CSS Code Conventions

2010/06/21 Updates

- 有時模組有需要用到第二階層，可以用 .shd, .sbd, .sft 做配合

```
<div id="mod">
    <div class="bd">
        <ul>
            <li>
                <div class="shd">...</div>
                <div class="sbd">...</div>
                <div class="sft">...</div>
            </li>
        </ul>
    </div>
</div>
```

- 用 .first-child 或 .last-child 作為第一筆及最後一筆的 class
- 訊息有下列三種樣式： em.info (green) em.highlight (桃紅) em.warn (紅)
- 統一縮寫
  - mod = module
  - button
  - func = function
  - intro = introduction

Required Comments

Each file must begin with a comment block in the following format:

```
/**
 * Today Module CSS
 * @module type_today_css
 * @author mseeley, dharmesh, cklaiber
 * @see http://twiki.corp.yahoo.com/view/FrontDoors/MetroTechSpecToday
 * @version CVS &#36;Id:$
 *
 * Review History:
 * 25-May-09 [bug 123456] acastro
 * 12-Jun-09 [bug 123456] nzakas
 */
```

A brief description should be included as well as a module name. The module name

should be unique in the product and should end up =_css=. A YUI Doc module is just a grouping of functionality, such as the DOM Utility, which is made up of numerous parts. This comment section should include the owners of the file, a URL to the documentation, CVS version, and code review history. Each entry for a code review should contain the date that the review was completed, the bug tracking the review, and the person who did the review.

General Format

In general, every CSS rule begins with a selector, followed by a left curly brace on the same line. Each CSS property name-value pair should be indented 4 spaces (no tabs), on it's own line ending with a semicolon. Example:

```
.selector1 {
    name1: value1;
    name2: value2;
    name3: value3;
}
```

The rule attribute names should be in alphabetical order.

Units of Measurement

The rules regarding units of measurement are as follows:

- Padding/margin should be specified in ems, not pixels.
- Borders can be specified in pixels.
- Any measurements related to images can be specified in pixels.

The use of ems only for padding / margin should not be considered a hard and fast rule. For example, if you are adding padding to an element, so that you can display a sprite as a background image, then pixels are often a better choice as the space required will not change when font size changes. Background image offsets should always be specified in pixels.

Consistency is also something to consider. It is generally better to be consistent with units used on each element, for example, don't specify left padding in pixels and right padding in ems. When you need to specify a width, padding / margin should also be specified in the same units as the width:

This will work at all font-sizes:

```
.type_module-name div {
    width: 10em;
    padding: 0 2em;
}
```

but the following is unpredictable because the padding may make the box wider than the parent element, i.e. 20px may be larger than 4% which would put the container over 100% width:

```
.type_module-name div {
    width: 96%;
    padding: 0 20px;
}
```

Selector Naming

Selectors should be descriptive of function rather than form. For instance, "red" and "bold" should not be used as class names because they indicate the form of the text rather than its function. Selectors such as "button" or "hd" (short for "head") are preferred because they indicate use.

Selector names should use dash case, meaning all words are lowercase with dashes as separators (=word1-word2-word3=). Examples:

- submit-button
- mod-hd
- user-icon

Note: Automatically generated class names, such as those for module types, use an underscore to separate words. This is intentional to tell the difference between these and other, developer-defined selectors.

Selector Specificity

Selectors should be as *least* specific as possible:

Bad

```
.type_module-name div ul li .exampleclass {
    ...
}
```

Good

```
.type_module-name .exampleclass {
    ...
}
```

Writing very specific rules means that you have to write even more specific rules to override them, and this is complicated, difficult and unnecessary. The cascade is your friend :)

There was previously an example of using IDs to sandbox styles. As all IDs used inside modules should be namespaced using the module ID (which is not always predictable) it is not possible to use these IDs for CSS hooks.

Module-Specific Rules

All module styles should begin with the module type (*type_xxxxx*). For example, for a module called *type_ymail*, the rule should be formatted like this:

```
.type_ymail .selector1 {
    name1: value1;
}
```

This ensures that the styles are applied to this module only.

Multiple Selectors Format

When writing CSS code, multiple selectors separated by a comma should each appear on their own line. For example:

```
.selector1,
.selector2,
.selector3 {
    name: value;
}
```

English Only

You shouldn't write any non-english characters including comments.

Font Size

Use YUI Fonts

Layout

Use YUI Grids