Frontend Tooling by @naoisegolden

Based on @soyjavi's "HTML5 WTF!"

Naoise* Golden Santos

Frontend Developer at XING

naoisegolden @ (gmail, twitter, github,…)

pronounced **nisha*

Task Managers

"a *rake* for javascript"

http://gruntjs.com/

Knowing `Grunt`

The Grunt ecosystem is huge and it's growing every day. With literally hundreds of plugins to choose from, you can use Grunt to automate just about anything with a minimum of effort. If someone hasn't already built what you need, authoring and publishing your own Grunt plugin to npm is a breeze.

Who uses Grunt?

Grunt Code

1. Intalling the CLI

```
npm install -g grunt-cli
grunt --version
```

2. Each time grunt is run, it looks for a locally npm grunt packages declared in Gruntfile

```
module.exports = (grunt) ->
  grunt.initConfig
    pkg: grunt.file.readJSON 'package.json'

    coffee:
      example: files: '/<%=pkg.name%>.js': '/<%=pkg.name%>.*.coffee'
    uglify:
      example: files: “/build/<%=pkg.name%>.js'     : '/<%=pkg.name%>.js'

  grunt.loadNpmTasks 'grunt-contrib-coffee'
  grunt.loadNpmTasks 'grunt-contrib-uglify'

  grunt.registerTask 'default', ['coffee', 'uglify']
```

WebSQL Code

3. Example of Package.json

```
{
  "name"                    : "example",
  "version"                 : "0.0.1",
  "description"             : "...",
  "homepage"                : "http://",
  "author"                  : “John Doe <johdoe@mail.com>”,
  "dependencies": {},
  "devDependencies": {
    "grunt-contrib-coffee"  : "*",
    "grunt-contrib-uglify"  : "*",
    "grunt-contrib-watch"   : "*"}
}
```

Grunt `Hack`

- Create a gruntfile to compile SASS and uglify JavaScript
- Add grunt-gh-pages to deploy to Github Pages
- Add a task to watch that automatically compiles and uglifies

Package Managers

"a *bundler* for javascript"

http://bower.io/

Knowing Bower

Bower is a package manager for the web. It offers a generic, unopinionated solution to the problem of front-end package management, while exposing the package dependency model via an API that can be consumed by a more opinionated build stack. There are no system wide dependencies, no dependencies are shared between different apps, and the dependency tree is flat.

Bower runs over Git, and is package-agnostic. A packaged component can be made up of any type of asset, and use any type of transport (e.g., AMD, CommonJS, etc.).

Bower Code

1. Intalling the CLI

```
npm install -g bower
```

2. Using...

```
# Using the dependencies listed in the current directory's bower.json
bower install

# Using a local or remote package
bower install <package-name>

# Search a determinate package
bower search [<name>]

# Using a different name and a specific version of a package
bower uninstall <package-name>
```

Bower Code

3. Example of Bower.json

```
{
  "name": "lungo-bootstrap",
  "version": "1.0.0",
  "main": "path/to/main.css",
  "ignore": [ ".jshintrc", "**/*.txt" ],
  "dependencies": {},
  "devDependencies": {
    "lungo"          : "*",
    "quojs"          : "*",
    "monocle"        : "*",
    "appnima.js"     : "*",
    "hope"           : "*",
    "CSSmethods"     : "*",
    "device.js"      : "*"
  }
}
```

Bower Hack

- Create a bower.json to auto install Bower Packages
- Change your grunt file to concat Bower JavaScript files in unique file.
- Register your own Bower Package