Coding Standards

Most important - BE CONSISTENT

Naming Convention

HTML & CSS

- Use .dash-seperated-classes #and-ids
- Don’t use shrtnms
- Use semantic names as much as possible - what’s the purpose of this element, not how it looks
- Strive to use one word to describe an element
- Don’t add redundant information e.g. “title-box” => “title”, “nav-list” => “nav”, “header-menu” => “header” or ”menu”
- Use <i> for presentational elements, it’s short and descriptive (i.e. “icon”)
- Use HTML5 tags as much as possible (header, footer, section, nav, aside, figure, etc)
- Use semantic tags for headings - h1, h2, etc
- Use Microformats/Microdata (Schema.org) where appropriate
- Use WAI-ARIA Roles where appropriate
- Avoid wrapping-up (like “element” < “element-wrap” or “element-box”), if no semantic name is available try to wrap-down (like “element” > “element-inner”)

JavaScript

- Strive to use one word to describe an object/variable
- Strive to use action words to name methods (e.g. create, start, init, get, etc)
- Use Capital Case for ClassNames (which you instantiate with new keyword)
- Use lowercase for namespaces and object literals
- Camelcase for variableNames
- TBD - Use JSLint as part of the development process

Filenames

- Use lowercase names with underscores for file_names

CSS Engineering

General

- Keep selectors (AFAP) as flat as possible
- CSS should be flat, the day it’ll be flat it’ll be reusable

- Or as Nicole Sullivan put it: “Work with CSS, not against it”
- Like the Normal Flow - learn it and use it
- Use OOCSS and SRP design patterns or SMASS
- Tips for DRYer CSS: http://csswizardry.com/2013/07/writing-dryer-vanilla-css/
- As a rule of thumb - redefining things in CSS is wrong. For some more anti-patterns: http://csswizardry.com/2012/11/code-smells-in-css/

LESS & SASS

- D O N ’ T N E S T
- Unless you have too
- Rule-of-thumb: if you reach 3 levels of nesting - start smelling for something fishy
- Rule-of-thumb: if your nesting scope is bigger than your thumb - something’s very wrong
- Mixins you must-have
    - Typography
    - Resets
    - Padding, margin, borders, align
    - Matomy mixins are good example
- Use frameworks for stuff like CSS3, grids, etc.
    - for SASS - you have Compass
    - for LESS - you have Bootstrap, Less Elements

Javascript Engineering

The MSO (Many-Small-Objects) Pattern

- Use many objects that do few things
- Rule-of-thumb: if you have two functions and one variable which are related to the same issue - you should wrap them with an object
- Wrap a collection of such related MSOs with a module
- A collection of such modules is an application / large-scale website
- Ux on Beer is a good example of this pattern (at small-scale)

Whitespace, etc

- Indent style is tabs (with the width of 4)
- Everything else is in the example: http://codepen.io/krulik/pen/ojLHC

References

- http://www.2ality.com/2013/07/meta-style-guide.html
- http://addyosmani.com/blog/javascript-style-guides-and-beautifiers/
- http://javascript.crockford.com/code.html
- https://github.com/rwldrn/idiomatic.js/

- https://github.com/necolas/idiomatic-css/
- http://na.isobar.com/standards/