CSS Coding Standards

TRIASSIC
Get IT right

Amendment Record			
Version	**Date**	**Reviewed and Approved by**	**Change Description**
1.0	22-Jul - 2019	Techbench	Initial Version

TABLE OF CONTENTS

Terminology

Concise terminology used in these standards:

```
selector {
  property: value;
}
```

property: value makes a *declaration*. Selector and declarations makes a *rule*.

Write valid CSS

All CSS code must be valid CSS3.

When using vendor prefixed properties, you can ignore CSS validation errors it generates.

Encoding of CSS files

Encoding of CSS files should be set to UTF-8.

Formatting

All CSS documents must use **two spaces** for indentation and files should have no trailing whitespace. Other formatting rules:

- Use soft-tabs with a two space indent.
- Use double quotes.
- Use shorthand notation where possible.
- Put spaces after `:` in property declarations.
- Put spaces before `{` in rule declarations.

- Use hex color codes `#000` unless using `rgba()`.
- Always provide fallback properties for older browsers.
- Use one line per property declaration.
- Always follow a rule with one line of whitespace.
- Always quote `url()` and `@import()` contents.
- Do not indent blocks.

```
.media {

 overflow: hidden;

 color: #fff;

 background-color: #000; /* Fallback value */

 background-image: linear-gradient(black, grey);

}

.media .img {

  float: left;

  border: 1px solid #ccc;

}

.media .img img {

  display: block;

}

.media .content {

 background: #fff url("../images/media-background.png") no-repeat;

}
```

Naming Conventions

Always use hyphens in class names. Do not use underscores or CamelCase notation.

```
/* Correct */

.sec-nav

/* Wrong */

.sec_nav

.SecNav
```

Values

Always define generic font families like sans-serif or serif.

```
/* Correct */

font-family: "Open Sans", Arial, Helvetica, sans-serif;

/* Wrong */

font-family: "Open Sans";
```

Shorten hexidecimal color values to 3 digits when possible:

```
background: #fff;
```

If you use 0 as a value, do not add a unit (px, em, etc.) after it.

```
/* Correct */
.nav a {
 padding: 5px 0 5px 2px;
}
/* Wrong */
.nav a {
 padding: 5px 0px 5px 2px;
}
```

Do not use default values if they are not necessary to override inherited values.

Selectors

Selectors should be on a single line, with a space after the selector, followed by an opening brace. A selector should end with a closing brace on the next line. Next selector related the the previous one should be on the next line with one additional line space between them.

```
.nav li {

 }

.nav a {

 }
```

Avoid very complex child and descendant selectors like:

```
/* Wrong */

.my-inbox .flyout-content .inner .message .inbox li div.take-action .actions ul li a  {

 }
```

Multiple selectors

Multiple selectors should each be on a single line, with no space after each comma.

```
.faqs a.open ,

.faqs a.close  {

 }
```

Properties

Every declaration should be on its own line below the opening brace. Each property should:

- have a single soft tab with 2 spaces before the property name and a single space before the property value.
- end in a semi-colon.
- Include a semi-colon at the end of all declarations, including the last declaration in a declaration block.
- When hex values are used for colors, use lowercase and, if possible, the shorthand syntax, e.g. `#aaa`. Colors may be expressed with any valid CSS value, such as hex value, color keyword, rgb() or rgba(). Note that IE8 does not support all color syntaxes and will require a fallback value.
- For property values that require quotes, use double quotes instead of single quotes,

  e.g. `font-family: "Arial Black", Arial, sans-serif;` and `content: " ";`.

- If a property does not require quotes (e.g. `url()`, do not add them.

  This means `background-image: url(path/image.png)`

  instead of `background-image: url("path/image.png")`

- Quote attribute values in selectors,

  e.g. `input[type="checkbox"]`.

- Include a space after each comma in comma-separated property or function values.
- Do not use spaces around the parentheses in a function,

  e.g. `color: rgba(0, 0, 0, 0.8);`

- Use lower case function names,

  correct: `color: rgba(0, 0, 0, 0.8);`

  incorrect: `color: RGBA(0, 0, 0, 0.8);`

```
.site-name span {
  position: absolute;
  top: 0;
  left: 0;
  z-index: 10;
}
```

Shorthand properties

Use shorthand properties when possible.

Order of properties

Order of properties can have the following structure: box model, typography and graphic layer or order properties alphabetically.

Properties with multiple values

When properties can have multiple values, each value should be separated with a space.

```
font-family: "Lucida Grande", "Lucida Sans Unicode", Verdana, lucida, sans-serif;
```

Preprocessors

- Limit nesting to 1 level deep. Reassess any nesting more than 2 levels deep. This prevents overly-specific CSS selectors.
- Avoid large numbers of nested rules. Break them up when readability starts to be affected. Preference to avoid nesting that spreads over more than 20 lines.
- Always place `@extend` statements on the first lines of a declaration block.
- Where possible, group `@include` statements at the top of a declaration block, after any `@extend` statements.

```
.selector-1 {

  @extend .other-rule;

  @include clearfix();

  @include box-sizing(border-box);

  margin: 10px;

  padding: 10px;

}
```

Comments

Follow the comments style used in normalize.css. The comments blocks should be a maximum of 80 characters wide.

This comment style is used as the separator of the main sections. There are 2 empty lines before and after it:

```
/* ============================================================================

 Section comment block

 ============================================================================*/
```

The following comment style is used as the separator of the subsections of the main sections. It has 2 empty lines before it and 1 empty line after it:

```
 /* Sub-section comment block

=========================================================================== */
```

This comment style is used for commenting particular page elements. It has 1 empty line before it and no empty lines after it (it is immediately followed by the rules):

```
 /* Pager */

.pager {

 padding-bottom: 5px;

 border-bottom: 1px solid #ccc;

}
```

Use upper case for the first letter in comments:

```
 /* Correct */

/* Pager */

/* Wrong */

/* pager */
```

Practical example

An example of various conventions.

```
/**
 * @file
 * Layouts for this theme.
 */

/**
 * Column layout with horizontal scroll.
 *
 * This creates a single row of full-height, non-wrapping columns that can be
 * browsed horizontally within their parent.
 *
 * Example HTML:
 *
 * <div class="grid">
 *    <div class="cell cell-3"></div>
 *    <div class="cell cell-3"></div>
 *    <div class="cell cell-3"></div>
 * </div>
 */
```

```
/**
 * Grid container
 *
 * Must only contain '.cell' children.
 */
.grid {
   height: 100%;
  /* Remove inter-cell whitespace */
   font-size: 0;
  /* Prevent inline-block cells wrapping */
   white-space: nowrap;
}

/**
 * Grid cells
 *
 * No explicit width by default. Extend with '.cell-n' classes.
 */
.cell {
   position: relative;
   display: inline-block;
   overflow: hidden;
 box-sizing: border-box;
   height: 100%;
  /* Set the inter-cell spacing */
   padding: 0 10px;
   padding: 0.625rem;
   border: 2px solid #333;
   vertical-align: top;
  /* Reset white-space */
   white-space: normal;
  /* Reset font-size */
   font-size: 16px;
   font-size: 1rem;
}


/* Cell states */
.cell.is-animating {
   background-color: #fffdec;
}


/* Cell dimensions */
.cell-1 { width: 10%; }
```

```
.cell-2 { width: 20%; }
.cell-3 { width: 30%; }
.cell-4 { width: 40%; }
.cell-5 { width: 50%; }

/* Cell modifiers */
.cell--detail,
.cell--important {
  border-width: 4px;
}
```