Grunt.js Essentials

Course Summary

Description

ProTech's Grunt.js training course teaches attendees how to automate repetitive development and management tasks using Grunt, including minification, compilation, unit testing, linting, and more.
Many common tasks are already available as Grunt plugins. Attendees learn to publish their own Grunt plugins or choose from the hundreds of plugins that are already available.

Objectives

At the end of this course, students will be able to:

- Install Grunt
- Create and execute Grunt tasks
- Configure builds for JavaScript applications that use common frameworks such as Require, Modernizr, etc.
- Use Grunt's existing plugins and write new ones

Prerequisites

All attendees must have prior experience developing with JavaScript. If attendees will not have prior JavaScript experience, we would be delighted to precede this class with a one- or two-day JavaScript primer.

Software Needed on Each Student PC

- Firefox with Firebug or Google Chrome
- Internet access
- A local installation of Node.js
- npm
- The latest release of Grunt
- A JavaScript development tool of your choice
- Additional lab files that Accelebrate would provide

Duration

One day

Grunt.js Essentials

Course Outline

I. Introduction

II. Grunt Overview
- A. Compiling Web Apps
- B. Build Tools & Task Runners
- C. Efficiency, Consistency, and Community
- D. Sample Workflows
- E. Built on node.js
- F. Gruntfile

III. Installing Grunt
- A. Installing node.js
- B. Using npm
- C. Installing grunt-cli
- D. package.json and Gruntfile
- E. Installing grunt
- F. Installing Plugins

IV. Tasks
- A. Sections of a Gruntfile
- B. grunt.initConfig
- C. Configurations and Targets
- D. Running Tasks
- E. Task Aliases
- F. Task Arguments & Options
- G. Running Tasks Asynchronously
- H. Logging Errors
- I. Task Dependencies
- J. Loading Tasks via Plugins
- K. File Operations
- L. File Formats & Patterns
- M. Using Templates & Creating Custom Templates
- N. Importing JSON or YAML Data

V. Configuring Builds with Commonly Used Frameworks
- A. Package Manager Tasks - npm and Bower
- B. CSS Preprocessing Tasks - LESS and Sass
- C. CSS Linting Tasks - RECESS
- D. CSS Prefixing Tasks - Autoprefixer
- E. Minification Tasks - CleanCSS and UnCSS
- F. JavaScript Tasks
- G. RequireJS, Modernizr, and Uglify Tasks
- H. JSLint and JSHint Tasks
- I. CoffeeScript and TypeScript Tasks
- J. Unit Testing Tasks
- K. JSUnit, QUnit, Jasmine, Karma and Mocha Tasks
- L. Versioning and Deployment Tasks;

VI. Plugins
- A. Designing a New Plugin
- B. grunt-init
- C. Grunt Plugin Template
- D. Debugging Plugins
- E. Storing Task Files
- F. Publishing a Plugin

VII. Conclusion