

Lesson 12: Writing Your Best Code

HTML Coding Practices

- write standards-compliant markup:
 - nest & close all elements correctly
 - use ids & classes appropriately
 - always validate code
- make use of semantic elements:
 - research & double-check proper semantic elements
 - perform routine code reviews
 - builds a more accessible website
- use the proper document structure:

```
<!DOCTYPE html>
<html>
<head>
<body>
```
- keep the syntax organized:
 - use lowercase letters w/in element names, attributes, & values
 - indent nested elements
 - strictly use double quotes
 - remove the slash at the end of self-closing elements
 - omit the values on boolean attributes
- use practical id & class values: should relate to content > styling
- use the alternative text attribute on images: screen readers & other accessibility software relies on alt attribute to provide context for images
 - should be very descriptive of what the image contains
 - if img is irrelevant → include blank alt text
 - use `css background-image > `
- separate content from style: use external stylesheets > inline styles
- avoid a case of "divitis": keep code lean
 - reuse styles → reduce markup
 - use HTML structural elements where suitable
- continually refactor code: remove old code & styles
 - look for ways to condense code & make it more manageable → refactor code

CSS Code Practices

- organize code w/ comments:
 - keep styles organized in logical groups
 - provide a comment before each group noting what it pertains to
 - (optional) build a table of contents at the top of the file
- write css using multiple lines & spaces:
 - place each selector & declaration on a new line
 - indent declarations w/in each selector
 - put a space - before the {

after the :
end each declaration w/ a ;

- use proper class names: should be modular
should pertain to content w/in an element > appearance, a.m.a.p
use lowercase letters & hyphen delimiters
- build proficient selectors: the longer a selector → the higher the specificity → the more likely it'll break the CSS cascade
avoid using ids w/in selectors
use shorter & primarily more direct selectors
nest selectors max= 2-3 levels deep
remove as many location-based qualifying selectors as possible
- use specific classes when necessary: use a class > a long & specific selector → renders faster
- use shorthand properties & values: → quickly set & identify styles
but not if only setting one value
- use shorthand hexadecimal color values: use 3-character hexadecimal color value whenever available
always use lowercase letters
- drop units from zero values: no matter which length unit is being used
it's unnecessary & provides no value
- group & align vendor prefixes: → property names & values stack vertically
→ style is easier to read & edit
alignment may vary depending on if the vendor prefix is placed on the property or the value
unprefixed version goes last/after all prefixed versions
- modularize styles for reuse: specifically w/ the use of classes
share styles assigned to a class across elements as necessary