

Command Line JS

- Rules reminder
- Thanks for Exit tickets
- [joke]: while prepping class, Reading book on anti-gravity; couldn't put it down!

JS1

Objectives

- The Internet & WWW
- The command line
- Client-Server models
- Git & GitHub
- Javascript on the command line

JS1

Installfest Recap

- [Ask class - 4 fingers] software installed:
    - Slack / Git / Node / IDE
- [Ask class] What are they for?
- [Ask class] New skill learned
    - In 2nd half of class
    - "Psuedocode"
    - What's it for?

JS1

Command Line JS

- Today's lesson

Command Line JS

JS & Web Tech

JS & Web Tech

Terminology

- Web Development
  - Frontend Development
  - Backend Development

- [Ask class]: Analogy for FE vs BE?
- Analogy: Clothing store
    - HTML: Structure / layout of store
    - CSS: Lighting & Styling / window
    - JS: Interaction: Try it on / purchase
    - Backend:
        - Shipping logistics
        - Rostering
        - Finances with Banks
        - Purchase history, etc

JS & Web Tech

Terminology

- Web Development
  - Frontend Development
  - Backend Development

- Focus on Frontend
  - With some Backend thanks to node (later in lesson)

JS & Web Tech

Javascript

- Pros
    - Interactivity
    - Few steps to run
    - Backend & IOT
    - Offline
    - AJAX
    - Progressive Enhancement
- Cons
    - [Ask class]: Who's coded in other languages?
    - Don't know environment
        - CPU
        - Memory
        - Features
        - Turned off (security / mobile)

JS & Web Tech

Javascript

Node.js

- Enabled JS out of browser
- Less context switching
  - Anecdote: (JS, PHP, SQL, Bash)
- Async
  - Eg; Paperboy stopping to wait for person
    - Aka; "*Blocking*"
- When would you use node?
  - When you need a backend
  - When you're writing scripts
  - When you're testing J

JS & Web Tech

Javascript

Node.js

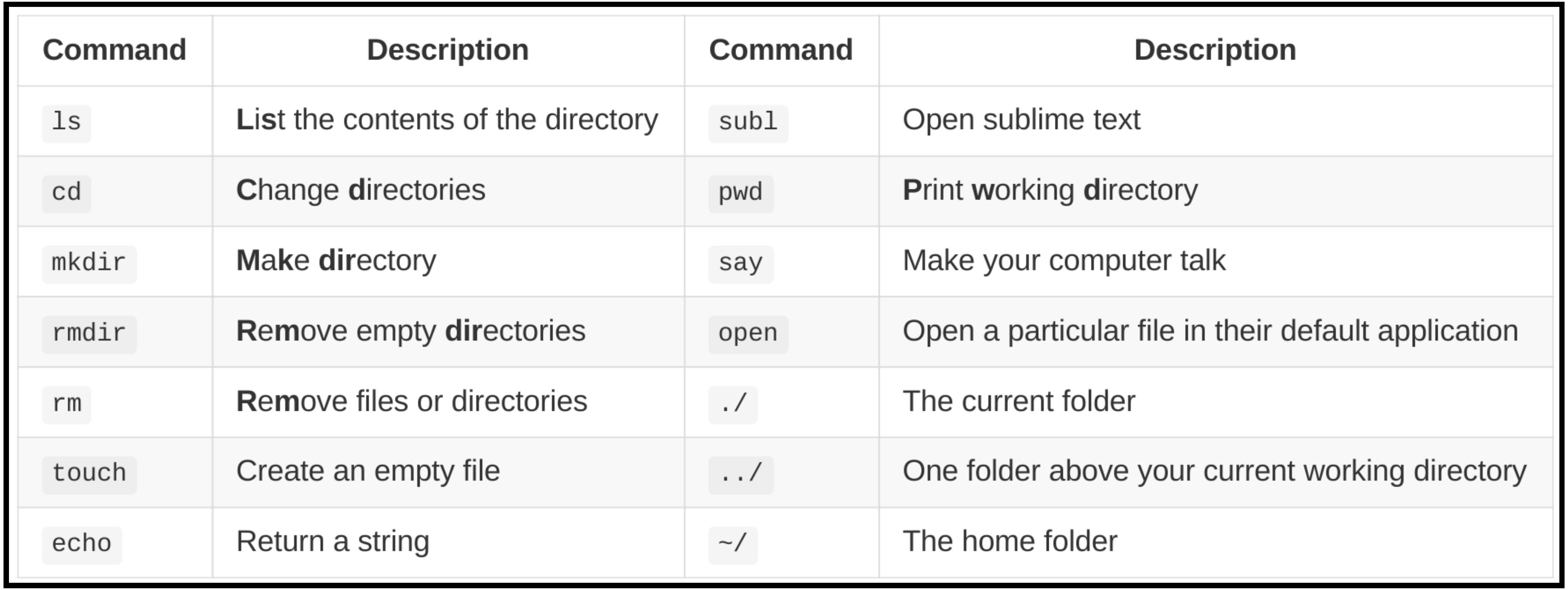
Frameworks & Libraries

- Frameworks exist (Angular, React, etc)
- We wont use them
- Fundamentals first (better!)
- *Will* use libraries (jQuery / Handlebars)
- Learning quick is important
  - Frameworks are born & die every day
- Any Q's before we move on?

Command Line JS

The Command Line

- Not only for JS
- Ran commands last lesson
- [Ask class]: Anyone familiar with command line?
  - Likes / dislikes?
- Better/faster than GUI
- Aka: Terminal, Bash, Shell, Command Prompt, "Running Commands"
  - All approx the same



Command	Description	Command	Description
`ls`	**L**i**s**t the contents of the directory	`subl`	Open sublime text
`cd`	**C**hange **d**irectories	`pwd`	**P**rint **w**orking **d**irectory
`mkdir`	**M**a**k**e **dir**ectory	`say`	Make your computer talk
`rmdir`	**R**e**m**ove empty **dir**ectories	`open`	Open a particular file in their default application
`rm`	**R**e**m**ove files or directories	`./`	The current folder
`touch`	Create an empty file	`../`	One folder above your current working directory
`echo`	Return a string	`~/`	The home folder

http://bit.ly/intro-to-commands

- Share in slack
- Walk through what each does
- Great power comes great responsibility
    - [Anecdote]: Accidentally deleted so many folders in my time
    - Think carefuly before running any destructive commands
    - No undo. It's all gone
- Codealong
    - Win: Git Bash
    - OSX / Linux: Terminal
    - `cd`, `mkdir`, `cd`, `touch`, `subl`, `mkdir`, `rmdir`
    - Compare to GUI often

The Command Line

Practice

http://bit.ly/command-line-practice

- Share link in Slack
- Hand up when done
- Questions?

Command Line JS

Internet

+

World Wide Web

- Internet is infrastructure
- request/response
- No owner, only technical rules
- WWW is HTML documents
- HTTP extends request/response
- 65+ billion web pages
- [Ask class] Other than HTTP? (FTP, BitTorrent, gaming online)

Command Line JS

World Wide Web

Client + Server

HTTP

DNS

- [on whiteboard]
- Client + Server = Request / response
- HTTP - "rules" of request / response
- DNS to find address
    - URL's like saying "near McD's", but what if McD's moves?
    - Special: 127.0.0.1

GILDAN
ULTRA
There's no place
like 127.Ø.Ø.1

- Example of a DNS
- Other example: 192.168.0.1

Command Line JS

World Wide Web

Client + Server

HTTP

DNS

- Quick recap on steps

Command Line JS

Git

- [3 fingers] familiarity (1 = not at all)
- Analogy; bathroom cleaning logbook.
    - who / what / when
    - Different people
    - same bathroom (codebase).

Command Line JS

Git

Magic Logbook for your code

- Can store every change ever made
- Can go back in time
- Can even rewrite history
    - [next slide] / Magic!

- But don't worry! We'll only use a few commands
- [Think Pair Share]: What are the steps to add new logbook entry for, say, the toilet example?
    - [on board]: Write steps
    - [on board]: Write equivalent in git
- [Codealong]:
    - First, run: `history -c` to clear history
    - In 'goals' dir
    - mkdir / init / status / add / status / commit / status / log
    - Write terminology too: *Repository*, *Commit*, *Branch*
        - Commits are cheap! Do loads, tiny changes.
        - Branches are cheap!
        - `master` branch default

Command Line JS

Git

Magic Logbook for your code

- But don't worry! We'll only use a few commands
- [Think Pair Share]: What are the steps to add new logbook entry for, say, the toilet example?
  - [on board]: Write steps
  - [on board]: Write equivalent in git
- [Codealong]:
  - First, run: `history -c` to clear history
  - In 'goals' dir
  - mkdir / init / status / add / status / commit / status / log
  - Write terminology too: *Repository*, *Commit*, *Branch*
    - Commits are cheap! Do loads, tiny changes.
    - Branches are cheap!
    - `master` branch default

Break

Git

Sharing Code

- How team shares?

Git

Sharing Code

```
git clone
```

- Using `clone`
  - Get a full copy of the log book
  - Not just a photocopy. A copy of the entire bathroom too!
  - alternative to `init`

Git

Sharing Code

```
git clone git://192.168.20.12/js1
```

- How team shares?
- [on laptop] run `~/dev/GA/shared-repos/start-daemon.sh`
- Class clone + explore
  - `ls`
  - `git status`
  - `git log`
- Show commands with: `history | grep git`
- Cumbersome :(
- Easier...

Command Line JS

GitHub

- Somewhere to put repositories
- Like Dropbox / Google Drive / etc
  - Facilitates sharing & storing (code)
  - Also allows reviewing + merging other's code
- Create account: http://github.com
- Setup git: https://help.github.com/articles/set-up-git/#setting-up-git
- Cache github password: https://help.github.com/articles/caching-your-github-password-in-git/

Command Line JS

GitHub

Create a repo on GitHub

- [Show class] New repo
  - Name it `<username>.github.io`
  - Click "Create repository"
  - "push an existing repository"
  - Hand up when done

Command Line JS

GitHub

Create a repo on GitHub

```
git push origin master
```

Explain:

- push vs pull (aka: clone)
- origin == github
- master == branch

Command Line JS

GitHub

http://<username>.github.io

- The page that was just pushed!
- Edit one of the pages in GitHub
    - Commit
    - Refresh
    - Note changes
- Check local; no changes
- `git pull` to get changes
- Check local; changes!
- And do reverse; local edit, then `git push`

Command Line JS

GitHub

- Edit on GitHub, then

```
git pull
```

- Edit locally, then

```
git push origin master
```

- Your turn

Command Line JS

Javascript on the command line

- JS isn't just in browser. Also in node!
- Installed node last lesson
- The good stuff!
- Uses Command Line
  - Swiss Army Knife
  - Been around forever
- Right of passage: Hello World
  - 1972, Brian Kernighan (author of the C language, which JS is based on)
- `node` to start
  - `console.log('Hello world')`
    - note the '`undefined`' means no further results (aka; a *return* value, learn that in later lesson)
  - `var x = 5`
  - `var y = 2`
  - console.log(x)
    - no quotes? Why?
  - x + y
  - y * 4
  - x = 9
  - x + y
  - Play for 5 minutes, see what you can discover
- Ctrl-c-c to exit
- If time:
  - https://www.codeschool.com/courses/try-git

JS1

Objectives

- Revisit each of the objectives on board

JS1

Next Lesson

- Data Types
- Variables
- Arrays & Iteration

JS1

Questions?

- Further learning:
    - https://www.codeschool.com/courses/try-git
    - https://egghead.io/lessons/node-js-using-the-node-js-repl-shell

JS1

Exit Tickets

http://ga.co/js1syd

- [share in Slack]

General Assembly

JS1