

HTML



HTML 5

Web Application Development



AGENDA

- HTML 5 Intro
- HTML 5 Support
- HTML 5 New Elements
- HTML 5 Semantics
- HTML 5 Migration
- HTML 5 Style Guide



HTML 5 INTRO

Web App Development

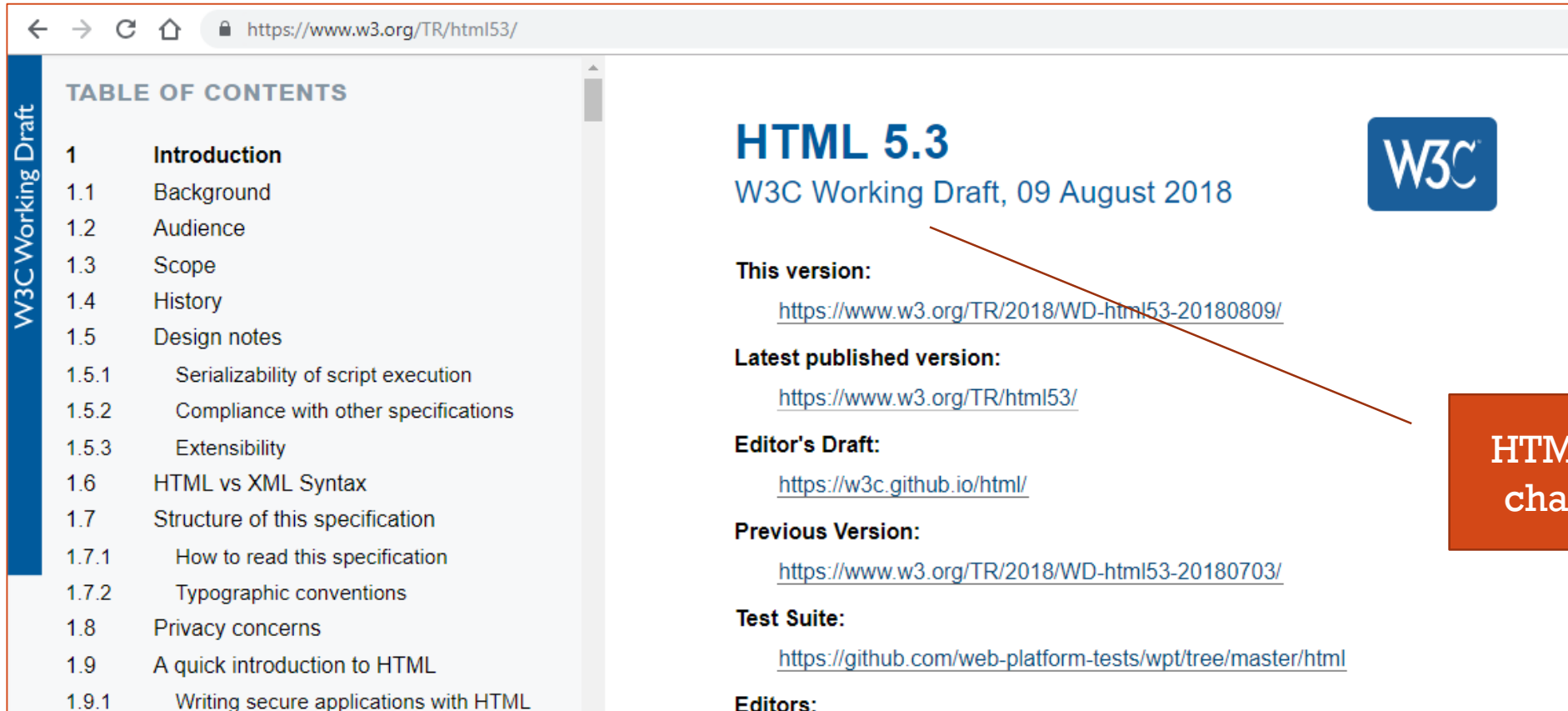
WHAT IS HTML 5?

- **HTML 5** (formerly spelled **HTML 5**) is a markup language used for structuring and presenting content on the World Wide Web.
- It is the fifth and current major version of the HTML standard, and subsumes XHTML.
- It currently exists in two standardized forms:
 - *HTML 5.2 Recommendation*^[4] by the World Wide Web Consortium (W3C, a broad coalition of organizations), intended primarily for Web content developers; and
 - *HTML Living Standard*^[5] by WHATWG (a small consortium of four browser vendors), intended primarily for browser developers, though it also exists in an abridged Web developer version.^[6]
 - There are minor conflicts between the two groups' specifications.

WHY HTML 5?

- HTML 5 was first released in public-facing form on 22 January 2008, with a major update and "W3C Recommendation" status in October 2014.
- Its goals are
 - to improve the language with support for the latest multimedia and other new features;
 - to keep the language both easily readable by humans and consistently understood by computers and devices such as Web browsers, parsers, etc., without XHTML's rigidity; and
 - to remain backward-compatible with older software.
- HTML 5 is intended to subsume not only HTML 4, but also XHTML 1 and DOM Level 2 HTML; the HTML 4 and XHTML specs were announced as *superseded* by HTML 5.2 on **27 March 2018**.

“HTML 5.2 IS DONE, HTML 5.3 IS COMING”



The screenshot shows the W3C Working Draft page for HTML 5.3. On the left is a vertical sidebar with the text 'W3C Working Draft' and a 'TABLE OF CONTENTS' section listing various topics from Introduction to Writing secure applications with HTML. The main content area features the title 'HTML 5.3' and the date 'W3C Working Draft, 09 August 2018' next to the W3C logo. Below this, it lists links for 'This version', 'Latest published version', 'Editor's Draft', 'Previous Version', 'Test Suite', and 'Editors'. A red arrow points from the 'Latest published version' link to an orange callout box on the right.

W3C Working Draft

TABLE OF CONTENTS

- 1 Introduction
 - 1.1 Background
 - 1.2 Audience
 - 1.3 Scope
 - 1.4 History
 - 1.5 Design notes
 - 1.5.1 Serializability of script execution
 - 1.5.2 Compliance with other specifications
 - 1.5.3 Extensibility
 - 1.6 HTML vs XML Syntax
 - 1.7 Structure of this specification
 - 1.7.1 How to read this specification
 - 1.7.2 Typographic conventions
 - 1.8 Privacy concerns
 - 1.9 A quick introduction to HTML
 - 1.9.1 Writing secure applications with HTML

HTML 5.3
W3C Working Draft, 09 August 2018

This version:
<https://www.w3.org/TR/2018/WD-html53-20180809/>

Latest published version:
<https://www.w3.org/TR/html53/>

Editor's Draft:
<https://w3c.github.io/html/>

Previous Version:
<https://www.w3.org/TR/2018/WD-html53-20180703/>

Test Suite:
<https://github.com/web-platform-tests/wpt/tree/master/html>

Editors:

W3C

HTML specifications
change frequently!

Source: <https://www.w3.org/TR/HTML53/>

HTML 5 INTRO

What is New in HTML 5?

- The DOCTYPE declaration for HTML 5 is very simple:
 - `<!DOCTYPE html>`
- The character encoding (charset) declaration is also very simple:
 - `<meta charset="UTF-8">`
 - The default character encoding in HTML 5 is UTF-8.

Example

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
</head>

<body>
  Content of the document.....
</body>

</html>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_intro

HTML 5 INTRO

New HTML 5 Elements

- The most interesting new HTML 5 elements are:
- New semantic elements like `<header>`, `<footer>`, `<article>`, and `<section>`.
- New attributes of form elements like `number`, `date`, `time`, `calendar`, and `range`.
- New graphic elements: `<svg>` and `<canvas>`.
- New multimedia elements: `<audio>` and `<video>`.

HTML 5 INTRO

New HTML 5 API's (Application Programming Interfaces)

- The most interesting new API's in HTML 5 are:
 - HTML Geolocation
 - HTML Drag and Drop
 - HTML Local Storage
 - HTML Application Cache
 - HTML Web Workers
 - HTML SSE (Server Sent Events)
- **Tip:** HTML Local storage is a powerful replacement for cookies.

HTML 5 INTRO

Removed Elements in HTML 5

- The following HTML4 elements have been removed in HTML 5:

Removed Element	Use Instead
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	
	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS, <s>, or
<tt>	CSS

HTML 5 INTRO

HTML History

- Since the early days of the World Wide Web, there have been many versions of HTML:
- From 1991 to 1999, HTML developed from version 1 to version 4.
- In year 2000, the World Wide Web Consortium (W3C) recommended XHTML 1.0. The XHTML syntax was strict, and the developers were forced to write valid and "well-formed" code.
- In 2004, W3C's decided to close down the development of HTML, in favor of XHTML.
- In 2004, WHATWG (Web Hypertext Application Technology Working Group) was formed. The WHATWG wanted to develop HTML, consistent with how the web was used, while being backward compatible with older versions of HTML.
- In 2004 - 2006, the WHATWG gained support by the major browser vendors.
- In 2006, W3C announced that they would support WHATWG.
- In 2008, the first HTML 5 public draft was released.
- In 2012, WHATWG and W3C decided on a separation:
- **WHATWG wanted to develop HTML as a "Living Standard"**. A living standard is always updated and improved. New features can be added, but old functionality cannot be removed.
- The [WHATWG HTML 5 Living Standard](#) was published in 2012, and is continuously updated.
- **W3C wanted to develop a definitive HTML 5 and XHTML standard.**
- The [W3C HTML 5](#) Recommendation was released 28 October 2014.
- The [W3C HTML 5.1 2nd Edition](#) Recommendation was released 3 October 2017.
- The [W3C HTML 5.2](#) Recommendation was released 14 December 2017.

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML 5 First Public Draft
2012	WHATWG HTML 5 Living Standard
2014	W3C Recommendation: HTML 5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML 5.1 2nd Edition
2017	W3C Recommendation: HTML 5.2



HTML 5 BROWSER SUPPORT

Web App Development

HTML BROWSER

HTML 5 Browser Support

We will assume modern browsers, but on the job you might not have that luxury!

- HTML 5 is supported in all modern browsers.
- In addition, all browsers, old and new, automatically handle unrecognized elements as inline elements.
- Because of this, you can "teach" older browsers to handle "unknown" HTML elements.
- You can even teach IE6 (Windows XP 2001) how to handle unknown HTML elements.

HTML BROWSER

Define Semantic Elements as Block Elements

- HTML 5 defines eight new **semantic** elements. All these are **block-level** elements.
- To secure correct behavior in older browsers, you can set the CSS **display** property for these HTML elements to **block**:

Example

```
header, section, footer, aside, nav, main,  
article, figure {  
    display: block;  
}
```

HTML BROWSER

Add New Elements to HTML

- You can also add new elements to an HTML page with a browser trick.
- This example adds a new element called `<myHero>` to an HTML page, and defines a style for it:
- The JavaScript statement `document.createElement("myHero")` is needed to create a new element in IE 9, and earlier.

Example

```
<!DOCTYPE html>
<html>
<head>
<script>document.createElement("myHero")</script>
<style>
myHero {
    display: block;
    background-color: #dddddd;
    padding: 50px;
    font-size: 30px;
}
</style>
</head>
<body>

<h1>A Heading</h1>
<myHero>My Hero Element</myHero>

</body>
</html>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_browsers_myhero

HTML BROWSER

Problem With Internet Explorer 8

- You could use the solution described above for all new HTML 5 elements.
- However, **IE8 (and earlier) does not allow styling of unknown elements!**
- Thankfully, Sjoerd Visscher created the HTML5SHIV! The HTML5SHIV is a JavaScript workaround to enable styling of HTML 5 elements in versions of Internet Explorer prior to version 9.
- You will require the HTML5SHIV to provide compatibility for IE Browsers older than IE 9.

HTML BROWSER

Syntax For HTML5SHIV

- The HTML5Shiv is placed within the `<head>` tag.
- The HTML5Shiv is a javascript file that is referenced in a `<script>` tag.
- You should use the HTML5SHIV when you are using the new HTML 5 elements such as: `<article>`, `<section>`, `<aside>`, `<nav>`, `<footer>`.
- You can download the latest version of HTML5shiv from github or reference the CDN version at <https://oss.maxcdn.com/libs/HTML5SHIV/3.7.0/HTML5SHIV.js>

Example

```
<head>  
  <!--[if lt IE 9]>  
    <script src="/js/HTML5SHIV.js"></script>  
  <![endif]-->  
</head>
```

Example

HTML BROWSER

HTML5SHIV Example

- If you do not want to download and store the HTML5SHIV on your site, you could reference the version found on the CDN site.
- The HTML5SHIV script must be placed in the `<head>` element, after any stylesheets:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<!--[if lt IE 9]>
  <script
src="https://oss.maxcdn.com/libs/HTML5SHIV/3.7.0/HTML5SHIV.js"></script>
<![endif]-->
</head>
<body>

<section>

<h1>Famous Cities</h1>

<article>
<h2>London</h2>
<p>London is the capital city of England. It is the most populous city in the
United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
</article>

<article>
<h2>Paris</h2>
<p>Paris is the capital and most populous city of France.</p>
</article>

<article>
<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.</p>
</article>

</section>

</body>
</html>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_browsers_HTML5SHIV





HTML 5 NEW ELEMENTS

Web App Development

HTML 5 NEW ELEMENTS

New Semantic/Structural Elements

BDI stands for
Bi-Directional
Isolation

Tag	Description
<u><article></u>	Defines an article in a document
<u><aside></u>	Defines content aside from the page content
<u><bdi></u>	Isolates a part of text that might be formatted in a different direction from other text outside it
<u><details></u>	Defines additional details that the user can view or hide
<u><dialog></u>	Defines a dialog box or window
<u><figcaption></u>	Defines a caption for a <figure> element
<u><figure></u>	Defines self-contained content
<u><footer></u>	Defines a footer for a document or section
<u><header></u>	Defines a header for a document or section

More

HTML 5 NEW ELEMENTS

New Semantic/Structural Elements - CONTINUED

Tag	Description
<u><main></u>	Defines the main content of a document
<u><mark></u>	Defines marked/highlighted text
<u><meter></u>	Defines a scalar measurement within a known range (a gauge)
<u><nav></u>	Defines navigation links
<u><progress></u>	Represents the progress of a task
<u><rp></u>	Defines what to show in browsers that do not support ruby annotations
<u><rt></u>	Defines an explanation/pronunciation of characters (for East Asian typography)
<u><ruby></u>	Defines a ruby annotation (for East Asian typography)
<u><section></u>	Defines a section in a document
<u><summary></u>	Defines a visible heading for a <details> element
<u><time></u>	Defines a date/time
<u><wbr></u>	Defines a possible line-break

HTML 5 NEW ELEMENTS

New Form Elements

Tag	Description
<u><datalist></u>	Specifies a list of pre-defined options for input controls
<u><output></u>	Defines the result of a calculation

HTML 5 NEW ELEMENTS

New Input Types

New Input Types	New Input Attributes
<ul style="list-style-type: none">•color•date•datetime•datetime-local•email•month•number•range•search•tel•time•url•week	<ul style="list-style-type: none">•autocomplete•autofocus•form•formaction•formenctype•formmethod•formnovalidate•formtarget•height and width•list•min and max•multiple•pattern (regexp)•placeholder•required•step

HTML 5 NEW ELEMENTS

HTML 5 - New Attribute Syntax

- HTML 5 allows four different syntaxes for attributes.
- This example demonstrates the different syntaxes used in an `<input>` tag:
- In HTML 5, all four syntaxes may be used, depending on what is needed for the attribute.

Type	Example
Empty	<code><input type="text" value="John" disabled></code>
Unquoted	<code><input type="text" value=John></code>
Double-quoted	<code><input type="text" value="John Doe"></code>
Single-quoted	<code><input type="text" value='John Doe'></code>

HTML 5 NEW ELEMENTS

New Media Elements

Tag	Description
<u><audio></u>	Defines sound content
<u><embed></u>	Defines a container for an external (non-HTML) application
<u><source></u>	Defines multiple media resources for media elements (<video> and <audio>)
<u><track></u>	Defines text tracks for media elements (<video> and <audio>)
<u><video></u>	Defines video or movie



HTML 5 SEMANTIC ELEMENTS

Web App Development

HTML 5 SEMANTIC ELEMENTS






What are Semantic Elements?

- A semantic element clearly describes its **meaning** to both the browser and the developer.
 - Examples of non-semantic elements: `<div>` and `` - Tells nothing about its content.
 - Examples of semantic elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

HTML 5 SEMANTIC ELEMENTS

Browser Support

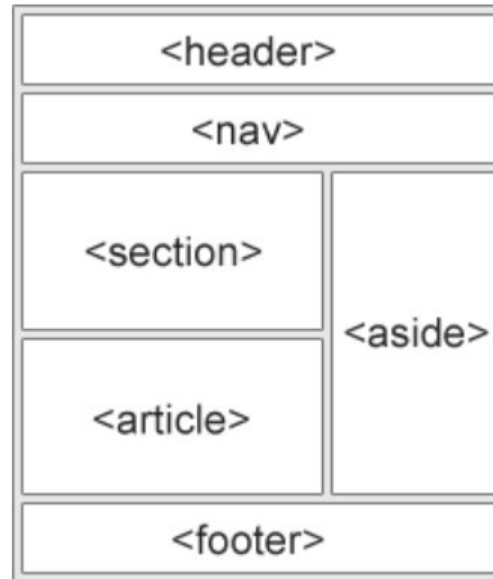
- HTML 5 semantic elements are supported in all modern browsers.
- In addition, you can "teach" older browsers how to handle "unknown elements".

				
Yes	Yes	Yes	Yes	Yes

HTML 5 SEMANTIC ELEMENTS

New Semantic Elements in HTML 5

- Many web sites contain HTML code like: `<div id="nav">` `<div class="header">` `<div id="footer">` to indicate navigation, header, and footer
- HTML 5 offers new semantic elements to define different parts of a web page:
 - `<article>`
 - `<aside>`
 - `<details>`
 - `<figcaption>`
 - `<figure>`
 - `<footer>`
 - `<header>`
 - `<main>`
 - `<mark>`
 - `<nav>`
 - `<section>`
 - `<summary>`
 - `<time>`



HTML 5 SEMANTIC ELEMENTS

HTML 5 <section> Element

- The <section> element defines a section in a document.
- According to W3C's HTML 5 documentation: "A section is a thematic grouping of content, typically with a heading."
- A home page could normally be split into sections for introduction, content, and contact information.

Example

```
<section>  
  <h1>WWF</h1>  
  <p>The World Wide Fund for Nature (WWF)  
is....</p>  
</section>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_section

HTML 5 SEMANTIC ELEMENTS

HTML 5 <article> Element

- The <article> element specifies independent, self-contained content.
- An article should make sense on its own, and it should be possible to read it independently from the rest of the web site.
- Examples of where an <article> element can be used:
 - Forum post
 - Blog post
 - Newspaper article

Example

```
<article>  
  <h1>What Does WWF Do?</h1>  
  <p>WWF's mission is to stop the degradation  
    of our planet's natural environment,  
    and build a future in which humans live in  
    harmony with nature.</p>  
</article>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_article

HTML 5 SEMANTIC ELEMENTS

Nesting `<article>` in `<section>` or Vice Versa?

- The `<article>` element specifies independent, self-contained content.
- The `<section>` element defines section in a document.
- **Can we use the definitions to decide how to nest those elements? No, we cannot!**
- So, on the Internet, you will find HTML pages with `<section>` elements containing `<article>` elements, and `<article>` elements containing `<section>` elements.
- You will also find pages with `<section>` elements containing `<section>` elements, and `<article>` elements containing `<article>` elements.
- Example for a newspaper: The sport `<article>` in the sport section, may have a technical section in each `<article>`.

HTML 5 SEMANTIC ELEMENTS

HTML 5 <header> Element

- The <header> element specifies a header for a document or section.
- The <header> element should be used as a container for introductory content.
- You can have several <header> elements in one document.
- The following example defines a header for an article:

Example

```
<article>
  <header>
    <h1>What Does WWF Do?</h1>
    <p>WWF's mission:</p>
  </header>
  <p>WWF's mission is to stop the degradation
of our planet's natural environment,
and build a future in which humans live in
harmony with nature.</p>
</article>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_header

HTML 5 SEMANTIC ELEMENTS

HTML 5 <footer> Element

- The <footer> element specifies a footer for a document or section.
- A <footer> element should contain information about its containing element.
- A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
- You may have several <footer> elements in one document.

Example

```
<footer>  
  <p>Posted by: Hege Refsnes</p>  
  <p>Contact information: <a  
href="mailto:someone@example.com">  
  someone@example.com</a>.</p>  
</footer>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_footer

HTML 5 SEMANTIC ELEMENTS

HTML 5 <nav> Element

- The <nav> element defines a set of navigation links.
- Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.

Example

```
<nav>  
  <a href="/html/">HTML</a> |  
  <a href="/css/">CSS</a> |  
  <a href="/js/">JavaScript</a> |  
  <a href="/jquery/">jQuery</a>  
</nav>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_nav

HTML 5 SEMANTIC ELEMENTS

HTML 5 <aside> Element

- The <aside> element defines some content aside from the content it is placed in (like a sidebar).
- The <aside> content should be related to the surrounding content.

Example

```
<p>My family and I visited The Epcot center  
this summer.</p>
```

```
<aside>  
  <h4>Epcot Center</h4>  
  <p>The Epcot Center is a theme park in Disney  
World, Florida.</p>  
</aside>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_aside

HTML 5 SEMANTIC ELEMENTS

HTML 5 <figure> and <figcaption> Elements

- The purpose of a figure caption is to add a visual explanation to an image.
- In HTML 5, an image and a caption can be grouped together in a <figure> element:
- The element defines the image, the <figcaption> element defines the caption.

Example

```
<figure>  
    
  <figcaption>Fig1. - Trulli, Puglia,  
Italy.</figcaption>  
</figure>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_figcaption

HTML 5 SEMANTIC ELEMENTS

Why Semantic Elements?

- With HTML4, developers used their own id/class names to style elements: header, top, bottom, footer, menu, navigation, main, container, content, article, sidebar, topnav, etc.
- This made it impossible for search engines to identify the correct web page content.
- With the new HTML 5 elements (<header> <footer> <nav> <section> <article>), this will become easier.
- According to the W3C, a Semantic Web: "Allows data to be shared and reused across applications, enterprises, and communities."

HTML 5 SEMANTIC ELEMENTS

Semantic Elements in HTML 5

- Below is an alphabetical list of the new semantic elements in HTML 5.
- The links go to our complete HTML 5 Reference.

Tag	Description
<u><article></u>	Defines an article
<u><aside></u>	Defines content aside from the page content
<u><details></u>	Defines additional details that the user can view or hide
<u><figcaption></u>	Defines a caption for a <figure> element
<u><figure></u>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
<u><footer></u>	Defines a footer for a document or section
<u><header></u>	Specifies a header for a document or section
<u><main></u>	Specifies the main content of a document
<u><mark></u>	Defines marked/highlighted text
<u><nav></u>	Defines navigation links
<u><section></u>	Defines a section in a document
<u><summary></u>	Defines a visible heading for a <details> element
<u><time></u>	Defines a date/time



HTML 5 MIGRATION

Web App Development

HTML 5 MIGRATION

Migration from HTML4 to HTML 5

- This chapter is entirely about how to migrate from HTML4 to HTML 5.
- This chapter demonstrates how to convert an HTML4 page into an HTML 5 page, without destroying anything of the original content or structure.
- You can migrate from XHTML to HTML 5, using the same recipe.

Typical HTML4	Typical HTML5
<code><div id="header"></code>	<code><header></code>
<code><div id="menu"></code>	<code><nav></code>
<code><div id="content"></code>	<code><section></code>
<code><div class="article"></code>	<code><article></code>
<code><div id="footer"></code>	<code><footer></code>

HTML 5 MIGRATION

A Typical HTML4 Page

Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>HTML4</title>
<style>
body {
    font-family: Verdana,sans-serif;
    font-size: 0.9em;
}

div#header, div#footer {
    padding: 10px;
    color: white;
    background-color: black;
}

div#content {
    margin: 5px;
    padding: 10px;
    background-color: lightgrey;
}

div.article {
    margin: 5px;
    padding: 10px;
    background-color: white;
}
```

```
div#menu ul {
    padding: 0;
}

div#menu ul li {
    display: inline;
    margin: 5px;
}
</style>
</head>
<body>
```

```
<div id="header">
    <h1>Monday Times</h1>
</div>
```

```
<div id="menu">
    <ul>
        <li>News</li>
        <li>Sports</li>
        <li>Weather</li>
    </ul>
</div>
```

```
<div id="content">
    <h2>News Section</h2>
    <div class="article">
        <h2>News Article</h2>
        <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Pellentesque in porta lorem. Morbi
condimentum est nibh, et consectetur tortor feugiat
at.</p>
    </div>
    <div class="article">
        <h2>News Article</h2>
        <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Pellentesque in porta lorem. Morbi
condimentum est nibh, et consectetur tortor feugiat
at.</p>
    </div>
</div>
```

```
<div id="footer">
    <p>&copy; 2016 Monday Times. All rights
reserved.</p>
</div>

</body>
</html>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_migrate_1

HTML 5 MIGRATION

Change to HTML 5 Doctype

- Change the **doctype**:
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
- to the HTML 5 doctype:

Example

`<!DOCTYPE html>`

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_migrate_2

HTML 5 MIGRATION

Change to HTML 5 Encoding

- Change the **encoding** information:
- `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- to HTML 5 encoding:

Example

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

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_migrate_3

HTML 5 MIGRATION

Add The HTML5SHIV

- The new HTML 5 semantic elements are supported in all modern browsers. In addition, you can "teach" older browsers how to handle "unknown elements".
- However, IE8 and earlier, does not allow styling of unknown elements. So, the HTML5SHIV is a JavaScript workaround to enable styling of HTML 5 elements in versions of Internet Explorer prior to version 9.
- Add the HTML5SHIV:

Example

```
<!--[if lt IE 9]>  
  <script  
src="https://oss.maxcdn.com/libs/HTML5SHIV/3.7.  
0/HTML5SHIV.js"></script>  
<![endif]-->
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_migrate_4

HTML 5 MIGRATION

Change to HTML 5 Semantic Elements

- The existing CSS contains id's and classes for styling the elements:

```
body {  
    font-family: Verdana,sans-serif;  
    font-size: 0.9em;  
}  
  
div#header, div#footer {  
    padding: 10px;  
    color: white;  
    background-color: black;  
}  
  
div#content {  
    margin: 5px;  
    padding: 10px;  
    background-color: lightgrey;  
}  
  
div.article {  
    margin: 5px;  
    padding: 10px;  
    background-color: white;  
}  
  
div#menu ul {  
    padding: 0;  
}  
  
div#menu ul li {  
    display: inline;  
    margin: 5px;
```

HTML 5 MIGRATION

Change to HTML 5 Semantic Elements

- Replace with equal CSS styles for HTML 5 semantic elements:

```
body {  
    font-family: Verdana,sans-serif;  
    font-size: 0.9em;  
}  
  
header, footer {  
    padding: 10px;  
    color: white;  
    background-color: black;  
}  
  
section {  
    margin: 5px;  
    padding: 10px;  
    background-color: lightgrey;  
}  
  
article {  
    margin: 5px;  
    padding: 10px;  
    background-color: white;  
}  
  
nav ul {  
    padding: 0;  
}  
  
nav ul li {  
    display: inline;  
    margin: 5px;  
}
```

HTML 5 MIGRATION

Change to HTML 5 Semantic Elements

- Finally, change the elements to HTML 5 semantic elements:

```
<body>

<header>
<h1>Monday Times</h1>
</header>

<nav>
<ul>
<li>News</li>
<li>Sports</li>
<li>Weather</li>
</ul>
</nav>

<section>
<h2>News Section</h2>
<article>
<h2>News Article</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Pellentesque in porta lorem. Morbi condimentum est
nibh, et consectetur tortor feugiat at.</p>
</article>
<article>
<h2>News Article</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Pellentesque in porta lorem. Morbi condimentum est
nibh, et consectetur tortor feugiat at.</p>
</article>
</section>

<footer>
<p>&copy; 2014 Monday Times. All rights reserved.</p>
</footer>

</body>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_migrate_5

HTML 5 MIGRATION

The Difference Between <article> <section> and <div>

- There is a confusing (lack of) difference in the HTML 5 standard, between <article> <section> and <div>.
- In the HTML 5 standard, the <section> element is defined as a block of related elements.
- The <article> element is defined as a complete, self-contained block of related elements.
- The <div> element is defined as a block of children elements.
- How to interpret that?
- In the example above, we have used <section> as a container for related <articles>.
- But, we could have used <article> as a container for articles as well.
- Here are some different examples:

Example <article> in <article>:

```
<article>
```

```
<h2>Famous Cities</h2>
```

```
<article>
```

```
<h2>London</h2>
```

```
<p>London is the capital city of England. It is  
the most populous city in the United Kingdom,  
with a metropolitan area of over 13 million  
inhabitants.</p>
```

```
</article>
```

```
<article>
```

```
<h2>Paris</h2>
```

```
<p>Paris is the capital and most populous city  
of France.</p>
```

```
</article>
```

```
<article>
```

```
<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.</p>
```

```
</article>
```

```
</article>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_migrate_6

HTML 5 MIGRATION

The Difference Between <article> <section> and <div>

Example <div> in <article>:

```
<article>

<h2>Famous Cities</h2>

<div class="city">
<h2>London</h2>
<p>London is the capital city of England. It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>
</div>

<div class="city">
<h2>Paris</h2>
<p>Paris is the capital and most populous city of France.</p>
</div>

<div class="city">
<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.</p>
</div>

</article>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_migrate_7

HTML 5 MIGRATION

The Difference Between `<article>` `<section>` and `<div>`

Example `<div>` in `<section>` in `<article>`

```
<article>

<section>
<h2>Famous Cities</h2>

<div class="city">
<h2>London</h2>
<p>London is the capital city of England. It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>
</div>

<div class="city">
<h2>Paris</h2>
<p>Paris is the capital and most populous city of France.</p>
</div>

<div class="city">
<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.</p>
</div>
</section>

<section>
<h2>Famous Countries</h2>

<div class="country">
<h2>England</h2>
<p>London is the capital city of England. It is the most populous city in the United Kingdom,
with a metropolitan area of over 13 million inhabitants.</p>
</div>

<div class="country">
<h2>France</h2>
<p>Paris is the capital and most populous city of France.</p>
</div>

<div class="country">
<h2>Japan</h2>
<p>Tokyo is the capital of Japan, the center of the Greater Tokyo Area,
and the most populous metropolitan area in the world.</p>
</div>
</section>

</article>
```

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryHTML5_migrate_8



HTML 5 STYLE GUIDE

Web App Development

HTML 5 STYLE GUIDE

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.
- HTML 5 is a bit more sloppy when it comes to code validation.

HTML 5 STYLE GUIDE

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.

HTML 5 STYLE GUIDE

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>`

HTML 5 STYLE GUIDE

Use Lower Case Element Names

- HTML 5 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

Example

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>
```


HTML 5 STYLE GUIDE

Close All HTML Elements

- In HTML 5, you don't have to close all elements (for example the `<p>` element).
- We recommend closing all HTML elements.

Example

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>
```

HTML 5 STYLE GUIDE

Close Empty HTML Elements

- In HTML 5, it is optional to close empty elements.
- 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!

Example

Allowed:

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

Also Allowed:

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

HTML 5 STYLE GUIDE

Use Lower Case Attribute Names

- HTML 5 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

Example

Bad:

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

Good:

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

HTML 5 STYLE GUIDE

Quote Attribute Values

- HTML 5 allows attribute values without quotes.
- We recommend quoting attribute values because:
 - Mixing uppercase and lowercase values is bad
 - Quoted values are easier to read
 - You MUST use quotes if the value contains spaces

Example

Very bad:

This will not work, because the value contains spaces:

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

Bad:

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

Good:

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

HTML 5 STYLE GUIDE

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.

Example

Bad:

```

```

Good:

```

```

HTML 5 STYLE GUIDE

Spaces and Equal Signs

- HTML 5 allows spaces around equal signs. But space-less is easier to read and groups entities better together.

Example

Bad:

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



Good:

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

HTML 5 STYLE GUIDE

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.
- (Good luck with that.)

HTML 5 STYLE GUIDE

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:

Example

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>
```


HTML 5 STYLE GUIDE

Blank Lines and Indentation

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:

```
<ol>
  <li>London</li>
  <li>Paris</li>
  <li>Tokyo</li>
</ol>
```

HTML 5 STYLE GUIDE

Omitting <html> and <body>?

- In HTML 5, the <html> tag and the <body> tag can be omitted.
- The following code will validate as HTML 5:
- 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).

Example

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

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

Try it yourself: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_syntax_nobody

HTML 5 STYLE GUIDE

Omitting <head>?

- In HTML 5, 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:
- 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.

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: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_syntax_nohead

HTML 5 STYLE GUIDE

Meta Data

- The <title> element is required in HTML 5. Make the title as meaningful as possible:
 - `<title>HTML 5 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>HTML 5 Syntax and Coding Style</title>`
`</head>`

HTML 5 STYLE GUIDE

Setting The Viewport

- HTML 5 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.

HTML 5 STYLE GUIDE

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.

- Problem with comments:
 - In HTML it can be tricky to comment out a section of code that has comments in it!

```
<!--
```

```
<p>hey </p><!-- just saying  
"hey" -->
```

```
<p> it was the best of times  
</p>
```

```
-->
```

HTML 5 STYLE GUIDE

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

# HTML 5 STYLE GUIDE

## Loading JavaScript in HTML

- Use simple syntax for loading external scripts (the type attribute is not necessary):
  - `<script src="myscript.js">`



# HTML 5 STYLE GUIDE

## 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: [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_syntax\\_javascript](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_syntax_javascript)

# HTML 5 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.

# HTML 5 STYLE GUIDE

## 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.

# HTML 5 STYLE GUIDE

## 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.

# HTML 5 STYLE GUIDE

## 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.



# THE END

Web App Development

# SOURCE: W3SCHOOLS

