



IE Masters in Finance

Specialize in FinTech and build a profile that stands out



HTML5 Style Guide and Coding Conventions

[< Previous](#)

[Next >](#)

HTML Coding Conventions

Web developers are often uncertain about the coding style and syntax to use in HTML.

Between 2000 and 2010, many web developers converted from HTML to XHTML.

With XHTML, developers were forced to write valid and "well-formed" code.

HTML5 is a bit more sloppy when it comes to code validation.

Be Smart and Future Proof

A consistent use of style makes it easier for others to understand your HTML.

In the future, programs like XML readers may want to read your HTML.

Using a well-formed-"close to XHTML" syntax can be smart.

Always keep your code tidy, clean and well-formed.

Use Correct Document Type

Always declare the document type as the first line in your document:

```
<!DOCTYPE html>
```

If you want consistency with lower case tags, you can use:

```
<!doctype html>
```

Use Lower Case Element Names

HTML5 allows mixing uppercase and lowercase letters in element names.

We recommend using lowercase element names because:

- Mixing uppercase and lowercase names is bad
- Developers normally use lowercase names (as in XHTML)
- Lowercase look cleaner
- Lowercase are easier to write

Bad:

```
<SECTION>  
  <p>This is a paragraph.</p>  
</SECTION>
```

Very Bad:

```
<Section>  
  <p>This is a paragraph.</p>  
</SECTION>
```

Good:

```
<section>  
  <p>This is a paragraph.</p>  
</section>
```

Cursos, Diplomado Programación

PHP, JAVA, Python, HTML5

Certificado por el Ministerio de Educación, Aprende a programar desde 0

uneweb.edu.ve

ABRIR

Close All HTML Elements

In HTML5, you don't have to close all elements (for example the `<p>` element).

We recommend closing all HTML elements.

Bad:

```
<section>
  <p>This is a paragraph.
  <p>This is a paragraph.
</section>
```

Good:

```
<section>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
</section>
```

Close Empty HTML Elements

In HTML5, it is optional to close empty elements.

Allowed:

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

Also Allowed:

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

However, the closing slash (/) is REQUIRED in XHTML and XML.

If you expect XML software to access your page, it is a good idea to keep the closing slash!

Use Lower Case Attribute Names

HTML5 allows mixing uppercase and lowercase letters in attribute names.

We recommend using lowercase attribute names because:

- Mixing uppercase and lowercase names is bad
- Developers normally use lowercase names (as in XHTML)
- Lowercase look cleaner
- Lowercase are easier to write

Bad:

```
<div CLASS="menu">
```

Good:

```
<div class="menu">
```

Quote Attribute Values

HTML5 allows attribute values without quotes.

We recommend quoting attribute values because:

- Developers normally quote attribute values (as in XHTML)
- Quoted values are easier to read
- You MUST use quotes if the value contains spaces

Very bad:

This will not work, because the value contains spaces:

```
<table class=table striped>
```

Bad:

```
<table class=striped>
```

Good:

```
<table class="striped">
```

Image Attributes

Always add the `alt` attribute to images. This attribute is important when the image for some reason cannot be displayed. Also, always define image width and height. It reduces flickering because the browser can reserve space for the image before loading.

Bad:

```

```

Good:

```

```

Spaces and Equal Signs

HTML5 allows spaces around equal signs. But space-less is easier to read and groups entities better together.

Bad:

```
<link rel = "stylesheet" href = "styles.css">
```

Good:

```
<link rel="stylesheet" href="styles.css">
```

Avoid Long Code Lines

When using an HTML editor, it is inconvenient to scroll right and left to read the HTML code.

Try to avoid code lines longer than 80 characters.

Blank Lines and Indentation

Do not add blank lines without a reason.

For readability, add blank lines to separate large or logical code blocks.

For readability, add two spaces of indentation. Do not use the tab key.

Do not use unnecessary blank lines and indentation. It is not necessary to indent every element:

Unnecessary:

```
<body>

  <h1>Famous Cities</h1>
```

```
<h2>Tokyo</h2>

<p>
  Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
  and the most populous metropolitan area in the world.
  It is the seat of the Japanese government and the Imperial Palace,
  and the home of the Japanese Imperial Family.
</p>

</body>
```

Better:

```
<body>

<h1>Famous Cities</h1>

<h2>Tokyo</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.
It is the seat of the Japanese government and the Imperial Palace,
and the home of the Japanese Imperial Family.</p>

</body>
```

Table Example:

```
<table>
  <tr>
    <th>Name</th>
    <th>Description</th>
  </tr>
  <tr>
    <td>A</td>
    <td>Description of A</td>
  </tr>
  <tr>
    <td>B</td>
    <td>Description of B</td>
  </tr>
</table>
```

List Example:

```
<ul>
  <li>London</li>
```

```
<li>Paris</li>
<li>Tokyo</li>
</ul>
```

Omitting <html> and <body>?

In HTML5, the `<html>` tag and the `<body>` tag can be omitted.

The following code will validate as HTML5:

Example

```
<!DOCTYPE html>
<head>
  <title>Page Title</title>
</head>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

Try it Yourself »

However, we do not recommend omitting the `<html>` and the `<body>` tag.

The `<html>` element is the document root. It is the recommended place for specifying the page language:

```
<!DOCTYPE html>
<html lang="en-US">
```

Declaring a language is important for accessibility applications (screen readers) and search engines.

Omitting `<html>` or `<body>` can crash DOM and XML software.

Omitting `<body>` can produce errors in older browsers (IE9).

Omitting <head>?

In HTML5, the `<head>` tag can also be omitted.

By default, browsers will add all elements before `<body>` to a default `<head>` element.

You can reduce the complexity of HTML by omitting the `<head>` tag:

Example

```
<!DOCTYPE html>
```

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

<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>

</html>
```

Try it Yourself »

However, we do not recommend omitting the `<head>` tag.

Omitting tags is unfamiliar to web developers. It needs time to be established as a guideline.

Meta Data

The `<title>` element is required in HTML5. Make the title as meaningful as possible:

```
<title>HTML5 Syntax and Coding Style</title>
```

To ensure proper interpretation and correct search engine indexing, both the language and the character encoding should be defined as early as possible in a document:

```
<!DOCTYPE html>
<html lang="en-US">
<head>
  <meta charset="UTF-8">
  <title>HTML5 Syntax and Coding Style</title>
</head>
```

Setting The Viewport

HTML5 introduced a method to let web designers take control over the viewport, through the `<meta>` tag.

The viewport is the user's visible area of a web page. It varies with the device, and will be smaller on a mobile phone than on a computer screen.

You should include the following `<meta>` viewport element in all your web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

A `<meta>` viewport element gives the browser instructions on how to control the page's dimensions and scaling.

The `width=device-width` part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The `initial-scale=1.0` part sets the initial zoom level when the page is first loaded by the browser.

Here is an example of a web page *without* the viewport meta tag, and the same web page *with* the viewport meta tag:

Tip: If you are browsing this page with a phone or a tablet, you can click on the two links below to see the difference.



Without the viewport meta tag



With the viewport meta tag

HTML Comments

Short comments should be written on one line, like this:

```
<!-- This is a comment -->
```

Comments that spans more than one line, should be written like this:

```
<!--  
  This is a long comment example. This is a long comment example.  
  This is a long comment example. This is a long comment example.  
-->
```

Long comments are easier to observe if they are indented two spaces.

Style Sheets

Use simple syntax for linking to style sheets (the `type` attribute is not necessary):

```
<link rel="stylesheet" href="styles.css">
```

Short rules can be written compressed, like this:

```
p.intro {font-family: Verdana; font-size: 16em;}
```

Long rules should be written over multiple lines:

```
body {  
  background-color: lightgrey;  
  font-family: "Arial Black", Helvetica, sans-serif;  
  font-size: 16em;  
  color: black;  
}
```

- Place the opening bracket on the same line as the selector
- Use one space before the opening bracket
- Use two spaces of indentation
- Use semicolon after each property-value pair, including the last
- Only use quotes around values if the value contains spaces
- Place the closing bracket on a new line, without leading spaces
- Avoid lines over 80 characters

Loading JavaScript in HTML

Use simple syntax for loading external scripts (the type attribute is not necessary):

```
<script src="myscript.js">
```

Accessing HTML Elements with JavaScript

A consequence of using "untidy" HTML styles can result in JavaScript errors.

These two JavaScript statements will produce different results:

Example

```
var obj = getElementById("Demo")  
  
var obj = getElementById("demo")
```

Try it Yourself »

[Visit the JavaScript Style Guide.](#)

Use Lower Case File Names

Some web servers (Apache, Unix) are case sensitive about file names: "london.jpg" cannot be accessed as "London.jpg".

Other web servers (Microsoft, IIS) are not case sensitive: "london.jpg" can be accessed as "London.jpg" or "london.jpg".

If you use a mix of upper and lower case, you have to be extremely consistent.

If you move from a case insensitive to a case sensitive server, even small errors will break your web!

To avoid these problems, always use lower case file names.

File Extensions

HTML files should have a **.html** or **.htm** extension.

CSS files should have a **.css** extension.

JavaScript files should have a **.js** extension.

Differences Between .htm and .html

There is no difference between the .htm and .html extensions. Both will be treated as HTML by any web browser or web server.

The differences are cultural:

.htm "smells" of early DOS systems where the system limited the extensions to 3 characters.

.html "smells" of Unix operating systems that did not have this limitation.

Technical Differences

When a URL does not specify a filename (like <https://www.w3schools.com/css/>), the server returns a default filename. Common default filenames are index.html, index.htm, default.html and default.htm.

If your server is configured only with "index.html" as default filename, your file must be named "index.html", not "index.htm."

However, servers can be configured with more than one default filename, and normally you can set up as many default filenames as needed.

Anyway, the full extension for HTML files is .html, and there's no reason it should not be used.

[< Previous](#)

[Next >](#)