PRACTICE

HW 1: Items 1-7

HW 2: Items 1-12

HW 3: Items 1-23

HW 4: Items 1-23

HW 5: Items 1-28

HW 6: Items 1-30

1. Include the <!DOCTYPE html>
2. Include the html, head, and body elements
3. Ensure the head has a title
4. Use proper semantics
5. Do not have HTML5 syntax errors: validate!
6. If an anchor has *target="_blank"*, also use *rel="noopener"*
7. Always give an alt for images and possible error messages for audio/video
8. Do not place block-level elements within p elements except for div
9. To include CSS, always link to an external style sheet
10. Never use !important
11. Always include a backup generic font
12. Do not have CSS syntax errors: validate!
13. To include JS, always source an external file
14. Use defer or async in the script tag in the head
15. Always use semicolons for statements
16. Include braces for all control flow structures
17. Use === and !== but not == or !=
18. Use only let and const as keywords to define variables
19. Use const when a variable should be constant
20. Use objects as a namespace
21. Use IFFEs to not pollute the global environment
22. Use primitives instead of their wrapper objects, "7" instead of " new Number(7) ", etc.
23. Debug code
24. Always use <?php … ?>; do not abbreviate to <? … ?>, etc.
25. Always use single quotes for PHP unless a string needs to be parsed
26. Avoid function calls within loops when possible
27. echo multiple items with a comma rather than concatenating
28. Add a single element to a PHP array with []

29. Never “SELECT *” in a SQL query
30. Always explicitly list the columns being INSERTed into with SQL

READABILITY

HW 1: Items 1-2

HW 2: Items 1-5

HW 3: Items 1-12

HW 4: Items 1-12

HW 5: Items 1-12

HW 6: Items 1-13

1. Always indent child elements unless they are html, head, or body
2. Use lowercase for all tags
3. Separate each property and value on a different line
4. Organize the style sheet in sequential order where possible and provide some comments
5. Avoid CSS code bloating by combining elements when possible
6. Use descriptive class names based on function/purpose
7. Comment code
8. Comment every branch of control flow
9. Document functions in the prescribed JS manner
10. Use descriptive variable names
11. Avoid magic numbers
12. Space code neatly
13. Use ALL CAPS for SQL keywords and commands