**Code Review Checklist**

**General**

1. Site uses a cache buster for expiring .js, .css, and images
2. JavaScript and CSS is minified and concatenated into logical groupings
3. Images have been optimized by ImageOptim (<http://imageoptim.com/>)

**Markup**

1. Code does not contain inline JavaScript event listeners
2. Code does not contain inline style attributes
3. Code does not contain deprecated elements & attributes
4. Page begins with a valid DTD (HTML5 doctype)
5. Code is indented using hard tabs
6. Tags and attributes are lowercase
7. Tags are closed and nested properly
8. Markup is semantic (e.g. class names do not denote presentation, Items in list form are housed in a UL, OL, or DL)
9. Tables are only used to display tabular data
10. Element IDs are unique
11. Code validates against the W3C validator
12. DOM nesting depth does not exceed 12 levels
13. Total page weight does not exceed client requirements (e.g. 1000kb)
14. TItle case is used for headers/titles and forced to all caps using the CSS declaration text-transform: uppercase;
15. Where text is included via images, CSS image replacement is used.

**Forms**

1. All user input is “sanitized”(Make Clean)
2. Form elements are paired with labels elements containing the for attribute
3. Form label/input pairs are wrapped with a block level element (e.g. <p>)
4. Forms are logically arranged within fieldsets

**Accessibility**

1. Page has a proper outline (H1-H6 order)
2. Alt attributes exist on all <img> elements
3. Video is accompanied by a transcript and closed captioning
4. Code validates against WCAG priority level 1 and 2
5. Events and styles applied to :hover are also applied to :focus
6. Tabindex order is logical and intuitive

**CSS**

1. Style blocks are externalized to .css files
2. Consistent naming conventions are used
3. CSS validates against the W3C validator
4. CSS selectors are only as specific as they need to be; grouped logically
5. A print-friendly .css file is included in the page
6. A reset.css file is included in the page
7. CSS sprites are used to combine images
8. CSS selectors gets more specific across files
9. CSS shorthand is used for properties that support it
10. CSS selectors are not tag qualified
11. CSS properties are alphabetical (except for vendor-specific properties) (Why? See: Parable of the brown M&Ms)

**JavaScript**

1. Script blocks are externalized to .js files
2. Consistent naming conventions are used
3. Code adheres to the K&R style
4. Core page features function with JavaScript disabled
5. JavaScript belongs to a namespace (no globally scoped code)
6. jQuery selectors are performant
7. jQuery objects are cached
8. Event delegation is used for binding events to 2 or more elements, or ajax'd elements
9. Script blocks are placed before the closing <body> tag
10. Code has been run through JSLint (jslint.com) or JSHint (jshint.com)

**SEO**

1. Uses a valid <title> element with a valid text node
2. Uses description meta data
3. Uses visible header tags

**Mobile**

1. Functions with JavaScript turned off
2. Image file sizes do not exceed 70kb
3. Appropriate use of HTML inputs (e.g. email, phone, etc) to trigger corresponding on-screen keyboards

**Code Base Checks**

1. All code is checked into SVN or other source code repository
2. Unused sections of code are removed
3. Code is commented where appropriate
4. Client-side code is free of any references to development and staging environments, URLs, or other development settings (e.g. dev Facebook application IDs)
5. Any environmental defaults reference production.
6. Any dependency definitions use exact versions.

**Code Review Tools**

1. Markup: DOM Monster bookmarklet
2. Markup: Diagnose Chrome Extension
3. Markup: W3C HTML validator
4. CSS: W3C CSS validator
5. Performance: YSlow
6. Performance: PageSpeed
7. JavaScript: JSLint/JSHint
8. Accessibility: SortSite
9. Accessibility: JAWS/NVDA/VoiceOver
10. Accessibility: CynthiaSays