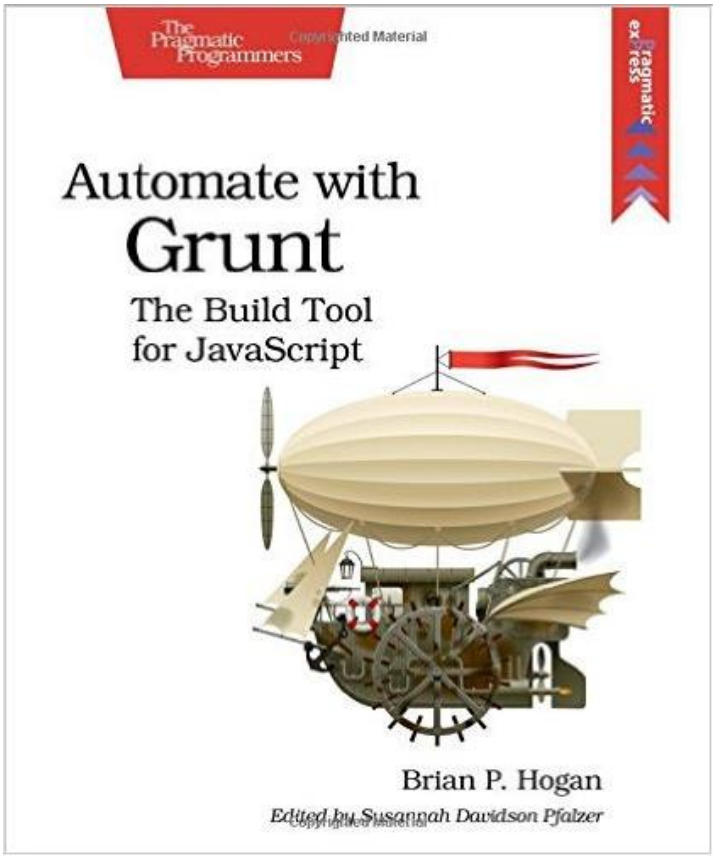
JavaScript Tooling

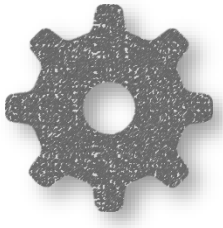
transpilers,
deployment,
package
managers,
configuration
management

Book

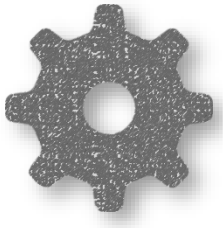
- Automate with Grunt: The Build Tool for JavaScript by Brian P. Hogan



Top tools of 2015



- http://stackshare.io/posts/top-50-developer-tools-and-services-of-2015
  - Bootstrap
  - jQuery
  - JavaScript
  - Git
  - Node.js
  - nginx
  - AngularJS
  - MySQL

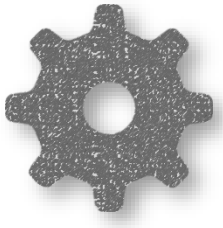


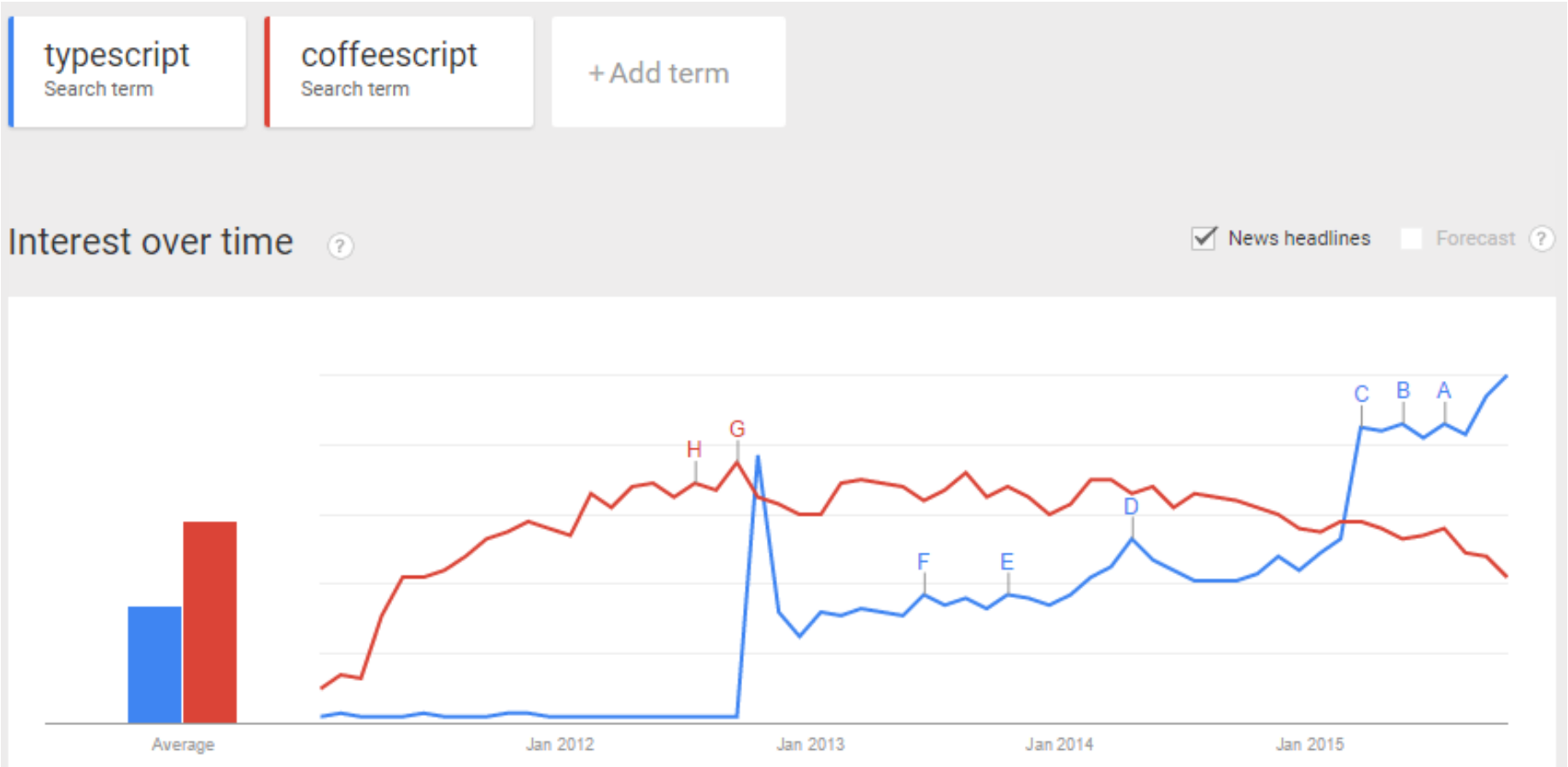
“you can write large programs in JavaScript. You just can’t maintain them”

- **Anders Hejlsberg**

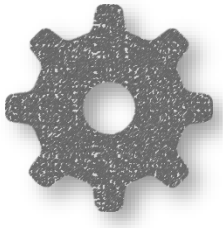
JavaScript Transpilers

JS transpilers





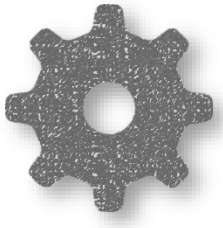
TypeScript



- http://www.typescriptlang.org/
- Key features:
  - Code encapsulation
  - Maintainable code
  - Tooling support
- Base language for Google Angular2!

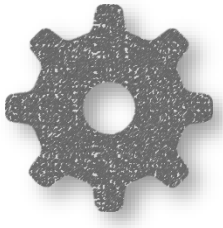
TypeScript

Other JS transpilers

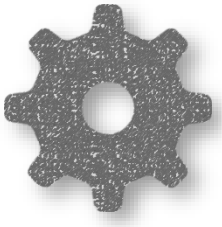


- CoffeeScript
  - http://coffeescript.org/
  - popular with Ruby people, used by Dropbox
- Babel.js
  - https://babeljs.io/
  - just beginning to catch on, ES6 use
- Traceur.js

Editor support

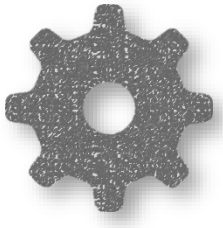


- Usually a plug-in
- Visual Studio Code
- Jetbrains WebStorm built in
  - File Watchers



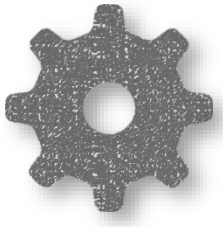
Package managers

About package managers



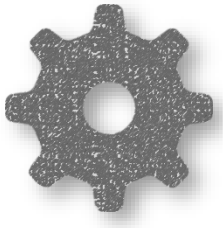
- Scripted download and installs
- Types
  - system level - Windows installer, nuget, mobile app stores
  - application level
- Features
  - Must handle shared libraries of code
  - Must retain local configurations
  - Must work with manual installs
  - Must allow upgrade suppression

Chocolatey



- https://chocolatey.org/
- Windows only
- Commands
  - Search - choco search something
  - List - choco list -lo
  - Install - choco install baretail
  - Update - choco update baretail
  - Uninstall - choco uninstall baretail

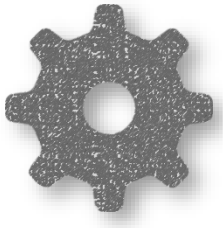
npm



- https://github.com/npm/npm
- Node Package Manager
- Installed with node.js
  - Also a msi

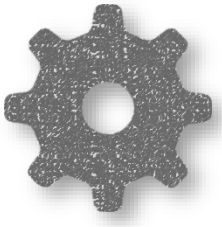
TRIAL

npm



- **module** - anything loaded with require( ) in a node.js program, a folder with index.js, any JavaScript file.
- **package** - usually a module, but not always
  - package.json - defines the package with names, versions…
    - an inventory list

npm

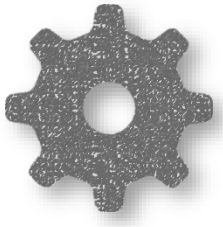


- install a package
  - programmatic use - npm install <package>
  - command line use (global) - npm install -g <package>
- show packages
  - in current directory: npm ls
  - in global directory: npm ls -g

Why does npm hate me?

npm is not capable of hatred. It loves everyone, especially you.

npmjs



- https://www.npmjs.com/
- Listings of node packaged modules for
- Search packages - by name, most depended on, most starred

lodash
The modern build of lodash modular utilities.
3.9.3 published 4 days ago by jdalton

async
Higher-order functions and common patterns for as…
1.0.0 published a week ago by aearly

underscore
JavaScript's functional programming helper library.
1.8.3 published 2 months ago by jashkenas

request
Simplified HTTP request client.
2.56.0 published yesterday by simov

commander
the complete solution for node.js command-line pro…
2.8.1 published a month ago by zhiyelee

express
Fast, unopinionated, minimalist web framework
4.12.4 published 2 weeks ago by dougwilson

debug
small debugging utility
2.2.0 published 3 weeks ago by tootallnate

chalk
Terminal string styling done right. Much color.
1.0.0 published 3 months ago by sindresorhus

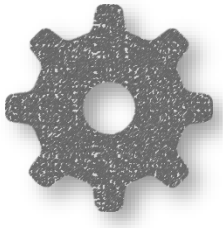
q
A library for promises (CommonJS/Promises/A,B,D)
1.4.1 published 2 weeks ago by kriskowal

colors
get colors in your node.js console
1.1.0 published a month ago by marak

mkdirp
Recursively mkdir, like `mkdir -p`
0.5.1 published 2 weeks ago by substack

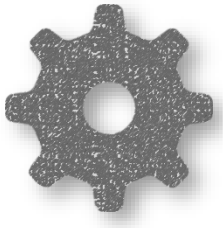
coffee-script
Unfancy JavaScript
1.9.3 published 2 days ago by jashkenas

npmcdn



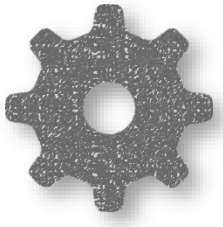
- https://npmcdn.com/
- a CDN for packages published by npm
- https://npmcdn.com/package@version/path/to/file is the format

npm-stat



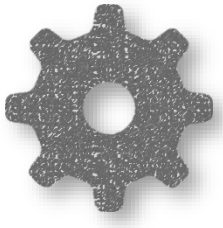
- https://npm-stat.com/charts.html

package.json



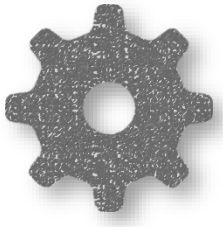
- **package.json** file
  - node's package descriptor
  - stores metadata about the module (name, version, author…)
  - lets you reinstall any dependencies or publish to npm without them
- Creating
  - use **npm init, npm install**
  - use **grunt-init gruntfile**, creates **Gruntfile.js** also
  - manually

node_modules



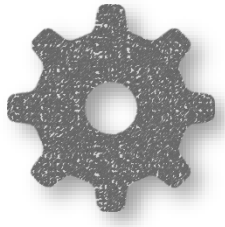
- Files installed with npm install are put in node_modules
- Delete folder and reinstall to fix some problems
  - Folders are often nested so deep Windows can't delete them
  - Install rimraf with npm and then delete with rimraf <dir>

Twitter Bower



- http://bower.io/
- client-side package management
  - define, version, and retrieve your dependencies
  - creates a JSON bower.json which installs all dependencies
    - great for distros
    - gets yours or standard packages from Git
- install a package (AngularJS)
  - **`bower install angular`**
- depends on node, npm, git
  - Windows requires installing msysgit
    - use option: Run Git from the Windows Command Prompt

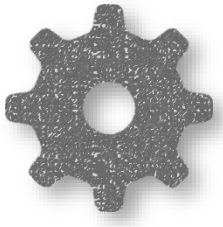
Yeoman



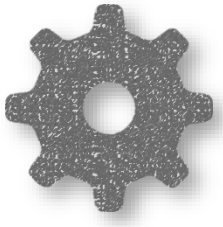
- http://yeoman.io/
- Yo, Grunt and Bower
- install with npm
  - npm install -g yo
- scaffolding, build, and package management
- uses 'generators' as scripts.
- Addy Osmani - Google, TodoMVC

YEOMAN

Yarn



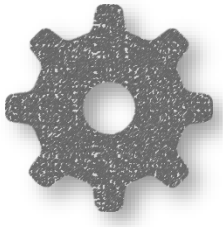
- https://yarnpkg.com/en/
- A replacement for npm that is better for
  - Scaled apps
  - Continuous integration
  - Reliable and repeatable package structures



Creating CSS from code and more

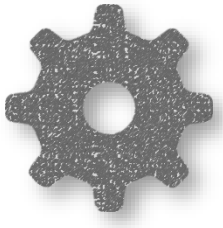
CSS Transpilers

CSS preprocessors



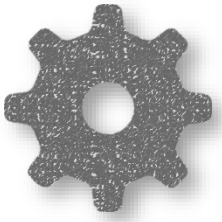
- CSS is not well managed at around 200+ lines
- Programming structures add maintainability
  - reusable values in variables
  - reusable modules (gradients, rounded corners, etc.)
- Automatic boilerplate code for vendor prefixes
  - -moz, -webkit, -o, -ms
  - animation, transition, transform, box-shadow, border-radius
- Calculated values
- Nested CSS
  - Results in minimized CSS code

CSS preprocessors

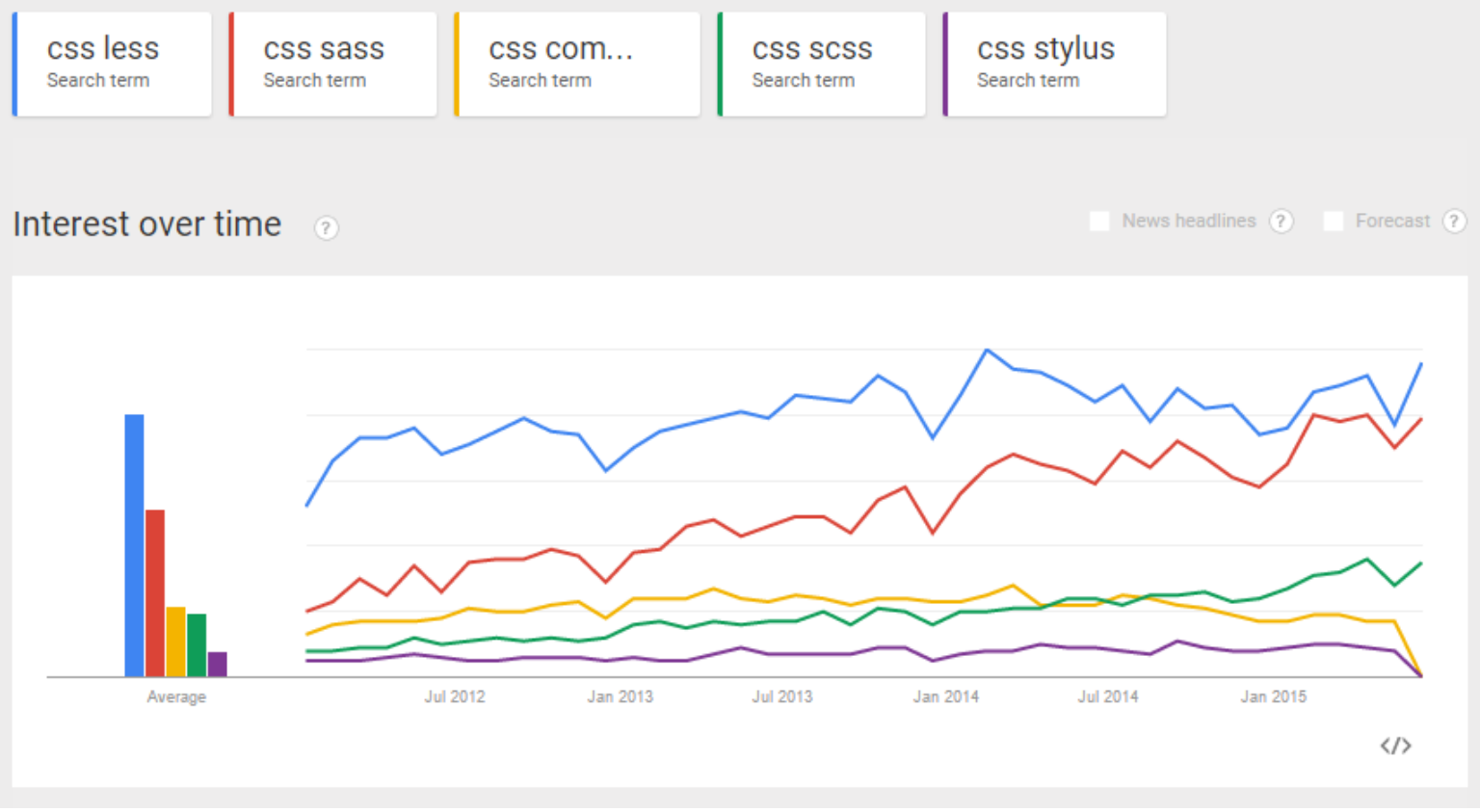


- Cons
  - features becoming unnecessary
    - browsers stopping the use of prefixing
    - CSS variables supported by Firefox and others most likely soon

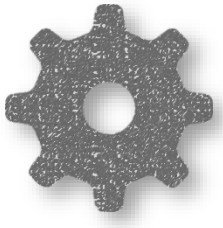
CSS preprocessors



- JavaScript based - LESS, Stylus
- Ruby based - SASS, Compass, SCSS

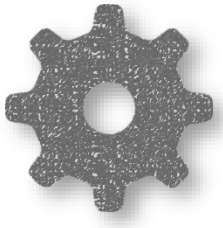


Ruby based



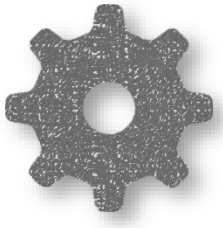
- adds programming syntax to CSS
  - variables
  - nesting…
  - mixins (libraries)
- Same engine, different syntax
  - SASS is like Python with indentation.
  - SCSS is like Java/C# with semicolons, braces
  - Compass - http://compass-style.org/
- Ruby dependency
  - requires executable to run by JS

SASS



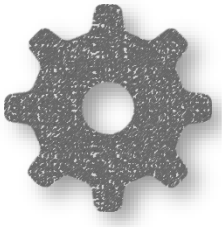
- SASS/SCSS
  - http://sass-lang.com/
- SASS & Compass extensions
  - http://www.sache.in/
- JavaScript versions
  - http://sass-lang.com/libsass
  - https://github.com/medialize/sass.js

JavaScript based

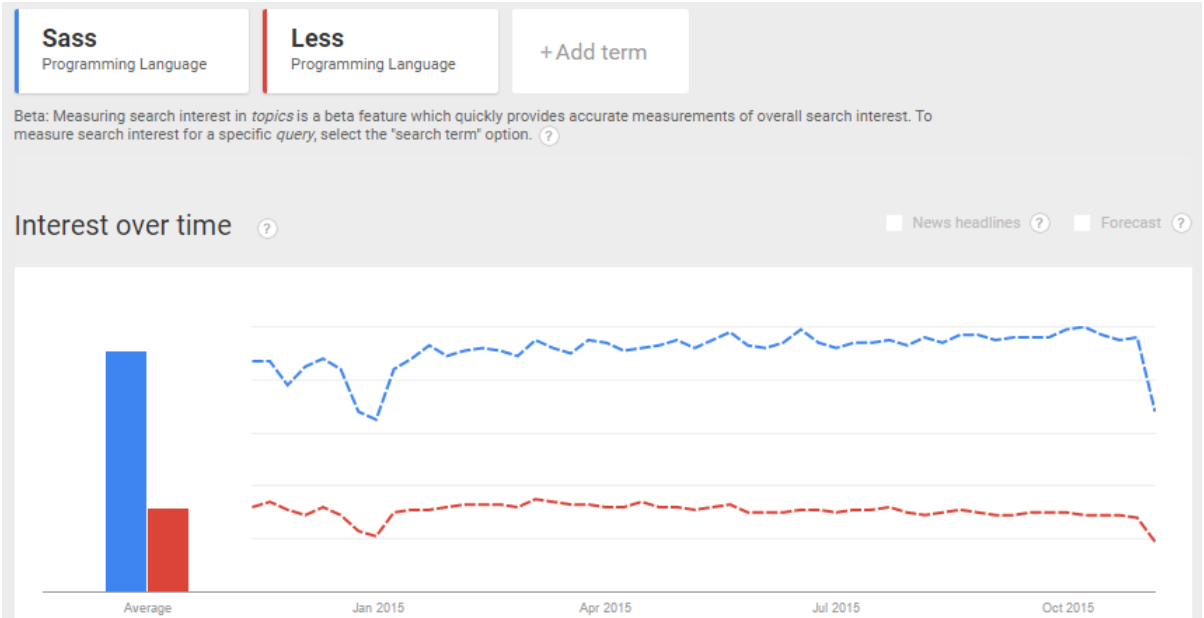


- Declarative (in text file)
  - LESS - http://lesscss.org/
  - Stylus - http://learnboost.github.io/stylus/
- Imperative (in code)
  - AbsurdJS
  - restyle( )

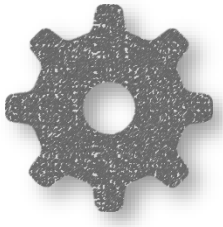
LESS



- Same functionality as SASS
- JavaScript based
- Libraries aren’t as popular as SASS

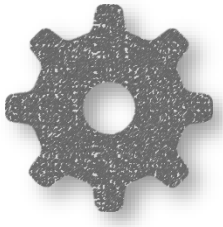


LESS



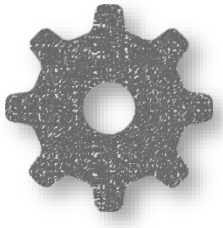
- Sublime packages
  - https://sublime.wbond.net/packages/LESS
  - https://packagecontrol.io/packages/lessc
- Workflow
  - http://www.webmaster-source.com/2013/08/28/setting-up-a-less-workflow-in-sublime-text/
- LESS mixin libraries
  - LESS HAT - http://lesshat.madebysource.com/ (86)
  - LESS elements - http://lesselements.com/ (17)
  - Preboot - http://getpreboot.com/
  - Clearless - http://clearleft.github.io/clearless/

PostCSS



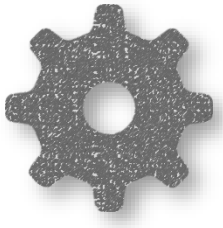
- https://github.com/postcss/postcss
- CSS parser/processor
  - not pre- or post-
  - think XML parser with code
  - part of your build flow in Gulp/Grunt/webpack/Express…
- Does everything SASS/LESS does and more but faster.

Minification



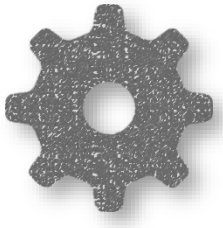
- compression
- removes all unnecessary characters
  - spaces, new lines,
  - comments
- does not affect the functionality of the source code
- .NET
  - http://www.asp.net/mvc/overview/performance/bundling-and-minification

Editor support



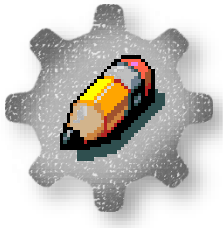
- Usually a plug-in
- WebStorm built in
  - File Watchers

Twitter Recess

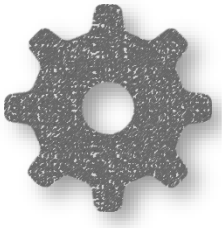


- http://twitter.github.io/recess/
- code quality tool for CSS built on top of LESS
- linter, transpiler
  - normalize whitespace, strip units from 0 values, reorder your properties

Exercises



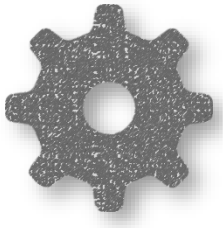
- Set up JavaScript environment
- Set up Microsoft Visual Studio Code
- LESS



To run a set of tasks

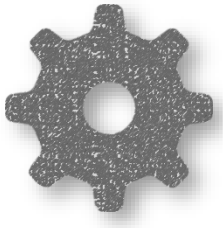
Deployment

Deployment terms



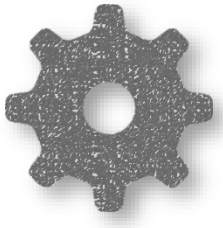
- **Minification**
  - reduces file size through several strategies
- **Continuous integration**
  - code, test, immediate build and integrate
  - merging all developer working copies into a shared mainline several times a day
- **Continuous delivery / deployment**
  - same as CI but after merge is immediately deployable.
- QA in production – skip the CD/D part and do later

Continuous integration / delivery



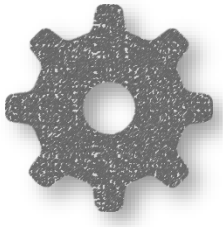
- Travis CI - https://travis-ci.org/
  - a hosted, distributed continuous integration service used to build and test projects hosted at GitHub
  - configured by adding a file named .travis.yml
  - automatically detects when a commit has been made and will try to build the project and run tests
- Jenkins - http://jenkins-ci.org/
- SnapCI - https://snap-ci.com/
  - simple cloud CD, Thoughtworks

Task runners / build automation



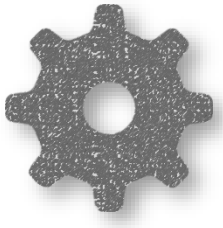
- Automates admin tasks by a DSL
  - compiling, transpiling, continuous integration, zipping, FTP, etc.
- Unix : make
- Ruby: rake
- Java: Ant, Maven, Gradle
- JavaScript: **Grunt**, **Gulp**, Broccoli, Webpack

Grunt



- http://gruntjs.com/ (Mar 2012)
  - minification, compilation, and linting
- uses npm packages (~4% of total)
- Simple JavaScript functions
- never a global install
  - npm install grunt
  - add to npm's package.json with --save-dev
- but command line tool is global
  - npm install grunt-cli -g

Grunt

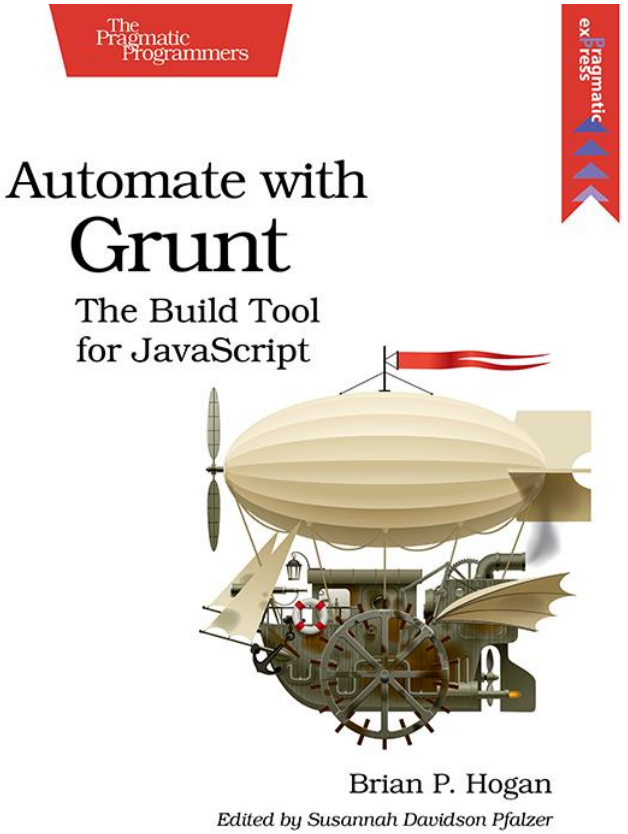


- Gruntfile.js script is in **CommonJS** style
- `// constants & functions`

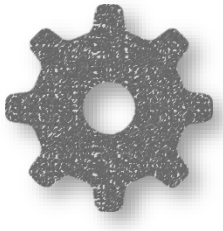
```
module.exports = function (grunt) {
    grunt.initConfig({
        // configuration
    });

// user-defined tasks
}
```

Grunt

- Automate with Grunt: The Build Tool for JavaScript
  - by Brian P. Hogan
  - May 2014
  - Jolt Finalist 2014



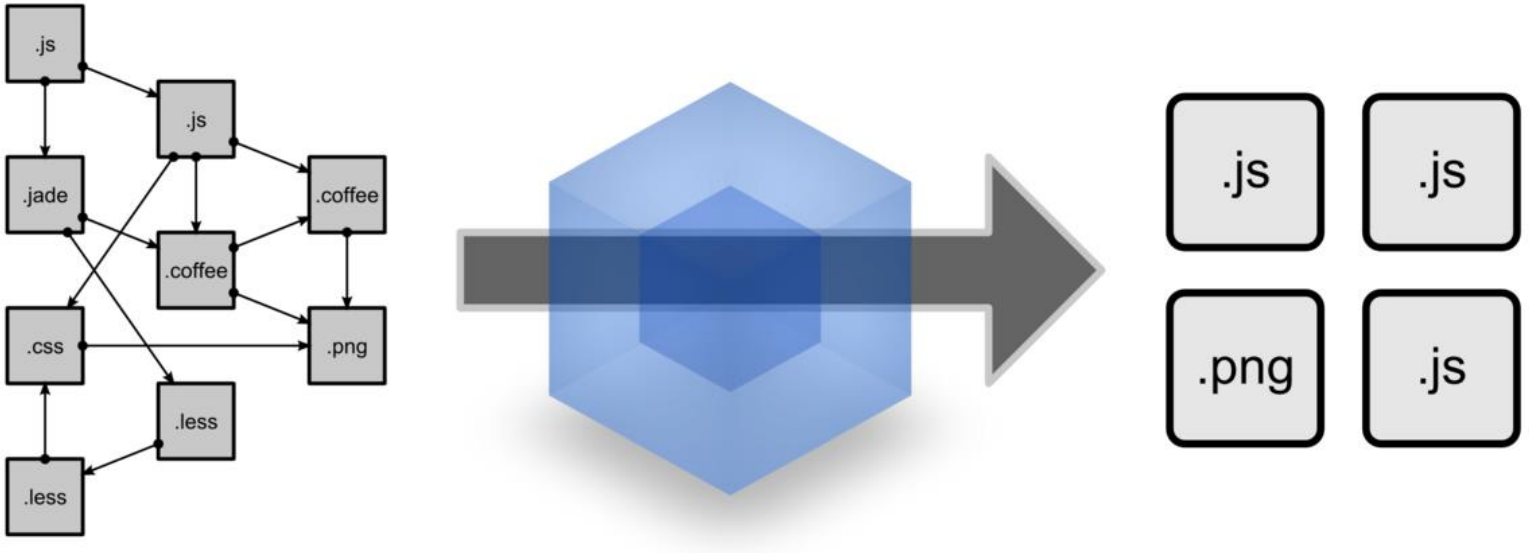
Gulp



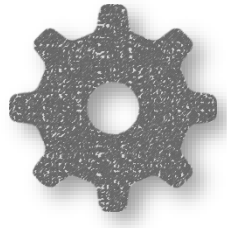
- http://gulpjs.com/
- Cooler than Grunt
  - gaining popularity
  - uses the streaming API of Node.js
- Steeper learning curve
  - most people recommend staying with Grunt after using both for production

Webpack

- http://webpack.github.io/
- module bundler / task runner

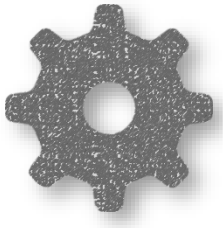


Broccoli



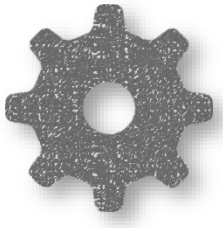
- Used for Angular-CLI
  - based on Ember CLI
  - https://www.solitr.com/blog/2014/02/broccoli-first-release/
    - useful discussion of other build tools

Uglify-JS



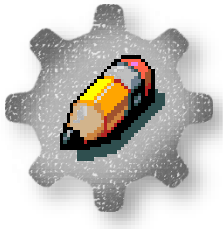
- http://marijnhaverbeke.nl/uglifyjs
- https://github.com/mishoo/UglifyJS
- JavaScript parser / mangler / compressor / beautifier library for NodeJS

eslint

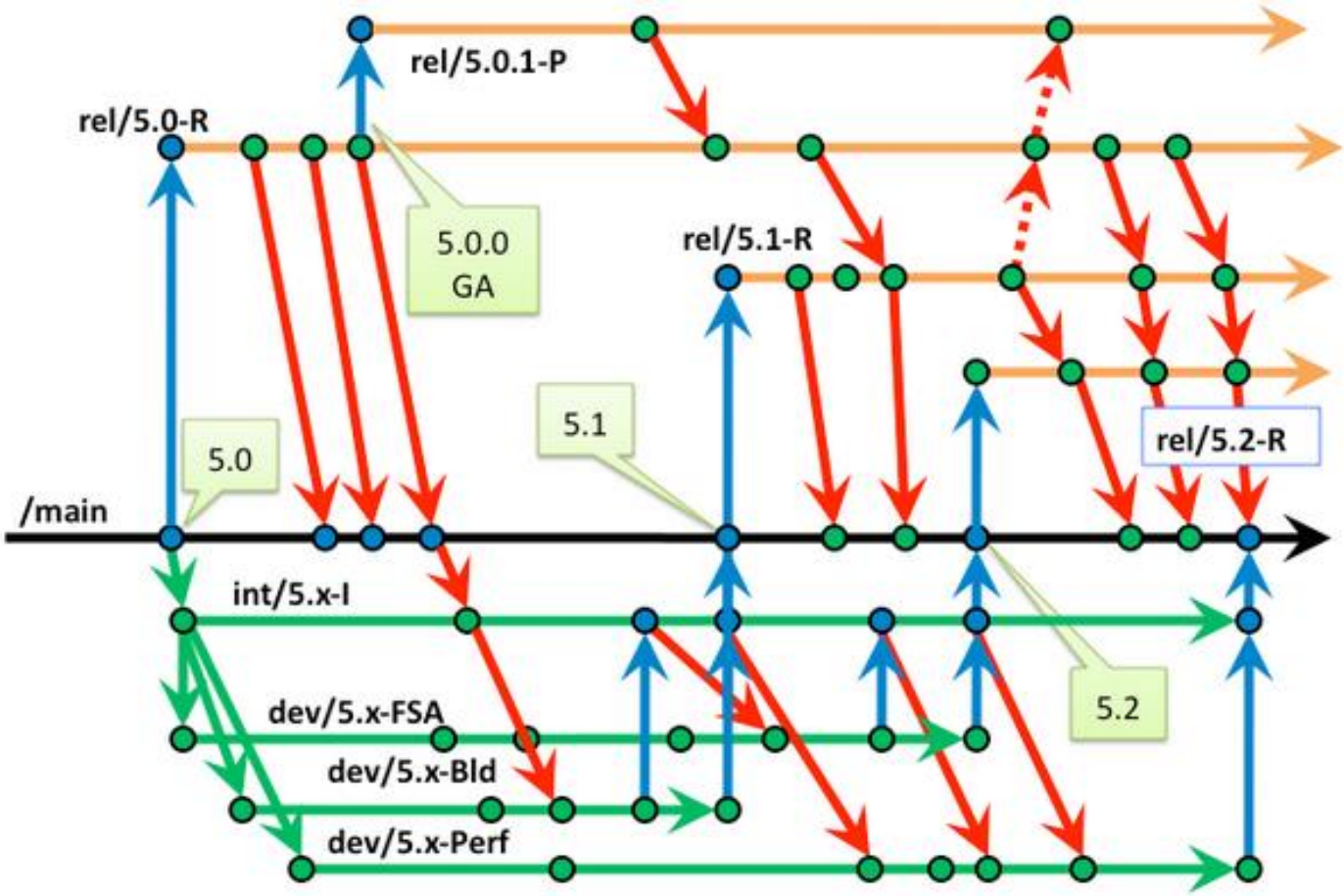


- Custom syntax checking
- **npm install -g eslint**
- install the VS Code eslint extension
- use eslint --init to create an initial eslint configuration by answering questions or by picking a popular configuration.
  - use JSON as the format of the eslint configuration file, then VS Code can provide you with IntelliSense when you edit the .eslintrc.json file.

Deployment exercises

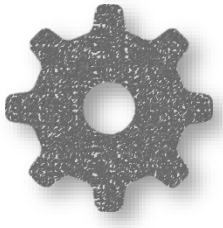


- Grunt - setup and install
- Grunt - test and config
- Grunt - linting with jshint
- Grunt - reinstall exercise modules
- Grunt - minify with Uglify
- VS Code + ESLint
- VS Code + grunt tasks



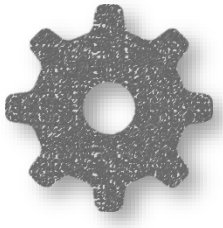
Software Configuration Management

About SCM



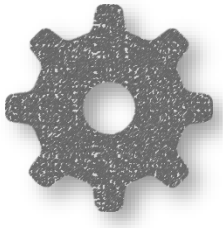
- aka Version Control
- ***Documentation***
  - View decisions made
  - View order in which components were developed
- ***Attribution***
  - Know who did what
  - Know who didn't document well enough
- ***Experimentation***
  - Encourage tests to see if code works and use them when they are successful

VCS vs DVCS



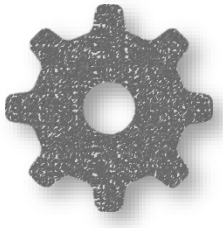
- VCS
  - centralized server
  - no server, no control, no work
- Distributed VCS
  - optional remote server
  - work offline
  - Git

SCM in other languages



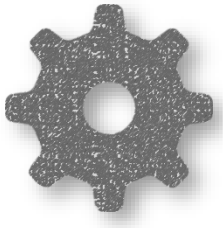
- Java
  - Subversion
- .NET
  - Team Foundation Server
- COBOL
  - CVS
  - Endeavor

SemVer



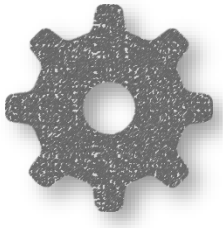
- SemVer (semantic version)
  - MAJOR . MINOR . PATCH
  - Major = when you make the API incompatible
    - a zero major version is not considered stable and in dev
  - Minor = backwards compatible features
  - Patch = backwards compatible bug fixes

Git



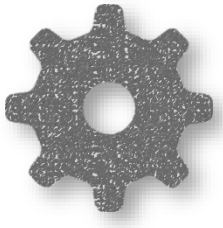
- Git - DVCS (“a stupid content tracker”) created in 2005 by Linus Torvalds to manage Linux
  - language
  - creates local or remote repositories
  - hosts can specialize in Git support

Git - terms



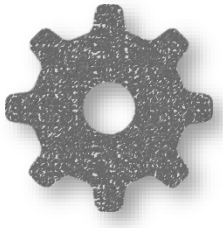
- **repo** - short for repository
  - files under git control
- **fetch** / **pull**
  - **fetch** gets branch info that you don't have without merging it
  - **pull** fetches branch and merges it to current branch
- **tags** - text markers used for version numbers (v1.0)

Git - terms



- **staged** / unstaged changes
  - a queued list of files different from last commit
- **commit** – save working file changes, a snapshot
- **push** - save working files to repo
  - only works if you cloned and no one else pushed

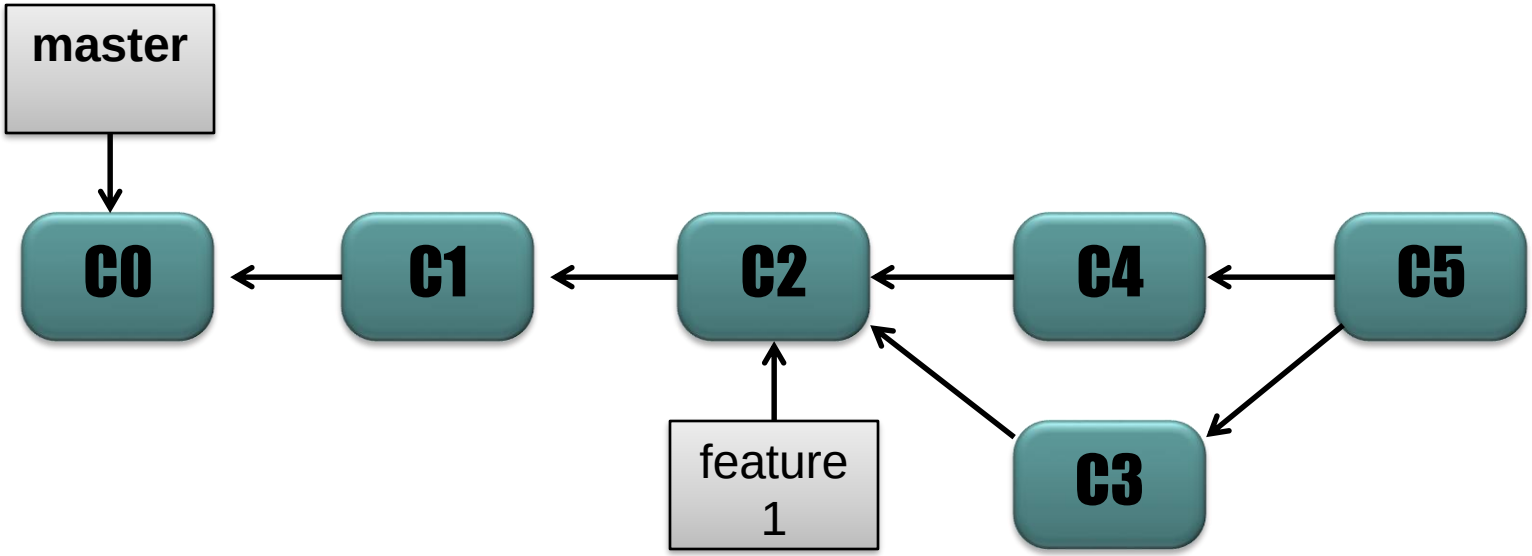
Git - terms



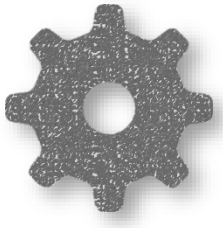
- **branch** - a reference/pointer to a commit
  - **HEAD** is the active branch
  - **checkout** switches view/HEAD to another branch
  - **master** is the default first branch of any repo, the trunk
- **origin** – the default name of your main remote repo
- **clone -** copies ALL file data from origin to another repo, usually your local one

Git - workflow

One person local workflow, not team
Branch by feature is not recommended.
See branch by abstraction.



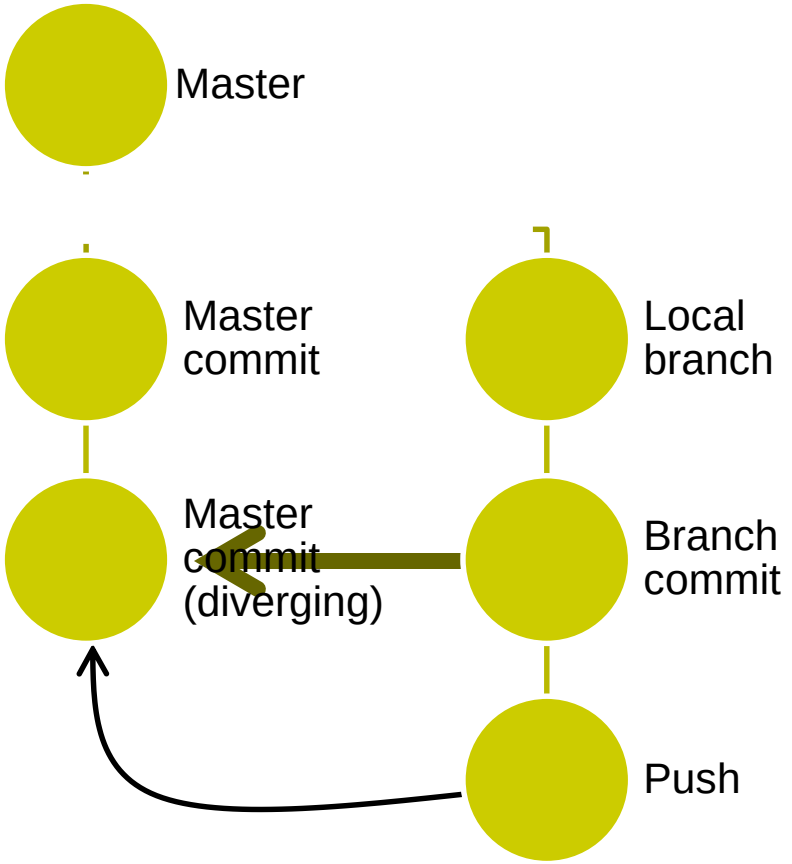
Git - terms



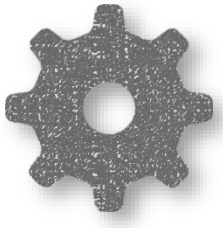
- **merge / rebase**
  - **merge** is 3 way (two children, one common parent) commit
  - **rebase** uses first child changes and adds them to second child, cleaner history only requiring a ff merge, not for team work because it loses commits.

Git - workflow

- Branches **push** back to where they came from
- **Rebase** allows push back to new branch



Git - .gitignore



- suppresses tracking
- .gitignore
  - a file to keep files from being placed in repos
  - place in repo directory
- Example
  - `# ignore packages installed by npm`
  - `node_modules`
  - `# put any other files you don't want to check in here,`
  - `# such as .DS_Store (OSX), *.bak, etc.`

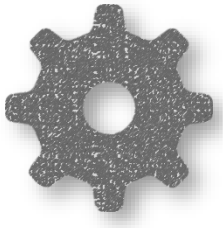
GitHub

- Hosted repositories
- Free accounts are public
- Private accounts
  - 5 repos = $7, 10 repos = $12,
    20 repos = $22, 50 repos = $50


```
git commit
git push
git help push
git pull
git merge
git help reset
git reset --hard
git rebase
git help rebase
git rebase master
git push
git help push
git help remote
git remote add --ta
git push
git push --f
```

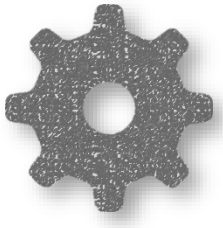
GitHub

Atlassian Bitbucket



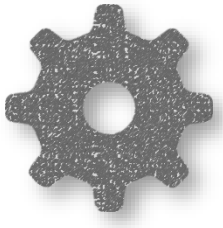
- https://bitbucket.org/
- Unlimited private repos for up to 5 users
- Works with Mercurial also
- Atlassian also makes JIRA, Confluence (Agile team project tools), SourceTree, Stash, Bamboo

Git clients 1



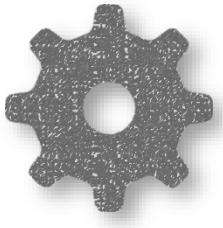
- *Atlassian SourceTree - https://www.sourcetreeapp.com/
  - best overall
  - also Mac OS X
- *GitHub Desktop - https://desktop.github.com/
- Axosoft GitKraken -
  - https://www.gitkraken.com

Git clients 2



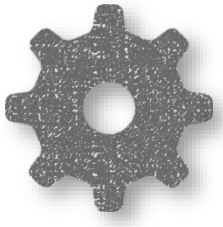
- msysGit - Git for Windows, Git GUI
  - http://msysgit.github.io/
  - http://git-scm.com/documentation
- Visual Studio / VS Code

Git - workflow



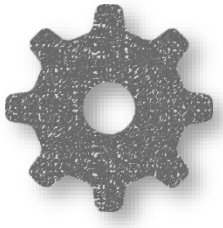
- Start project
  - **initialize** repo (.git folder) in project folder
  - **add** files in project - project has been staged in the index
  - store files permanently - **commit** the files with message

Git - workflow



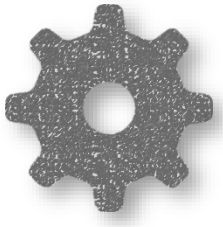
- Local - code, test, pass - **commit**
  - change files and test
  - add updated contents to index
    - or **Rescan** to find unstaged changes and **Stage Changed**
  - commit changes (save)
    - **Sign off** to add name and email to message

Git - workflow

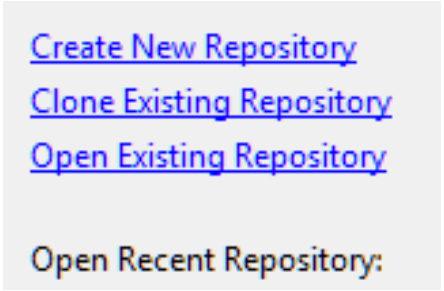


- Local - code, **commit**, test, fail, fix, test, pass, **commit**
  - create a branch and switch to that branch
  - make changes and commit that branch
  - after successful test, merge branch back to master
    - if conflicts exist, markers will be left in the files, edit and recommit merge results.
  - delete branch

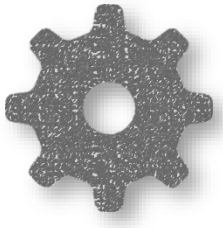
Git - workflow



- Enhancements - work on code from Github or use to start your own
  - fork the remote repo
    - practice a fork at https://github.com/octocat/Spoon-Knife
  - clone Github repo and make changes
    - no directory should exist before using Git Gui
  - notify owner to pull changes
    - owner will fetch changes to preview

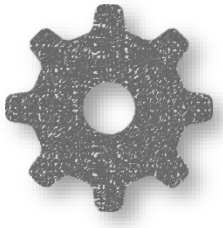


Git - workflow



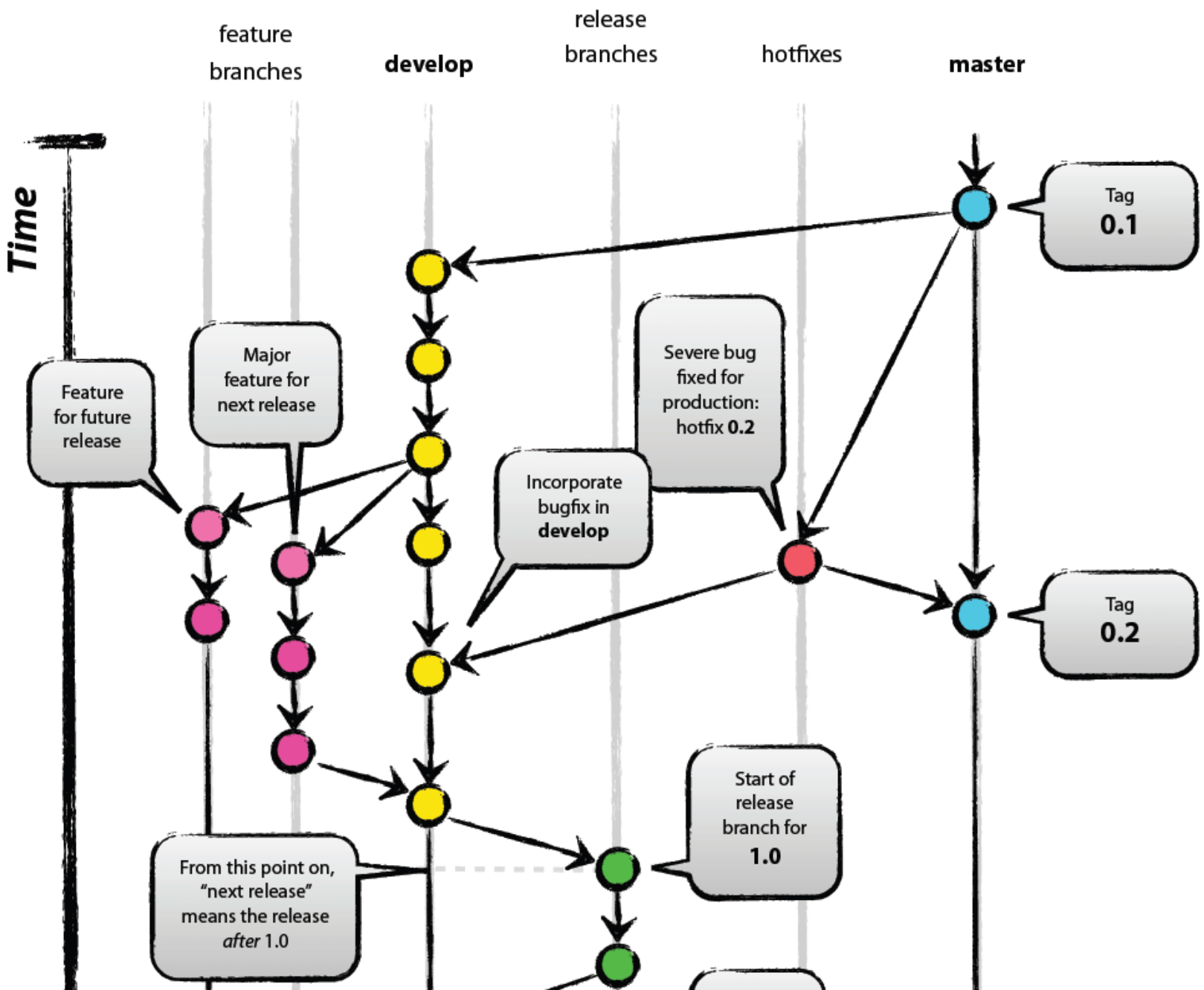
- Remote workflow - push repo to remote host
  - set up account on host, create repo, copy repo URL
  - push selected branch to destination repo host
    - use username / password

Git - workflow

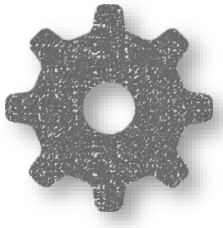


- Team workflow
  - Common branches
    - develop
      - feature / topic - independent scope, unit tested
      - release - packaged for testing and deployment, version assigned
    - hotfix - repairing bugs in the master, merged to develop also

http://nvie.com/files/Git-branching-model.pdf



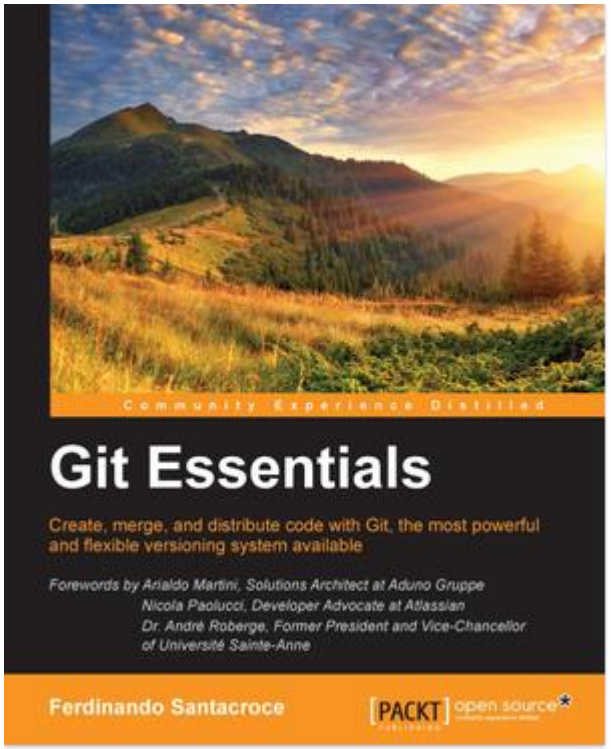
RawGit CDN



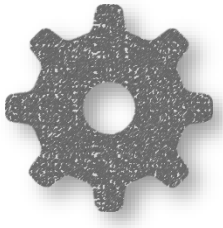
- http://rawgit.com/
- paste in raw URL
  - https://raw.githubusercontent.com/systemjs/systemjs/master/dist/system.js
- get back testing URL
  - https://rawgit.com/systemjs/systemjs/master/dist/system.js
- and production URL
  - https://cdn.rawgit.com/systemjs/systemjs/master/dist/system.js

Git - resources

- **Git Essentials** by Ferdinando Santacroce, Packt, April 2015
- Code School - introductory course on Git for the command line
  - https://try.github.io

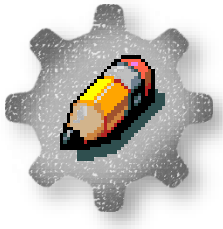


Tools, misc

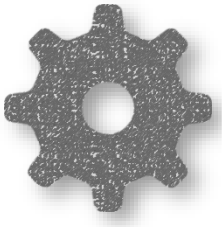


- Scheduling (cron)
  - https://github.com/rschmukler/agenda

Software Configuration Management exercises

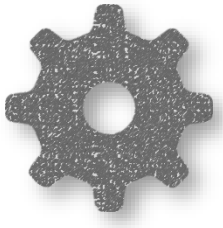


- SourceTree: install and use
- SourceTree : use with GitHub



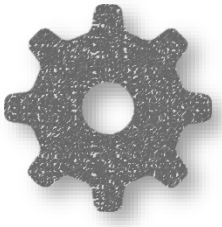
Debugging

JSLint / JSHint

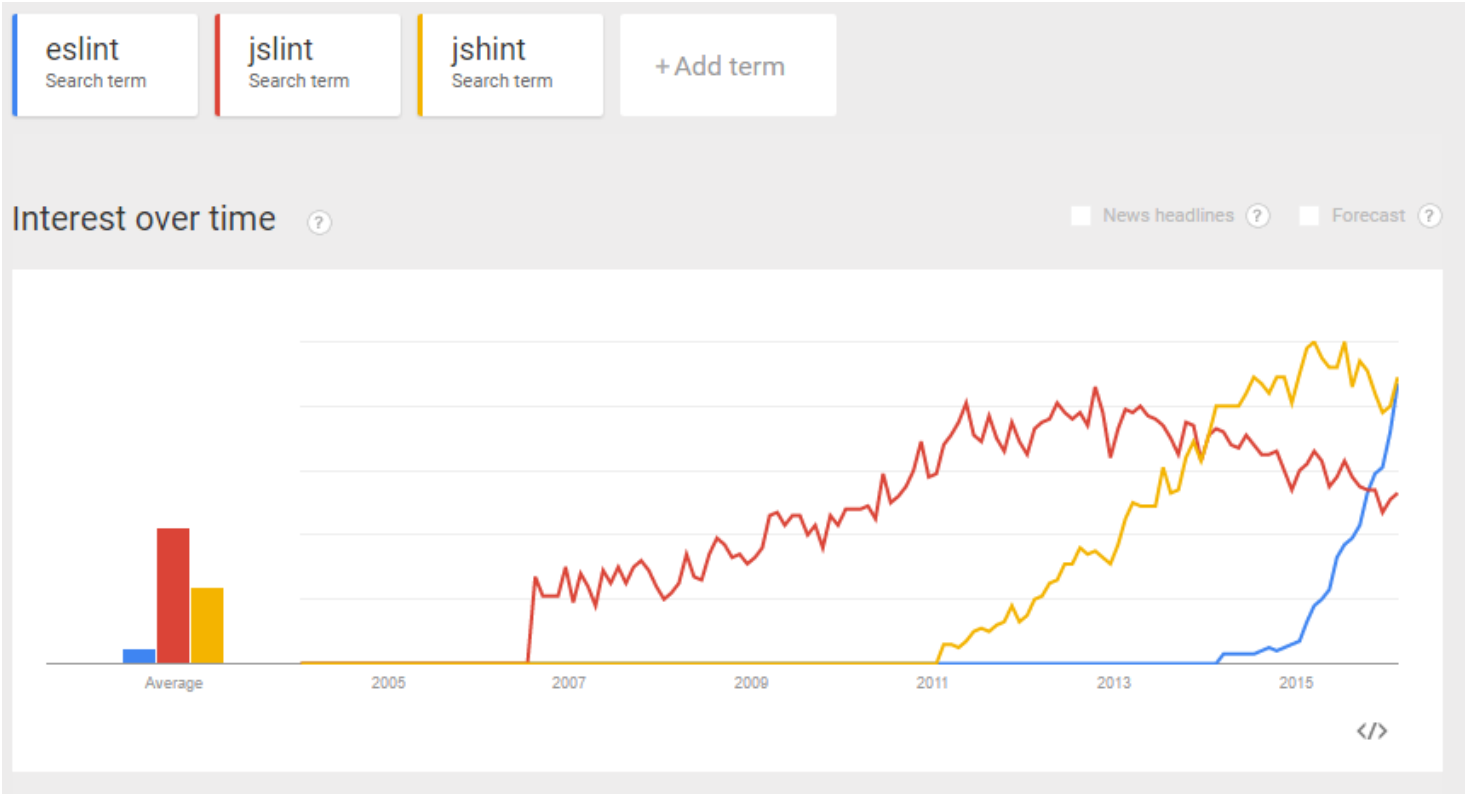


- checks syntax errors in JavaScript
- JSLint
  - developed and maintained by Douglas Crockford
  - opinionated (like DC)
- JSHint forked JSLint
  - http://jshint.com/
  - by Anton Kovalyov

ESLint

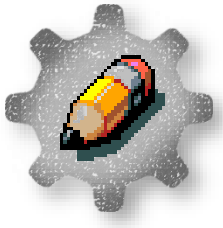


- http://eslint.org/
- Pluggable rule syntax checking



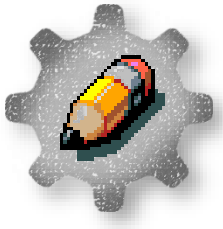
ESLint

Remote debuggers

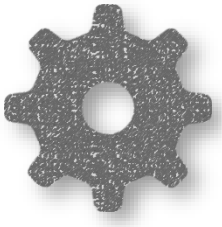


- Ghostlab $ - uses Chrome DevTools
  - https://www.vanamco.com/ghostlab/
- Weinre – uses Safari's WebInspector
  - https://www.npmjs.com/package/weinre
- Browsersync
  - https://www.browsersync.io/
- Not active
  - Microsoft Vorlon.js
  - Adobe Edge Inspect $ - Weinre branch

Dev Tools

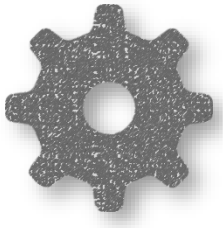


- Debugging interactively
- Node debugging



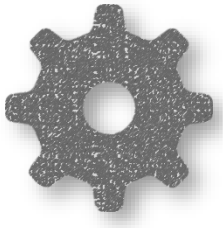
Script loading

Intro



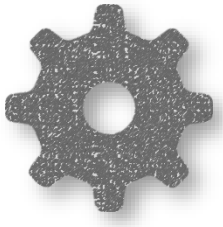
- developer's small modules vs. browser optimization for few requests
- client-side or server-side
- Module system - programmatically load scripts when needed, JIT, client-side
  - <script src="//code.jquery.com/jquery-1.11.0.min.js"></script>
  - <script src="//code.jquery.com/jquery-migrate-1.2.1.min.js"></script>
  - <script src="js/mysite.js"></script>

Client/Server, script loading



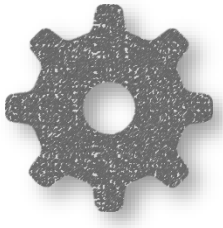
- Module system alternatives
  - JavaScript raw - globals, globals + namespace
  - best use **Require.js** - **CommonJS, AMD**
    - http://requirejs.org/
  - other - ES6, UMD
- Package managers - download scripts with dependencies
  - **npm, Bower**
  - Ender, Volo, Jam, Component

Client/Server, script loading - CommonJS



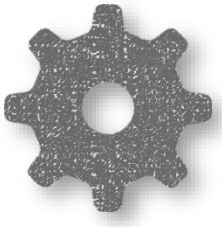
- http://www.commonjs.org/
- 3 variables defined
  - **require** – a function is used to import modules, returning the corresponding exports object.
  - **module** – an object representing the module itself. The module object contains the exports object. In the case of Node.js, it also contains meta-information, such as id, parent, and children.
  - **exports** – a plain JavaScript object, which may be augmented to expose functionality to other modules. The exports object is returned as the result of a call to require.

Client/Server, script loading - CommonJS



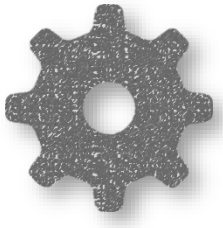
- Used by node.js
- Other CommonJS extensions
  - Browserify - great bundler for client side
  - Ender - http://ender.jit.su/
  - Component - http://componentjs.com/

Client/Server, script loading - Browserify



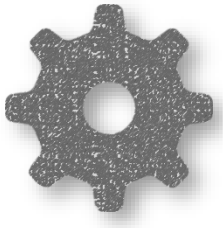
- http://browserify.org/
  - https://github.com/substack/browserify-handbook
- **Module** bundler for npm modules.
- Wraps require( ), keeps track of dependencies.
- Express, grunt, gulp versions

Client/Server, script loading – AMD vs CJS



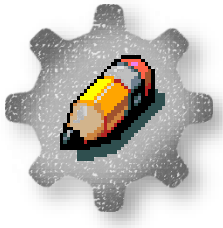
- Underscore removes AMD - con Jan 2012
- Recommended over CommonJS by Require.js
  - http://requirejs.org/docs/whyamd.html#commonjs
- Simplicity - pro AMD
  - http://tagneto.blogspot.com/2012/01/simplicity-and-javascript-modules.html
- AMD is not the Answer - con Jan 2012
  - http://tomdale.net/2012/01/amd-is-not-the-answer/
- Most likely winner will be ES6 in a few years (my opinion)

Client/Server, script loading – ES6 package management

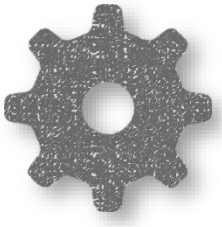


- Rollup - bundler
  - https://github.com/rollup/rollup
- JSPM
  - http://jspm.io/

Exercises

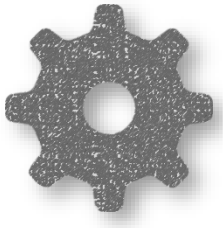


- JavaScript: module loading
- Require.js: module loading



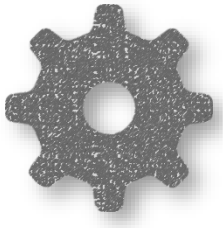
Resources

CDNs



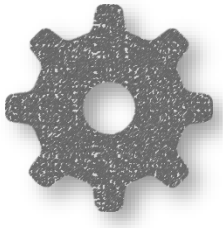
- Content Delivery Networks
  - 3rd party hosting of code, great for testing and speed
  - Mostly JavaScript but some CSS
- Limited packages
  - Google
    - https://developers.google.com/speed/libraries/
  - Microsoft
    - http://www.asp.net/ajax/cdn

CDNs



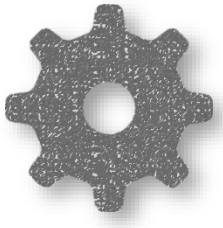
- *CDNJS
  - http://cdnjs.com/
- jsDelivr
  - http://www.jsdelivr.com/

Learning JavaScript



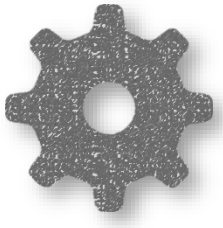
- Codecademy.com - http://www.codecademy.com/tracks/javascript
- Code School JavaScript Path - https://www.codeschool.com/paths/javascript
  - JavaScript, JQuery, Backbone.js, Node.js, CoffeeScript, Ember.js, AngularJS

Learning JavaScript



- News
  - http://javascriptweekly.com
- Tutorials
  - http://www.codecademy.com – Interactive tutorials for JavaScript

Reference



- Mozilla Developer Network
  - JavaScript docs - https://developer.mozilla.org/en-US/docs/Web/JavaScript
  - JavaScript Guide - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide
  - JavaScript reference - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference

MDN MOZILLA DEVELOPER NETWORK