Why should you care about how your code is formatted?

It aims at improving collaboration, code quality, and enabling supporting infrastructure. It applies to raw, working files that use HTML and CSS, including GSS files. Tools are free to obfuscate, minify, and compile as long as the general code quality is maintained.

How many spaces should you indent at a time?

Indent by 2 spaces at a time. Don’t use tabs or mix tabs and spaces for indentation.

Should the code be lowercase or uppercase?

Use only lowercase. All code has to be lowercase: This applies to HTML element names, attributes, attribute values (unless text/CDATA), CSS selectors, properties, and property values (with the exception of strings).

What is the recommended encoding?

Use UTF-8 (no BOM). Make sure your editor uses UTF-8 as character encoding, without a byte order mark. Specify the encoding in HTML templates and documents via <meta charset="utf-8">. Do not specify the encoding of style sheets as these assume UTF-8.

How should you use comment to explain your code? How do you do it in HTML and in CSS?

Explain code as needed, where possible. Use comments to explain code: What does it cover, what purpose does it serve, why is respective solution used or preferred?

HTML : <!-- --> CSS : /\* \*/

How do you validate your HTML?

Use valid HTML where possible. Use valid HTML code unless that is not possible due to otherwise unattainable performance goals regarding file size. Use tools such as the [W3C HTML validator](https://validator.w3.org/nu/) to test. Using valid HTML is a measurable baseline quality attribute that contributes to learning about technical requirements and constraints, and that ensures proper HTML usage.

What is the purpose of the alt attribute?

Provide alternative contents for multimedia. For multimedia, such as images, videos, animated objects via canvas, make sure to offer alternative access. For images that means use of meaningful alternative text (alt) and for video and audio transcripts and captions, if available. Providing alternative contents is important for accessibility reasons: A blind user has few cues to tell what an image is about without @alt, and other users may have no way of understanding what video or audio contents are about either. (For images whose alt attributes would introduce redundancy, and for images whose purpose is purely decorative which you cannot immediately use CSS for, use no alternative text, as in alt="".)

Should you separate structure (markup), presentation (styling), and behavior (scripting)? If yes, why?

Separate structure from presentation from behavior. Strictly keep structure (markup), presentation (styling), and behavior (scripting) apart, and try to keep the interaction between the three to an absolute minimum. That is, make sure documents and templates contain only HTML and HTML that is solely serving structural purposes. Move everything presentational into style sheets, and everything behavioral into scripts. In addition, keep the contact area as small as possible by linking as few style sheets and scripts as possible from documents and templates. Separating structure from presentation from behavior is important for maintenance reasons. It is always more expensive to change HTML documents and templates than it is to update style sheets and scripts.

Should you use double or single quotation marks around attribute values?

When quoting attributes values, use double quotation marks. Use double ("") rather than single quotation marks ('') around attribute values.

How can you validate your CSS code?

Use valid CSS where possible. Unless dealing with CSS validator bugs or requiring proprietary syntax, use valid CSS code. Use tools such as the [W3C CSS validator](https://jigsaw.w3.org/css-validator/) to test. Using valid CSS is a measurable baseline quality attribute that allows to spot CSS code that may not have any effect and can be removed, and that ensures proper CSS usage.

When would you use meaningful ID/class names? When would you use generic ones?

Use meaningful or generic ID and class names. Instead of presentational or cryptic names, always use ID and class names that reflect the purpose of the element in question, or that are otherwise generic. Names that are specific and reflect the purpose of the element should be preferred as these are most understandable and the least likely to change.

Generic names are simply a fallback for elements that have no particular or no meaning different from their siblings. They are typically needed as “helpers.” Using functional or generic names reduces the probability of unnecessary document or template changes.