



[Course](#) > [Modul...](#) > [5.5 CSS...](#) > CSS be...

## CSS best practices (OPTIONAL)

# CSS best practices

You will find below an excerpt of CSS best practices (see the [full slide set](#)) that were written by Erika J. Etemad (also known as [fantasai](#)). Erika is an expert on the [W3C CSS Working Group](#) (since 2004!) and a longtime contributor to the Mozilla Project. In addition to editing many of the CSS3 specifications, she's worked on layout engine testing and development for Gecko and managing the CSS test suites at W3C.

### Executive summary

- **Logical source order:**  
The order of the HTML content should make sense even without the CSS: for accessibility, mobile optimization, device adaptability, and long-term maintainability.
- **Liquid layouts and relativity:**  
Use smart relative sizing: to optimize layouts while minimizing media query code forks.
- **Media queries:**  
Adapt to screen size changes; get font size adaptation free by using ems.
- **Prevent zombie code:**  
Dead code may come alive as CSS changes. Delete it before it does, and ruins your layout.
- **Test in multiple browsers:**  
Your favorite browser is not always right.
- **Don't use proprietary features!**  
Keep the Web open to everyone! Don't rely on the latest - WebKit- invention.
- **Turn off CSS:**  
A well-coded page will be understandable without it.



## Foundations

- Indent your code for readability ease
- Learn how to code CSS before relying on frameworks (such as Bootstrap, etc.)
- **Separate content and style**
  - Use semantic markup, ie., "classes for meaning, not for show".  
The following article is helpful to understand this concept: [Meaningful CSS: Style Like You Mean It](#) (Tim Baxter, May 2016 - A list apart).
  - Use `<table>` for tabular data: don't use tables for layout, but if your content is tabular like a catalog, a calendar, or a price list, then the table element is the correct markup.
- **Linearized logical source order**  
The order of the HTML content should make sense even without the CSS.  
Benefits are numerous as it *works best*:
  - for long-term site maintainability
  - for mobile
  - for accessibility
  - as a foundation for device adaptation (media queries)
- **Linguistic variations**: set the language correctly for better typography (see the section entitled "[why Internationalization is important](#)")

## Testing

- **Test without CSS**: turn off CSS, and if the page makes no sense, fix your markup.
- **Test in multiple environments**:
  - Resize the window
  - Zoom the text
  - Try a mobile browser
  - Navigate by keyboard
- **Test in multiple browsers**: remember that just testing in Chrome does not work for everyone! ;)



## Adaptability

- Containing block size? → Use percents.
- Viewport size? → Use viewport units: `vw`, `vh`, `vmin`, `vmax`
- Font height? → Use `em` or `rem`.
- Font pitch? → Use `em` or `ch`.
- Content size? → Use `auto` or `min-content`/`max-content`.
- Combination of the above? → Use the appropriate layout formulas: `flex`, `min-width`, `max-width`, etc.

## Defensive Coding

- Use `!important` to define overriding rules, not for fixups
- Duplicate selectors if you need to increase specificity, or
- Simplify selectors if you need to decrease specificity

- For example, avoid:*

[illegible]

- **Drop dead code:** you tried something and it didn't work? Delete it right away!
- Code to Standard
- Don't rely on proprietary extensions
- Don't use experimental features in production *or* commit to keeping up-to-date.
- Provide fallbacks / use `@supports`.