HTML 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

Write valid HTML

All HTML code must be valid and well formed. You must validate it against the HTML specification pertaining to the project you are working on. All documents must be using the HTML5 doctype and the <html> element should have a "lang"attribute. The <head> should also at a minimum include "viewport" and "charset" meta tags. Unless another specification is requested or needed, use HTML5 Document Type Definition:

```
<!DOCTYPE html>

<html lang="en">

 <head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title>Example Site</title>

 </head>

 <body></body>

</html>
```

Formatting

All HTML documents must use **two spaces** for indentation and there should be no trailing whitespace. HTML5 syntax must be used and all attributes must use double quotes around attributes.

```
<video autoplay="autoplay" poster="poster_image.jpg">

  <source src="foo.ogg" type="video/ogg">

</video>
```

HTML5 elements should be used where appropriate reserving <div> and <span> elements for situations where there is no semantic value (such as wrapping elements to provide styling hooks).

Lowercase names

Element and attribute names must be in all lower case:

```
<!-- Correct -->

<input name="name" type="text" />

<!-- Wrong -->

<input name="name" TYPE="text" />
```

Closing tags

Non-empty elements must have corresponding closing tags.

```
<h1>My title</h1>
<p>Some text</p>
```

Empty elements must be followed by a corresponding closing tag:

```
<span></span>
```

Elements with a single tag, such as HR, BR, INPUT, IMG must end with />:

```
<br/>
<hr/>
<img src="john.jpg" alt="John Doe" width="200" height="100"/>
```

Nested elements

Nested elements must be nested appropriately - for example:

```
<!-- Correct -->
<div>
 <p>Some text</p>
</div>
```

The <p> tag and its corresponding closing tag, </p>, are both nested inside the <div> and </div> tags.

If elements overlap they are not properly nested. This is illustrated in the following code:

```
<!-- Wrong -->
<div>
 <p>Some text</div>
</p>
```

Attribute values

Attribute values, even numeric attributes should be quoted—for example:

```
<!-- Correct -->
<input name="age" type="text" size="3" />

<!-- Wrong -->
<input name=age type=text size=3 />
```

Indentation

Use soft tabs with 2 spaces for code indentation.

Use indentation consistently to enhance the readability of the code.

When elements carry over more than one line of code, indent the contents of elements between the start tag and the end tag. This will make it easy to see where the element begins and ends.

Example:

```
<div class="container">
 <header class="header">
  <h1>Site Name<span></span></h1>
 </header>
 <!-- / header -->
 <hr>
 <nav class="navigation">
  <ul>
   <li><a href="#">Link</a></li>
   <li><a href="#">Link</a></li>
   <li><a href="#">Link</a></li>
   <li><a href="#">Link</a></li>
   <li><a href="#">Link</a></li>
  </ul>
 </nav>
 <!-- / navigation -->
</div>
```

```
<!-- / container -->
```

Encoding and charset

Set encoding of HTML document and its charset to UTF-8 Normalization Form C (NFC):

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

Special characters

Encode special characters, for example:

```
&amp;
&copy;
&raquo;
&gt;
```

HTML anchors

When you need to link to the section inside a HTML document use ID attribute:

```
<a href="#section">link</a>
<div id="section"></div>
```

If it isn't possible to use IDs (for example because of ASP.NET platform), use a named anchor:

```
<a href="#section">link</a>
<a name="section"></a>
```

Comments

Insert ending comment after closing tag of the HTML section in this format:

```
<!-- / name-of-class-or-id -->
```

Do not use starting comment.

Examples:

```
<ol class="accessibility-nav">
    <li><a href="#navigation">Skip to navigation</a></li>
    <li><a href="#content">Skip to content</a></li>
    <li><a href="#sidebar">Skip to sidebar</a></li>
</ol>
<!-- / accessibility-nav -->

<p>
    <a href="#" title="Go to homepage"><em>Home</em></a>
</p>
<!-- / breadcrumb →
```

Forms

Form fields must always include a <label> element with a "for" attribute matching the "id" on the input. This helps accessibility by focusing the input when the label is clicked, it also helps screen readers match labels to their respective inputs.

```
<label for="field-email">email</label>
<input type="email" id="field-email" name="email" value="" />
```

Each <input> should have an "id" that is unique to the page. It does not have to match the "name" attribute.

Forms should take advantage of the new HTML5 input types where they make sense to do so, placeholder attributes should also be included where relevant. Including these can provided enhancements in browsers that support them such as tailored inputs and keyboards.

```
<div>
    <label for="field-email">Email</label>
    <input type="email" id="field-email" name="email"
     value="name@example.com">
</div>
<div>
    <label for="field-phone">Phone</label>
    <input type="phone" id="field-phone" name="phone" value=""
     placeholder="+44 077 12345 678">
</div>
<div>
    <label for="field-url">Homepage</label>
    <input type="url" id="field-url" name="url" value=""
    placeholder="http://example.com">
</div>
```