1. Use **Normalize CSS** to avoid cross browser issues.There are two ways to **use Normalize** in a project: edit the source to customize your own **Normalize** stylesheet, or **use** it as a base and add styles on top.
2. **Use meaningful naming conventions for classes or IDs**. For example, *warning*, *important*, *downloadableImage* and *submenu* are all good names. They describe what a certain element represents, and they are not likely to change.
3. **Keep it Consistent**. For example, I use ".caption-right" to float images which contain a caption to the right.There are certain classes that you can create in nearly every theme, and you can use the same name each time
4. **Initialize the pseudo classes** . A **CSS pseudo-class** is a keyword added to a selector that specifies a special state of the selected elements. For example, hover, focus, blur and etc.
5. **Organize the Stylesheet with a Top-down Structure**. It always makes sense to lay your stylesheet out in a way that allows you to quickly find parts of your code.
6. **Make Use of Generic Classes**. Instead of adding that particular style to each ID, you can create generic classes and add them to the IDs or other CSS classes. For example, I find myself using float:right and float:left over an over in my designs. So I simply add the classes .left and .right to my stylesheet, and reference it in the elements.
7. **Don't use Negative Margins or Paddings**. Negative margins and paddings is that they’re don’t exist in real life.
8. Check for Closed Elements First When Debugging.If you're noticing that your design looks a tad wonky, there's a good chance it's because you've left off a closing *</div>*.
9. **Avoid Using Inline Styles**. Use of **inline styles** in HTML is a bad practice because the browser doesn't understand it well.
10. **Don’t use !important use nesting**. Nesting is a feature of CSS preprocessors, allowing authors to nest selectors within selectors in order to create shortcuts.
11. **Use !important only when override the inline styles**. Adding the !important keyword to any CSS rule lets the rule **forcefully precede over all the other CSS rules** for that element.It even **overrides the inline styles** from the markup.
12. **Don’t define/override the names for classes or IDs.**
13. **Use the shorthand css**. **Shorthand** properties are **CSS** properties that let you set the values of multiple other**CSS** properties simultaneously.
14. **Maintain separate file for css hacks, with browser name**.A **CSS hack** applies **CSS** in one or more specific browser versions while that same**CSS** will be ignored by other browsers.
15. **Create separate file for theming don't mix with the other files**.
16. **Define all SCSS variables in the \_variable.scss**. Sass variables are *imperative*, which means if you use a variable and then change its value, the earlier use will stay the same.
17. Maintain \_ prefixes for all configuration file example themes, typos, variables and etc.
18. Use the latest feature to create classes in scss like mixins, extends and each