# CS 328 CSS Coding Standards so far

last modified: 2024-03-01

- You are expected to use `normalize.css` for your documents.

  - You may change that link from `html-template.html` to point to a different copy of `normalize.css`, however, if you prefer.

  - Place any additional `link` elements for additional external style sheet(s) **after** the `link` element for `normalize.css` (but still **within** the `head` element).

- Except for `normalize.css`, you may **not** use any CSS frameworks or predefined libraries unless you get **prior**, **explicit** approval. One of this course's purposes is to provide you with some practice with the basics of "plain" CSS.

- In general, you are expected to use external CSS style sheets -- avoid use of internal or inline CSS styles (unless you have a VERY good reason, and have gotten prior, explicit approval).

  - for CS 328, you are expected to use the following `link` element for these, placed within the `head` element:

    `<link href="`*url-of-.css-file*`" type="text/css" rel="stylesheet" />`

    (It is perfectly fine to write this across more than 1 line if the external CSS file's URL is long.)

- You are expected to write valid CSS for CS 328.

  - Your external CSS files should validate using the W3C CSS Validator at https://jigsaw.w3.org/css-validator/.

- From **Homework 4** onward, you are expected to use foreground and background colors that are at least WCAG 2 AA Compliant based on the tester at:

  https://snook.ca/technical/colour_contrast/colour.html

- Remember that CSS property names should always be all-lowercase.

- CS 328 CSS rule style standards:

  - no `property: value;` declaration(s) should be on the same line as a rule's selector(s)

  - each `property: value;` declaration within the rule should be indented by 3 or more spaces, and lined up even with the others within that rule

  - only put ONE `property: value;` declaration per line

  - even though it is not syntactically required, put a semicolon after the last `property: value;` declaration within a rule

  - for example:

    ```
    p.centered
    {
        text-align: center;
        color: red;
    ```

```
    }
```

- If you give a value for property `font-family` in a rule, include a generic font family as one of the options. For example,

```
p.even_steven
{
    font-family: "Courier New", monospace;
}
```

- Because a class goal is to work toward responsive design, use relative units for sizes instead of absolute units -- for example, use units such as `em` or `%` instead of `px` or `pts`

    - (Note: this is not to imply that this is all there is to responsive design, by a long shot -- I understand there are whole books devoted to this -- BUT it is a small step in the right direction, I hope.)

- HTML forms should not use the `table` element for layout purposes -- CSS should be used for such layout instead. The `table` element should be only used for truly-tabular data.