

Introduction

This style guide acts as the official guide to follow in your projects. Udacity evaluators will use this guide to grade your projects. There are many opinions on the "ideal" style in the world of Front-End Web Development. Therefore, in order to reduce the confusion on what style students should follow during the course of their projects, we urge all students to refer to this style guide for their projects.

General Formatting Rules

Capitalization

Use only lowercase.

All code has to be lowercase. This applies to HTML element names, attributes, attribute values (unless text/CDATA).

Not Recommended:

```
<A HREF="/" />Home</A>
```

Recommended:

```
<a href="/">Home</a>
```

Trailing Whitespace

Remove trailing white spaces.

Trailing white spaces are unnecessary and can complicate diffs.

Not Recommended:

```
<p>What?</p> __
```

Recommended:

```
<p>What?</p>
```

If using Sublime Text, this can be done automatically each time you save a file by adding the following to your User Settings JSON file (you should be able to find this within Sublime Text's menu):

```
"trim_trailing_white_space_on_save": true
```

Indentation

Indentation should be consistent throughout the entire file. Whether you choose to use tabs or spaces, or 2-spaces vs. 4-spaces - just be consistent!

General Meta Rules

Encoding

Use UTF-8 (no BOM).

Make sure your editor uses UTF-8 as character encoding, without a byte order mark. Specify the encoding in HTML templates and documents with `<meta charset="utf-8">`.

Comments

Explain code as needed, where possible.

Use comments to explain code: What does it cover, what purpose does it serve, and why is the respective solution used or preferred?

Action Items

Mark todos and action items with `TODO:`.

Highlight todos by using the keyword `TODO` only, not other formats like `@@`. Append action items after a colon like this: `TODO: action item`.

Recommended:

```
<!-- TODO: add other fruits -->
```

```
<ul>
```

```
<li>Apples</li>
```

```
<li>Oranges</li>
```

```
</ul>
```

HTML Style Rules

Document Type

Use HTML5.

HTML5 (HTML syntax) is preferred for all HTML documents: `<!DOCTYPE html>`.

Do not close self-closing elements, ie. write `
`, not `
`.

HTML Validity

Use valid HTML.

Using valid HTML is a measurable baseline quality that ensures proper HTML usage and contributes to learning about technical requirements and constraints.

Not Recommended:

```
<title>Page Title</title>
```

```
<article>This is an article.
```

Recommended:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Page Title</title>
```

```
</head>
```

```
<body>
```

```
<article>This is an article.</article>
```

```
</body>
```

```
</html>
```

Semantics

Use HTML according to its purpose.

Use elements for what they have been created for. For example, use heading elements for headings, `p` elements for paragraphs, `a` elements for anchor, etc. Using HTML according to its purpose is important for accessibility, reuse and code efficiency reasons.

Not Recommended:

```
<div onclick="goToRecommendations();">All recommendations</div>
```

Recommended:

```
<a href="recommendations/">All recommendations</a>
```

Multimedia Fallback

Provide alternative contents for multimedia.

For multimedia, such as images, video, or animated objects via canvas, make sure to offer alternative access. For images that means use of meaningful alternative text and for video and audio transcripts and captions, if available.

Providing alternative contents is important for accessibility reasons. A blind user has few cues to tell what an image is about without the `alt` attributes, and other users may have no way of understanding what video or audio contents are about either.

For images whose alt attributes would introduce redundancy and for images whose purpose is purely decorative which you cannot immediately use CSS for, use no alternative text, as in `alt=""`.

Not Recommended:

```

```

Recommended:

```

```

Separation of Concerns

Separate structure from presentation from behavior.

Strictly keep structure (markup), presentation (styling), and behavior (scripting) apart, and try to keep the interaction between the three to an absolute minimum.

That is, make sure documents and templates contain only HTML and HTML that is solely serving structural purposes. Move everything presentational into style sheets, and everything behavioral into scripts. In addition, keep the contact area as small as possible by linking as few style sheets and scripts as possible from documents and templates.

Separating structure from presentation from behavior is important for maintenance reasons. It is almost always more expensive to change HTML documents and templates than it is to update style sheets and scripts.

Entity References

Do not use entity references.

There is no need to use entity references like `—`, `”`, or `☺`, assuming the same encoding (UTF-8) is used for files and editors as well as among teams.

The only exceptions apply to characters with special meaning in HTML (like `<` and `&`) as well as control or “invisible” characters (like no-break spaces).

Not Recommended:

```
The currency symbol for the Euro is &ldquo;&eur;&rdquo;.
```

Recommended:

```
The currency symbol for the Euro is "€".
```

type Attributes

Omit type attributes for style sheets and scripts.

Do not use type attributes for style sheets and scripts. Specifying type attributes in these contexts is not necessary as HTML implies `text/css` and `text/javascript` as defaults. This can be safely done even for older browsers

Not Recommended:

```
<link rel="stylesheet" href="css/style.css" type="text/css">
```

Recommended:

```
<link rel="stylesheet" href="css/style.css">
```

Not Recommended:

```
<script src="js/app.js" type="text/javascript"></script>
```

Recommended:

```
<script src="js/app.js"></script>
```

HTML Formatting Rules

General Formatting

Use a new line for every block, list or table element and indent every such child element.

Independent of the styling of an element (as CSS allows elements to assume a different role per display property), put every block, list or table element on a new line.

Also, indent them if they are child elements of a block, list or table element (if you run into issues around whitespace between list items it's acceptable to put all `li` elements in one line).

Recommended:

```
<blockquote>
```

```
<p><em>Space</em>, the final frontier.</p>
```

```
</blockquote>
```

```
<ul>
```

```
<li>Moe</li>
```

```
<li>Curry</li>
```

```
<li>Larry</li>
```

```
</ul>
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
<th scope="col">Income</th>
```

```
<th scope="col">Taxes</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>$5.00</td>
```

```
<td>$4.50</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

HTML Quotation Marks

When quoting attribute values, use double quotation marks.

Not Recommended:

```
<a href='login/' class='btn btn-secondary'>Login</a>
```

Recommended:

```
<a href="login/" class="btn btn-secondary">Login</a>
```