

# WEB CODING

This document is referred in the Grading Criteria (rubrics) and contains industry standard best practice.

You are highly encouraged to follow these practices and incorporate them in the assessments in order to produce high quality work and achieve competency or higher grades in this course.

---

## HTML

When producing HTML for a web project at Yoobee School of Design, you should:

- × Use the correct HTML5 doctype
- × Validate HTML <http://validator.w3.org/>
- × Don't use deprecated tags
- × Remove redundant tags and commented out tags
- × Don't use HTML for styling purposes
- × Use the alt attribute for all images
- × Format your code
- × Ensure all file, image and anchor links work
- × Use a consistent HTML structure for all pages unless justified
- × Delivered markup as utf-8 (the most friendly for internationalisation)
- × Don't use tables for layout unless justified
- × Don't use div or p tags for lists unless justified
- × Use label fields to label each form field
- × If nesting views across files (like in some Wordpress templates), use HTML comments to indicate which elements you're closing
- × Don't use images where CSS styles could be used instead
- × Use title-case for headers and titles and use CSS to apply all caps or all lowercase, i.e. text-transform:uppercase/lowercase

---

## CSS

When producing CSS for a web project at Yoobee School of Design, you should:

- × Plan and create your HTML first
- × Use CSS to adhere to a responsive design plan
- × Use only one style sheet per page (IE conditional styles excepted)
- × Validate your CSS <http://jigsaw.w3.org/css-validator/>
- × Use comments to add meaning
- × Include a CSS reset at the start of the style sheet
- × Organise the style sheet with a top down structure from generic to specific
- × Use classes (or multiple classes), combine rules and take other measures to reduce repetition
- × All CSS should be consistent with a methodology, such as SMACSS, BEM, or OOCSS
- × Reduce specificity as much as your CSS methodology will allow
- × Selectors must be named consistently and meaningfully, not start with numbers and have no spaces or special characters
- × Use absolute positioning only when default positioning can't fulfill a use case
- × If using floats, remember to clear them
- × Don't use an image where there is a CSS alternative
- × Use external files unless justified

# WEB CODING

This document is referred in the Grading Criteria (rubrics) and contains industry standard best practice.

You are highly encouraged to follow these practices and incorporate them in the assessments in order to produce high quality work and achieve competency or higher grades in this course.

---

## CSS cont'd

When producing HTML for a web project at Yoobee School of Design, you should:

- × Use the cascade appropriately
- × Use inheritance where possible
- × Don't over-qualify selectors
- × Use shorthand for properties like padding and margin
- × Don't leave commented out styles in production CSS
- × Compress CSS for production

---

## Javascript

When producing Javascript for a web project at Yoobee School of Design, you should:

- × Use external scripts unless justified
- × Minimise the number of external scripts a page uses
- × Acknowledge all third party scripts using comments
- × Code should be written to the DRY principle i.e. no repetition. Create functions that take parameters, and return values to foster code reuse
- × Comment your code <https://github.com/shri/JSDoc-Style-Guide>
- × Ensure site is usable when Javascript is turned off
- × Use consistent naming: functionNamesLikeThis, variableNamesLikeThis, ClassNamesLikeThis, EnumNamesLikeThis, methodNamesLikeThis, CONSTANT\_VALUES\_LIKE\_THIS, and file-names-like-this-1.0.min.js
- × Name functions and variables logically
- × Prefer single quotes over double quotes
- × Always declare variables with var (or in ES6 - var, let, const)
- × Always use semicolons
- × Don't declare a named function within a block [https://google.github.io/styleguide/javascriptguide.xml?showone=Function\\_Declarations\\_Within\\_Blocks#Function\\_Declarations\\_Within\\_Blocks](https://google.github.io/styleguide/javascriptguide.xml?showone=Function_Declarations_Within_Blocks#Function_Declarations_Within_Blocks)
- × Minimize the use of global variables
- × Separate event handling from functionality
- × Minify all Javascript for production
- × Use a Javascript linting tool
- × Organize your code into the Modular Pattern, Object Literal/Singleton, or Objects with Constructors where possible