

Seven basic layers:

Settings: Global variables, configuration switches. The wider layer affects a large part of the DOM

Tools: Global mixins, helpers & functions.

Generic: Low level styles, normalise.

Elements: HTML type selectors (h1, p, lists, links...). Lower layer don't use type selector. Unclassed HTML elements.

Objects: Cosmetic-free design patterns, generic names. First layer with class selectors.

Components: Affects the UI. Only class selector. Designed components. A bit more explicit.

Trumps: Helpers, overrides and exceptions. High specificity selectors, ID & *!important*. Target a single DOM node.

Add a rule :

- How high is the selector specificity?
- How much of the DOM is impacted?
- Is this an override?
- If it is, should we create a new component instead?

Do & don't :

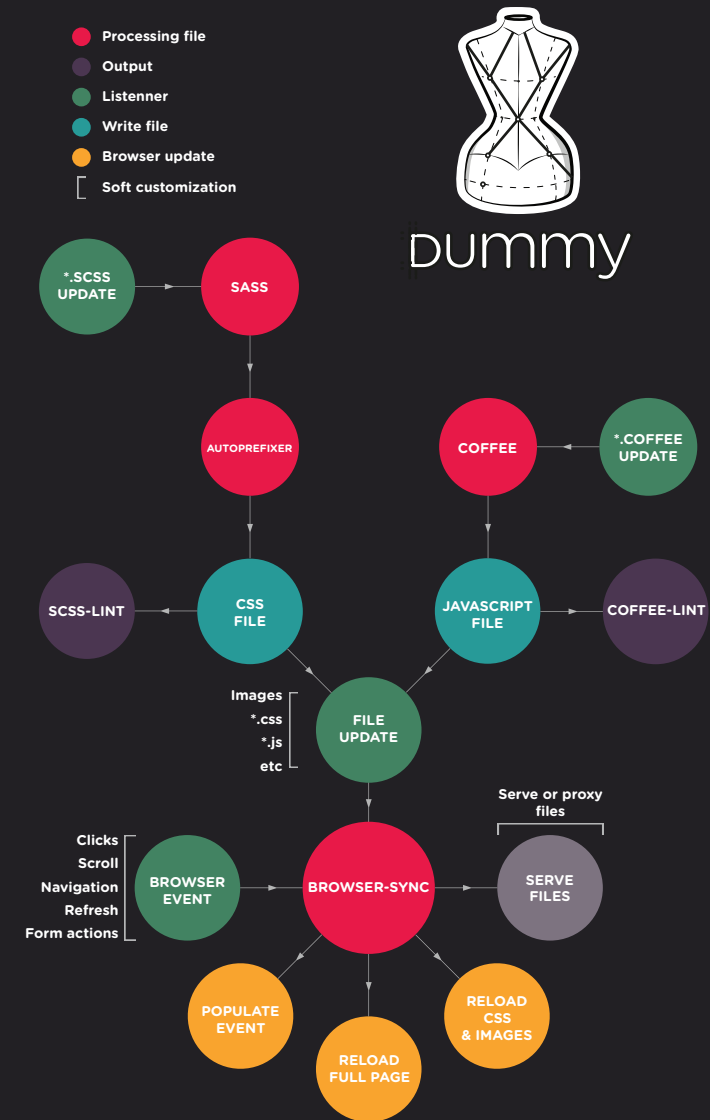
Avoid contextual styles, use OOCSS and Sass inheritance.

Add new layers as you need to: Theming and A/B testing layers goes right up the trumps.

Move the complexity to the HTML, adding new classes to the html is easy and helps to keep your CSS sane.

Open/Close principle, elements should be closed to modification and open to extension.

Never override a rules in a lower layer.

**Configuration:**

```
grunt/parameters.coffee
```

Installation:

```
github.com/Inuit/dummy/#usage
```