

Xhtml - Reference

By John Bentley

Introduction

This document brings together three standards:

- The W3C "HTML 5 A vocabulary and associated APIs for HTML and XHTML" standard.
- (Bentley, 2018. HTML Polyglot Markup Rules, \\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Html-PolyglotMarkupRules.docx), based on The W3C "Polyglot Markup: A robust profile of the HTML5 vocabulary."
- Basic markup and coding conventions stipulated by John Bentley for when authoring HTML 5 web pages. Chiefly stipulating the use of XHTML Syntax and serving the web pages as an "application/xhtml+xml" mime type. John Bentley stipulations will be marked with "jlb" or "JLB".

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html/>

(Bentley, 2018. HTML Polyglot Markup Rules,

\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Html-PolyglotMarkupRules.docx)

(W3C Polyglot, 2015. Polyglot Markup: A robust profile of the HTML5 vocabulary, W3C Candidate Recommendation) www.w3.org/TR/html-polyglot/

Essentially that entails a description of using HTML 5 with a XHTML conforming syntax, serving the page as an "application/xhtml+xml" mime type. The markup is further constrained conform to the Polyglot specification.

The polyglot specification defines a markup (and other aspects) that is common to both the HTML 5 spec and an XHTML syntax of it, thereby facilitating an easy switching between serving a web page as "application/xhtml+xml" or "text/html" and having it render identically. Essentially that results in mandating the XHTML syntax and forbidding the HTML syntax.

That will help if ever my predilection for using the more strict XHTML syntax is meet with resistance from the HTML Syntax crowd, the more popular choice, as I move my web pages into that (social) environment. All I'd need to do is change my file extensions from ".xhtml" to ".html", thereby changing the mime type with which the pages are served (that would need double checking that the web server is configured in this way) from "application/xhtml+xml" or "text/html".

Key

- "JLB Polyglot". A rule in virtue of (W3C 2015, "Polyglot Markup: A Robust Profile of the HTML5 Vocabulary", <https://www.w3.org/TR/html-polyglot/>) as interpreted by John Bentley in (Bentley 2019, "Xhtml-PolyglotMarkupRules.Docx", \\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Xhtml-PolyglotMarkupRules.docx).
- "JLB". A rule in virtue of what John Bentley requires, beyond what (W3C 2015, "Polyglot Markup: A Robust Profile of the HTML5 Vocabulary", <https://www.w3.org/TR/html-polyglot/>) or (Bentley 2019, "Xhtml-PolyglotMarkupRules.Docx", \\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Xhtml-PolyglotMarkupRules.docx) requires.

Basics

Use Polyglot Markup

(JLB) HTML 5 pages will use polyglot markup. Polyglot markup is HTML 5 using XHTML syntax (a syntax that is XML valid) with some additional restrictions to ensure the page looks the same whether served as a mime type of "text/html" or "application/xhtml+xml".

See (Bentley, 2015. XML Quick Reference) for writing well-formed XML.

(W3C Polyglot, 2015. *Polyglot Markup: A robust profile of the HTML5 vocabulary*, W3C Candidate Recommendation)
<http://www.w3.org/TR/2014/CR-html-polyglot-20140717/#principles>

Syntax

Relative Paths

Count the number of folders between the current document and the root. That is the number of "../" to prefix.

```
<!-- Root/Bjax/Examples/currentPage.html -->  
<link type="text/css" rel="stylesheet" href="../../Web/Apps/Softmake/Source/softmake.css" />  
<!-- Root/Web/Apps/..... -->
```

Local file references

Use three slashes after 'file:'

```

```

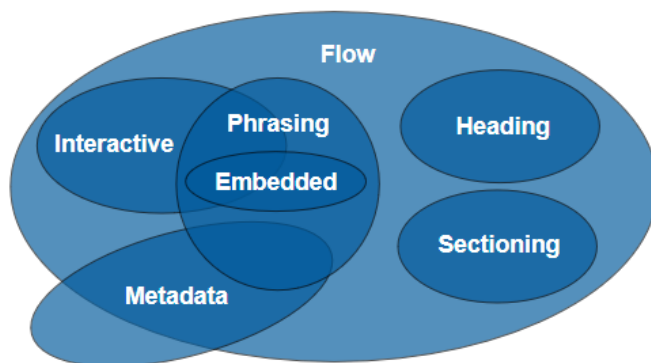
Elements

Element Overview

Categorising Elements

In the HTML 5 spec (W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) there are two separate categorising schemes for html elements:

1. **Kinds of content element category.** A "content model" "kinds of content" (or "categories"), including the following, and other categories:



(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/dom.html#kinds-of-content>

2. **Table of contents element category.** Elements categorised in the HTML 5 Specs's Table of contents, under "4 The elements of HTML". This is the categorisation scheme we'll use below.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/Overview.html#contents>

3. **JLB (John Luke Bentley's) chief surrounding elements:** main, article, sections, h1 ... h6 and sections.

As grouped in the Table of Contents

HTML 5 Elements grouped by HTML 5 Spec table of contents. Key: Elements New in HTML 5 will be **emphasized**.

Table of Contents Heading	Element
Root	html
Document Metadata	head title base link meta style
Sections	body article section nav aside h1, h2, h3, h4, h5, and h6 header footer address
Grouping Content	p hr pre blockquote ol ul li dl dt dd figure figcaption div main
Text-Level semantics	a em strong small s cite q dfn abbr data time code var samp kbd sub sup i b u mark ruby rb rt rtc rp bdi bdo span br wbr
Edits	ins del
Embedded content	img iframe embed object param video audio source track map area
Links	a area
Tabular data	table caption colgroup col tbody thead tfoot tr td th
Forms	form label input button select datalist optgroup option textarea keygen output progress meter fieldset legend
Scripting	script noscript template canvas

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/Overview.html#contents>

Element Reference

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/Overview.html#contents>

[Root](#)

html

the root of an HTML document.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-gb" xml:lang="en-gb">
<head>
```

[Document Metadata](#)

head

"a collection of metadata for the Document"

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-gb" xml:lang="en-gb">
<head>
  <title>A document with a short head</title>
</head>
<body>
```

title**base****link (element)****Overview**

Allows authors to link to external resources (but use script element for javascript).

```
<link rel="icon" href="favicon/my-favicon.png" />
<link rel="stylesheet" href="html5-reset.org.jlb.css" />
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/document-metadata.html#the-link-element>

The link (`rel`) types are divided into:

- Standard: icon, stylesheet, author, search, etc; and
- Other.

Other link types

The other link types are openly defined at http://microformats.org/wiki/existing-rel-values#HTML5_link_type_extensions . These sometimes work in conjunction with meta elements with name/content values.

```
<!-- Link type extensions
http://microformats.org/wiki/existing-rel-values#HTML5\_link\_type\_extensions -->
<link rel="schema.dc" href="http://purl.org/dc/elements/1.1/" />

<!-- Example meta name extensions, using dublin core.
See: https://wiki.whatwg.org/wiki/MetaExtensions -->
<meta name="dc.title" content="HTML 5 - Polyglot - Template - Example" />
<meta name="dc.creator" content="John Bentley" />
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/links.html#other-link-types>
See also [Meta element name/content attribute conventions](#)

favicon (link with rel "icon").

A "favicon" is a image that is associated with a website (or webpage) and used for bookmarking and displaying on a browser tab.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/links.html#rel-icon>
(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/document-metadata.html#the-link-element>

Create your favicon: from SVG; 32 X 32 pixels; save to png:

1. Use an SVG editor like <https://svg-edit.github.io/svgedit/releases/svg-edit-2.8.1/svg-editor.html> or <http://www.janvas.com/> (in Chrome).
2. Create your favicon in svg and export to png. The svg will be the editing source, the png file will be that which is used in production.

Historically, it was 16 x 16 pixel but you can (in order to edit in greater detail) use 32 x 32 pixels and the browsers will shrink it down OK.

(Bentley, n.d.) 2016-04-10

See C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\html5-favicon.xhtml

Link to your favicon image file from with a head section of (x)html. No need for mime type specification, no need for "shortcut" keyword in `rel` attribute.

```
<!-- Contemporary style: can be any name; of png file type; in any folder -->
<link rel="icon" href="favicon/my-favicon.png" />

<!-- Other (more detailed) forms -->
<link rel="icon" href="favicon.png" type="image/png" sizes="16x16" />

<!-- Traditional style: named "favicon"; ico file type; in root folder; "shortcut" precedes
"icon" -->
<link rel="shortcut icon" href="favicon.ico" />
```

Filetype support:

Image file extension	Mime Type	Support
ico	image/vnd.microsoft.icon	Yes
png	image/png	Yes
gif	image/gif	Yes
svg		limited

https://en.wikipedia.org/wiki/Favicon#File_format_support

Browser Support:

Browser	Popularity (2016-04-10)	Non root folder (e.g. href="favicon/my-favicon.png")	Need type attribute?
Chrome	51.9%	Yes	No
Safari	16.2%	Untested	Untested
IE and Edge	14.3%	Yes	No
Firefox	10.0%	Yes	No

(Bentley, n.d.) 2016-04-15 18:04

Refresh your favicon in firefox, for dev purposes, by deleting the favicon store:

1. Close firefox.

2. Delete "C:\Users\John\AppData\Roaming\Mozilla\Firefox\Profiles\2dc8fcq5.JLB-1561627104281\favicons.sqlite"
3. Reopen firefox.

(Stack Exchange Inc, 2015) [konzo's answer to How do I force a favicon refresh?](https://stackoverflow.com/a/32823504/872154)
<https://stackoverflow.com/a/32823504/872154>

stylesheet

Reference stylesheets as follows with the type attribute. For details see [Attributes > type](#).

```
<link rel="stylesheet" href="simple.css" />
<!-- No need for mime type specification -->
<link rel="stylesheet" href="simple.css" type="text/css" />
```

(Bentley, n.d.) 2016-04-17 15:28. Chrome, IE, FF.

meta

Overview

The meta element uses a name/content or http-equiv/content attribute pairs. Generally we only use the name/content pairs.

```
<!-- The following two meta name types are generally recommended (jlb) -->
<meta name="description" content="Ensure this is informative for the user. 160 characters max. Displays in search engine results. Gimpe dit molestias nihil numquam odio qui voluptate, voluptatum." />
<meta name="author" content="John Bentley" />

<!-- Pragma directives like the following are rarely used, and generally not recommended -->
<meta http-equiv="Refresh" content="300" />
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/document-metadata.html#the-meta-element>

HTML5: <meta name="generator" content="software package" /> " must not be used on pages whose markup is not generated by software, e.g. pages whose markup was written by a user in a text editor."

```
<!-- Do this -->
<meta name="generator" content="Fancy Page Creator 8.2" />

<!-- Not this -->
<meta name="generator" content="lovingly hand coded by me." />
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/document-metadata.html#the-meta-element>

<meta name="robots" content="..." /> can be used to control web search indexing on a page by page basis.

```
<meta name="robots" content="noindex, nofollow" />
```

See [Indexing control](#)

Meta element name/content attribute conventions

In the meta element the name/content attribute pairing has two conventions:

1. "Standard metadata names" and content values; and
2. "Extensions to the predefined set of metadata names".

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/document-metadata.html#standard-metadata-names>
<https://www.w3.org/TR/html5/document-metadata.html#other-metadata-names>

Standard metadata names include:

- application-name
- author
- description
- generator

```
<!-- The following two meta name types are generally recommended (jlb) -->
<meta name="description" content="Ensure this is informative for the user. 160 characters
max. Displays in search engine results. Gimpedit molestias nihil numquam odio qui voluptate,
voluptatum." />
<meta name="author" content="John Bentley" />
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/document-metadata.html#standard-metadata-names>

The `keywords` metadata name is discouraged.

Many search engines do not consider such keywords, because this feature has historically been used unreliably and even misleadingly as a way to spam search engine results in a way that is not helpful for users.

Google does not use the keywords metadata name.

(W3C HTML5.3, 2017) "4.2.5.1. Standard metadata names", <https://www.w3.org/TR/html5/document-metadata.html#standard-metadata-names>
 (Google Webmaster Central Blog, 2009) "Google does not use the keywords meta tag in web ranking", <https://webmasters.googleblog.com/2009/09/google-does-not-use-keywords-meta-tag.html>

Extensions to the predefined set of metadata names are those registered at [WHATWG Wiki MetaExtensions page](https://wiki.whatwg.org/wiki/MetaExtensions). Anyone can edit this page and propose an extension.

```
<!-- Link type extensions
http://microformats.org/wiki/existing-rel-values#HTML5_link_type_extensions -->
<link rel="schema.dc" href="http://purl.org/dc/elements/1.1/" />

<!-- Example meta name extensions, using dublin core.
See: https://wiki.whatwg.org/wiki/MetaExtensions -->
<meta name="dc.title" content="HTML 5 - Polyglot - Template - Example" />
<meta name="dc.creator" content="John Bentley" />
<meta name="dc.type" content="Event" />
<meta name="dc.type" content="Software" />
```

(WHATWG n.d., "MetaExtensions - WHATWG Wiki", Accessed 2021-06-24.
<https://wiki.whatwg.org/wiki/MetaExtensions>) <https://wiki.whatwg.org/wiki/MetaExtensions>
 See also [Other link types](#)

*todo ? Use Highwire Press tags (e.g. `citation_title`). As a last resort use Dublin Core tags (e.g. `DC.title`). See <https://wiki.whatwg.org/wiki/MetaExtensions>.

export bibliographic data in HTML "<meta>" tags. Google Scholar supports Highwire Press tags (e.g., citation_title), Eprints tags (e.g., eprints.title), BE Press tags (e.g., bepress_citation_title), and PRISM tags (e.g., prism.title). Use Dublin Core tags (e.g., DC.title) as a last resort - they work poorly for journal papers because Dublin Core doesn't have unambiguous fields for journal title, volume, issue, and page numbers.

(Google n.d., "Google Scholar Help > Inclusion Guidelines for Webmasters", Accessed 2021-06-24.
<https://scholar.google.com/intl/en/scholar/inclusion.html#indexing>
<https://scholar.google.com/intl/en/scholar/inclusion.html#indexing>

Zotero reads DC, prism, epress, bepress, and google highwire metatags from the site header.

(adamsmith 2014, "How to Create Website with Zotero-Friendly Metadata",
https://forums.zotero.org/discussion/comment/180759/#Comment_180759
https://forums.zotero.org/discussion/comment/180759/#Comment_180759

you can directly edit your HTML code of your webpage, using one of the following open standards, to expose your metadata

(Zotero 2018, "Zotero > Start > Dev > Exposing_metadata > Exposing Your Metadata > Using an Open Standard for Exposing Metadata",
https://www.zotero.org/support/dev/exposing_metadata#using_an_open_standard_for_exposing_metadata
https://www.zotero.org/support/dev/exposing_metadata#using_an_open_standard_for_exposing_metadata

(WHATWG n.d., "MetaExtensions - WHATWG Wiki", Accessed 2021-06-24.
<https://wiki.whatwg.org/wiki/MetaExtensions>) <https://wiki.whatwg.org/wiki/MetaExtensions>

Best practice of meta element name/content attribute

Default usage, for creating in Zotero an Item Type of 'Web Page'.

```
<!--
  In Zotero the whole entry will be imported as ItemType 'Journal Article',
  If Highwire Press tags citation_title or citation_journal_title are present
-->
<link rel="schema.dc" href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.dcterms" href="http://purl.org/dc/terms/" />

<meta name="dc.title" content="Epictetus Mixed" />
<meta name="dc.creator" content="Graver, Margaret" />

<!-- <meta name="citation_journal_title" content="Cool Journal" /> -->

<!-- Zotero recognized in Extra as Publisher -->
<meta name="dc.publisher" content="Great Publisher" />

<!--
  Regard as: OriginalDate (or FirstPublished )
  No Specification endorsed Metadata seems to exist for this.
  We use our own "original-date" which exists in CSL
  https://docs.citationstyles.org/en/stable/specification.html#date-variables
  and is supported by Zotero in Extra
-->
<meta name="original-date" content="2002-12-23" />

<!-- Not recognized in Zotero: But might like to use -->
<!-- Regard as: LastModified and therefore conceptually, The Date.
  The publication date of an academic paper.
  Google stipulates citation_publication_date over citation_date.
  https://www.google.com/intl/en/scholar/inclusion.html#indexing.
  Not recognized in Zotero -->
<!-- <meta name="citation_publication_date " content="2008/12/23" /> -->

<!--
  Regard as: LastModified and therefore conceptually, The Date.
  Date of formal issuance (e.g., publication) of the resource.
  Zotero recognized in Date.
-->
```

```

<meta name="dcterms.issued" content="2008-12-23" />

<!--
  Don't use as we really want the LastModified Date to be put in Zotero Date,
  and broadly be regarded as The Date.
  Date on which the resource was changed.
  Zotero recognized in Extra as Last Modified.
-->
<!-- <meta name="dcterms.modified" content="2021-06-15" /> -->

<!-- A summary of the resource. Zotero recognized in Abstract. -->
<meta name="dcterms.abstract" content="dcterms.abstract. Lorem ipsum dolor sit
amet consectetur adipiscing elit." />

```

(Bentley, n.d., *Bentley Experimentation*) 2021-06-24,
[\\atlas\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\metadata-best-practice--create-web-page-type.xhtml](#)

Default usage, for creating in Zotero an Item Type of 'Journal Article'.

```

<!-- In Zotero the whole entry will be imported as ItemType 'Journal Article',
  If citation_title or citation_journal_title is present-->
<meta name="citation_title" content="Epictetus Create Journal Article Type" />
<meta name="citation_author" content="Graver, Margaret" />

<!-- Zotero recognized in Extra as Publisher -->
<meta name="citation_publisher" content="Great Publisher" />

<!-- The title of the journal in which an academic paper is published. Zotero recognized
in Publication -->
<meta name="citation_journal_title" content="Cool Journal" />

<!-- Not recognized in Zotero: But might like to use -->
<!-- The publication date of an academic paper.
  Google stipulates citation publication date over citation date.
  https://www.google.com/intl/en/scholar/inclusion.html#indexing.
  Not recognized in Zotero -->
<meta name="citation_publication_date " content="2008/12/23" />

  <!--
  Regard as: FirstPublished
  Date of formal issuance (e.g., publication) of the resource.
  Zotero recognized in Date.
-->
<meta name="dcterms.issued" content="2008-12-23" />

  <!--
  Regard as: LastSubstantiveModification
  Date on which the resource was changed.
  Zotero recognized in Extra as Last Modified.
-->
<meta name="dcterms.modified" content="2021-06-15" />

  <!--
  Regard as: LastMinorModification
  No Metadata seems to exist for this.
-->

  <!-- Not recognized in Zotero -->
  <!-- <meta name="citation_date " content="2008/12/23" /> -->
  <!-- <meta name="citation_pdf_url" content="http://example.com/articles/pdf/Epictetus.pdf"
/> -->
  <!-- <meta name="citation_fulltext_html_url"
content="http://example.com/articles/html/Epictetus.xhtml" /> -->

  <!-- No citation_modified_date -->
  <!-- No citation_description or citation_abstract -->

```

(Bentley, n.d., *Bentley Experimentation*) 2021-06-24,
[\\atlas\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\metadata-best-practice--create-journal-article-type.xhtml](#)

Use Highwire press tags like this.

```
<!-- ##### Highwire Press tags -->
<!-- In Zotero the whole entry will be imported as ItemType 'Journal Article',
      If citation_title or citation_journal_title is present-->
<meta name="citation_title" content="Epictetus Highwire Press tags" />
<meta name="citation_author" content="Graver, Margaret" />

<!-- Zotero recognized in Extra as Publisher -->
<meta name="citation_publisher" content="Great Publisher" />

<!-- The title of the journal in which an academic paper is published. Zotero recognized
in Publication -->
<meta name="citation_journal_title" content="Cool Journal" />

<!-- Not recognized in Zotero -->

<!-- Not recognized in Zotero: But might like to use -->
<!-- The publication date of an academic paper.
      Google stipulates citation_publication_date over citation_date.
      https://www.google.com/intl/en/scholar/inclusion.html#indexing.
      Not recognized in Zotero -->
<meta name="citation_publication_date" content="2008/12/23" />

<!-- <meta name="citation_date" content="2008/12/23" /> -->
<!-- <meta name="citation_pdf_url" content="http://example.com/articles/pdf/Epictetus.pdf"
/> -->
<!-- <meta name="citation_fulltext_html_url"
content="http://example.com/articles/html/Epictetus.xhtml" /> -->

<!-- No citation_modified_date -->
<!-- No citation_description or citation_abstract -->
```

(Bentley, n.d., **Bentley Experimentation**) 2021-06-24,
[\\atlas\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\metadata-highwire-press-tags.xhtml](#)

Use Dublin Core tags like this.

```
<!-- In Zotero the whole entry will be imported as ItemType: Web Page.
      Unless Highwire Press Tags citation_title or citation_journal_title are present -->
<link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
<link rel="schema.DCTERMS" href="http://purl.org/dc/terms/" />
<meta name="dc.title" content="Epictetus Dublin Core" />
<meta name="dc.creator" content="Graver, Margaret" />

<!-- Zotero recognized in Extra as Publisher -->
<meta name="dc.publisher" content="Great Publisher" />

<!-- Date of formal issuance (e.g., publication) of the resource. Zotero recognized in
Date. -->
<meta name="dcterms.issued" content="2008-12-23" />

<!-- Date on which the resource was changed. Zotero recognized in Extra as Last Modified.
-->
<meta name="dcterms.modified" content="2021-06-15" />

<!-- A summary of the resource. Zotero recognized in Abstract. -->
<meta name="dcterms.abstract" content="dcterms.abstract. Lorem ipsum dolor sit amet
consectetur adipisicing elit." />

<!-- ##### Don't Use -->
<!-- An account of the resource. Description may include but is not limited to: an
abstract, a table of contents, a graphical representation, or a free-text account of the
resource. Zotero recognized in Abstract. -->
<!-- <meta name="dcterms.description" content="dcterms.description. Lorem ipsum dolor sit
amet consectetur adipisicing elit." /> -->

<!-- A point or period of time associated with an event in the lifecycle of the resource.
Zotero recognized in Date. -->
<!-- <meta name="dcterms.date" content="2007-12-30" /> -->

<!-- Not recognized in Zotero -->
<!-- <meta name="dc.created" content="2008-12-23" /> -->
```

```
<!-- Not recognized in Zotero -->
<!-- <meta name="dc.modified" content="2021-06-15" /> -->

<!-- No url (pdf, html, etc) fields -->
```

(Bentley, n.d., *Bentley Experimentation*) 2021-06-24,
[\\atlas\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\metadata-dublin-core.xhtml](#)

Datetime metadata candidates

The following metadata names, registered at the [WHATWG Wiki MetaExtensions page](#), could be used for a page datetime. Dublin core dates can use ISO 8601, including the Extended Date/Time Format Specification [EDTF].

```
<!--
  Regard as: OriginalDate (or FirstPublished )
  No Specification endorsed Metadata seems to exist for this.
  We use our own "original-date" which exists in CSL
  https://docs.citationstyles.org/en/stable/specification.html#date-variables
  and is supported by Zotero in Extra
-->
<meta name="original-date" content="2002-12-23" />

<!-- Not recognized in Zotero: But might like to use -->
<!-- Regard as: LastModified and therefore conceptually, The Date.
  The publication date of an academic paper.
  Google stipulates citation_publication_date over citation_date.
  https://www.google.com/intl/en/scholar/inclusion.html#indexing.
  Not recognized in Zotero -->
<!-- <meta name="citation_publication_date " content="2008/12/23" /> -->

<!--
  Regard as: LastModified and therefore conceptually, The Date.
  Date of formal issuance (e.g., publication) of the resource.
  Zotero recognized in Date.
-->
<meta name="dcterms.issued" content="2008-12-23" />

<!--
  Don't use as we really want the LastModified Date to be put in Zotero Date,
  and broadly be regarded as The Date.
  Date on which the resource was changed.
  Zotero recognized in Extra as Last Modified.
-->
<!-- <meta name="dcterms.modified" content="2021-06-15" /> -->
```

Dublin Core dates use well formatted datetimes:

Recommended practice is to express the date, date/time, or period of time according to ISO 8601-1 [ISO 8601-1] or a published profile of the ISO standard, such as the W3C Note on Date and Time Formats [W3CDTF] or the Extended Date/Time Format Specification [EDTF]

(Dublin Core Metadata Initiative 2020, "DCMI: Date", <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/elements11/date/>) <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/elements11/date/>

(Bentley, n.d., *Bentley Experimentation*) 2021-06-24,
[\\atlas\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\metadata-best-practice--create-journal-article-type.xhtml](#)

style (block)

For style linking (i.e. referencing a stylesheet external) see [Document Metadata > link > style](#).

Create a style block in a <head> as follows.

```
<head>
  <style>
    /*  */
    /* A comment */

    h1 {
      color: blue;
    }
    /*  */
  </style>
  ...
</head>
```

(W3C HTML5.3, 2017) <https://www.w3.org/TR/html53/document-metadata.html#the-style-element>
(Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-example.xhtml

The use of `type="text/css"` for css, is discouraged. See For details see [Attributes > type](#).

For comments and CDATA escaping in a style block see [Make safe raw text in <script> and <style>](#).

script

External script.

```
<script src="quirksmode-check.js"></script>
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/scripting-1.html#the-script-element>
(Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-example.xhtml

Inline script.

```
<script>
  /*  */
  function replaceTextInElement(id, paragraphText) {
    var targetElementNode = document.getElementById(id);
    targetElementNode.childNodes[0].nodeValue = paragraphText;
  }
  ...
  /*  */
</script>
```

(Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-example.xhtml

For comments and CDATA escaping in a script block see [Make safe raw text in <script> and <style>](#).

You can drop the "type" attribute if the mime type of the script is "text/javascript".


```

<!-- Do this -->
<script src="quirksModeCheck.js"></script>

<!-- Old school -->
<script src="quirksmode-check.js" type="text/javascript"></script>

```

"The type attribute gives the language of the script or format of the data. If the attribute is present, its value must be a valid MIME type. ... The default, which is used if the attribute is absent, is "text/javascript"."

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/scripting-1.html#the-script-element>

Sections

body

header

header ... represents introductory content for its nearest ancestor sectioning content [article, section, nav, aside] or sectioning root [body, fieldset, blockquote, figure, td].

- When the nearest ancestor sectioning content or sectioning root ... is the body element, then it applies to the whole page.
- A header typically contains a group of introductory [text] or navigational aids.
- A header ... is intended to usually contain ... an h1–h6, but this is not required.

```

<body>
  <main>
    <header>
      <h1>Apples</h1>
      <p>By Joe Blow</p>
      <p>The apple is the pomaceous fruit of the apple tree.</p>
      <p>Tasty, delicious fruit!</p>
      <nav>
        <ul>
          <li><a href="#redDelicious">Red Delicious</a></li>
          <li><a href="#grannySmith">Granny Smith</a></li>
        </ul>
      </nav>
    </header>
    <section>
      <h2 id="redDelicious">Red Delicious</h2>
      <p>These bright red apples are the most common found in many supermarkets.</p>
    </section>
    <section>
      <h2 id="grannySmith">Granny Smith</h2>
      <p>These juicy, green apples make a great filling for apple pies.</p>
    </section>
  </main>
</body>

```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/sections.html#the-header-element>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\html5-section.xhtml

The header element can also be used to wrap a section's table of contents, a search form, or any relevant logos.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/sections.html#the-header-element>

The header element is not sectioning content; it doesn't introduce a new section.

(W3C HTML5.3, 2017), " 4.3.7. The header element", <https://www.w3.org/TR/html5/sections.html#the-header-element>

footer

footer ... represents a footer for its nearest ancestor sectioning content [article, section, nav, aside] or sectioning root [body, fieldset, blockquote, figure, td].

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/sections.html#the-footer-element>

address [Contact info]

The address element represents the contact information for its nearest article or body element ancestor. Some notes:

- The address element must not be used to represent arbitrary addresses (e.g. postal addresses), unless those addresses are in fact the relevant contact information.
- The address element must not contain information other than contact information.

```
<address>
  <A href=" ../People/Raggett/">Dave Raggett</A>,
  <A href=" ../People/Arnaud/">Arnaud Le Hors</A>,
  contact persons for the <A href="Activity">W3C HTML Activity</A>
</address>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/sections.html#the-address-element>

JLB: The address element would have been better named "contact".

Grouping Content

p

authors are strongly encouraged to always mark up all paragraphs with the p element

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/edits.html#edits-and-paragraphs>

hr

pre

"a block of preformatted text", that is whose whitespace is preserved.

- ..." to represent a block of computer code, the `pre` element can be used with a `code` element;
- to represent a block of computer output the `pre` element can be used with a `samp` element.
- Similarly, the `kbd` element can be used within a `pre` element to indicate text that the user is to enter."

```
<p>This is the <code>Panel</code> constructor:</p>
<pre><code>function Panel(element, canClose, closeHandler) {
  this.element = element;
  this.canClose = canClose;
  this.closeHandler = function () { if (closeHandler) closeHandler() };
}</code></pre>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/grouping-content.html#the-pre-element>

jlB: place the first line of the code adjacent to the first `<pre>` tag (and any code, output, or `kbd` tag.

```
<pre><code>function Panel(element, canClose, closeHandler) {
```

jlB: Remove any whitespace on the line before `<pre>` and `</pre>` tags.

```
<!-- Do this -->
...
    (full stop) to indicate the end of the program.
    </p>

    <!-- Referencing computer code as a block -->
<p>This is the <code>Panel</code> constructor:</p>
<pre><code class="language-javascript">function Panel(element, canClose, closeHandler) {
  this.element = element;
  this.canClose = canClose;
  this.closeHandler = function () { if (closeHandler) closeHandler() };
}
</code></pre>
...
</section>

<!-- Not this -->
    (full stop) to indicate the end of the program.
    </p>

    <!-- Referencing computer code as a block -->
<p>This is the <code>Panel</code> constructor:</p>
<pre><code class="language-javascript">function Panel(element, canClose,
  closeHandler) {
  this.element = element;
  this.canClose = canClose;
  this.closeHandler = function () { if (closeHandler) closeHandler() };
}
</code></pre>
</section>
```

blockquote

"content that is quoted from another source, optionally with a citation which must be within a footer or cite element"

```
<blockquote>
  The people recognize themselves in their commodities; they find their
  soul in their automobile, hi-fi set, split-level home, kitchen equipment.
```

```

- <cite><a href="http://en.wikipedia.org/wiki/Herbert_Marcuse">Herbert Marcuse</a></cite>
</blockquote>

<blockquote>
<p>I contend that we are both atheists. I just believe in one fewer
god than you do. When you understand why you dismiss all the other
possible gods, you will understand why I dismiss yours.</p>
<footer>- <cite>Stephen Roberts</cite></footer>
</blockquote>

```

Lists

overview

No text is permitted between the opening ol or ul and the first li.

```

<!-- Do this -->
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>
<ul>
  <li>The first item</li>
  <li>The second item</li>
</ul>

<!-- Not this -->
<ul>Lorem ipsum dolor sit amet consectetur adipisicing elit.
  <li>The first i tem</li>
  <li>The second item</li>
</ul>

```

(Castro, 2003, p.205)

To collapse the distance between a list introduction and the list items do this:

```

<p style='margin-bottom: 0;'>To collapse the distance between a list introduction and the
list items do this:</p>
<ul style='margin-top: 0;'>
  <li>The first i tem</li>
  <li>The second item</li>
</ul>

```

C:\Data\Sda\Code\Web\Examples\XhtmlExamples\quickReferenceExamples.html

ol

"list of items, where the items have been intentionally ordered such that changing the order would change the meaning of the document. ... The start attribute, if present, must be a valid integer giving the ordinal value of the first list item".

There is a CSS mapping of 'list-style-type' to the attribute 'type' with a value of either: decimal; lower-alpha; upper-alpha; lower-roman; upper-roman'.

```

<p>Stages of development:</p>
<ol start="2" style="list-style-type: lower-roman">
  <li>Placid.</li>
  <li>Interested.</li>
  <li>Compelled.</li>
</ol>

ul

<p>For instance, this fantastic sentence has bullets relating to</p>
<ul>
  <li>wizards,</li>
  <li>faster-than-light travel, and</li>
  <li>telepathy,</li>
</ul>
<p>and is further discussed below.</p>

```

(W3C HTML5.3, 2017), "The ol element", <https://www.w3.org/TR/html53/grouping-content.html#the-ol-element>

li

dl

"an association list consisting of zero or more name-value groups (a description list) ... Name-value groups may be terms and definitions, metadata topics and values, questions and answers, or any other groups of name-value data."

```

<!-- General Form -->
<dl>
  <dt>Term</dt>
  <dd>Description.</dd>

  <dt>Term</dt>
  <dd>Description.</dd>

  <dt>Term</dt>
  <dt>Term</dt>
  <dt>Term</dt>
  <dd>Description.</dd>

  <dt>Term</dt>
  <dd>Description.</dd>
  <dd>Description.</dd>
  <dd>Description.</dd>

  <dt>Term</dt>
  <dt>Term</dt>
  <dd>Description.</dd>
  <dd>Description.</dd>
</dl>

<!-- fields and values -->
<dl>
  <dt>Last modified time</dt>
  <dd>2004-12-23T23:33Z</dd>

  <dt>Recommended update interval</dt>
  <dd>60s</dd>

  <dt>Authors</dt>
  <dt>Editors</dt>
  <dd>Robert Rothman</dd>
  <dd>Daniel Jackson</dd>
</dl>

```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/grouping-content.html#the-dl-element>

A common use for a description list is as a definition list. In production, when a description list is a definition list use a dfn element.

```

<dl>
  <dt><dfn>Apartment</dfn>, n.</dt>
  <dd>A home for rats.</dd>

  <dt><dfn>Flat</dfn>, n.</dt>
  <dd>A deflated tire.</dd>

  <dt><dfn>Home</dfn>, n.</dt>
  <dd>That which takes most of your time.</dd>
</dl>

```

A description list can have:

1. One term for one description.
2. Many terms for one description.
3. One terms for many descriptions.
4. Many terms for many descriptons.

```

<!--
  For these examples we won't include a dfn element
-->
<dl>
  <!-- One term for one description -->
  <dt>Apartment</dt>
  <dd>A home for rats.</dd>

  <dt>Flat</dt>
  <dd>A deflated tire.</dd>

  <dt>Home</dt>
  <dd>That which takes most of your time.</dd>

  <!-- Many terms for a description -->
  <dt>Elephant</dt>
  <dt>Lion</dt>
  <dt>Giraffe (a majestic beast with a long neck)</dt>
  <dd>Animals in Ngorongoro crater, Tanzania.</dd>

  <!-- One term for many descriptons -->
  <dt>Bank</dt>
  <dd>The land alongside or sloping down to a river or lake.</dd>
  <dd>A financial establishment that uses money deposited by customers for investment, pays
  it out when required, makes loans at interest, and exchanges currency.</dd>
  <dd>A set of similar things, especially electrical or electronic devices, grouped together
  in rows.</dd>

  <!-- Many terms for many descriptions -->
  <dt>Authors</dt>
  <dt>Editors</dt>
  <dd>Robert Rothman</dd>
  <dd>Daniel Jackson</dd>
</dl>

```

(webplatform.org, 2016) <http://docs.webplatform.org/wiki/html/elements/dl>

Separate each term(s) and description(s) set with a line.

```

<dl>
  <dt>Term</dt>
  <dd>Description.</dd>

  <dt>Term</dt>
  <dd>Description.</dd>

  <dt>Term</dt>
  <dt>Term</dt>
  <dt>Term</dt>
  <dd>Description.</dd>

  <dt>Term</dt>
  <dd>Description.</dd>
  <dd>Description.</dd>
  <dd>Description.</dd>
</dl>

```

There is no markup that wraps each term-descriptoin set. Therefore, for readability, use this separating line. Most IDEs will honour the new line (or can be configured to honour the new line), but will not honour other ways of marking the separation (e.g. by putting term(s) and descripton(s) on one line).

(Bentley, 2016. *Web Code Convention - Naming and Style - General*,
 file://Atlas/Users/John/Documents/Sda/Info/Web/System/Conventions/Web Code Convention - Naming and Style -

General.docx)

https://www.reddit.com/r/webdev/comments/4agb27/html_spec_proposal_description_lists_ought_be/

You can't apply list styles directly to dl elements but you can use various styling options to make the dl stand out.

```
<!-- You can do this -->
dt , dt > dfn
{
  font-weight: bold;
}

dd
{
  margin: 0 0 0.5em 2em;
}

<!-- The following doesn't work -->
dl
{
  list-style: disc;
}
```

(Bentley, n.d.) 2016-02-06 18:47 C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-example.xhtml

The browser default formatting of `dl(s)` best handles the various cardinality of lists (one to one, one to many, and many to one). However, an alternative to the default formatting, mostly putting a term and description on one line, is as follows.

```
/*
Description lists (and sometimes using a dfn element).

An alternative to the default format.
*/

dl {
  padding-top: 0.5em;
  border-top: 1px solid black;
}

dt {
  float: left;
  width: 8em;
}

dd {
  margin-left: 8em;
}

dt + dt
{
  float: none;
}

/* Pretty extras */
dt, dt > dfn {
  font-weight: bold;
}

dt::after {
  content: ":";
}
```

(Bentley, n.d.) 2016-03-13 06:01

C:\users\john\documents\sda\code\web\libraries\html5library\html5-polyglot-template-example.xhtml

C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-basic.css

<http://code.webplatform.org/gist/c75690a550a3a0589fe3> (In Chrome)

dt

"the term, or name, part of a term-description group in a description list"

dd

"the description, definition, or value, part of a term-description group in a description list"

figure

"flow content, optionally with a caption, that is self-contained (like a complete sentence) and is typically referenced as a single unit from the main flow of the document. used to annotate illustrations, diagrams, photos, code listings, etc."

```
<p>In <a href="#l4">listing 4</a> we see the primary core interface
API declaration.</p>
<figure id="l4">
  <figcaption>Listing 4. The primary core interface API declaration.</figcaption>
  <pre><code>interface PrimaryCore {
  boolean verifyDataLine();
  void sendData(in sequence<byte> data);
  void initSelfDestruct();
}</code></pre>
</figure>
<p>The API is designed to use UTF-8.</p>

<!-- Next example -->

<p>This case centered on some sort of "intellectual property"
infringement related to a comic (see Exhibit A).</p>

<figure>
  
  <figcaption>Exhibit A. The alleged <cite>rough copy</cite> comic.</figcaption>
</figure>
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/grouping-content.html#the-figure-element>

figcaption

"The first figcaption element child of the [figure] element, if any, represents the caption of the figure element's contents. If there is no child figcaption element, then there is no caption."

div

JLB chief surrounding elements

The Sectioning Content (kind of content)

In the "kind of content" categorization Sectioning Content includes: <article>, <section>, <nav>, and <aside>. They define the scope of headings (h1 ... h6) and footers (<footer>).

(W3C HTML5.3, 2017), "3.2.4.2.3. Sectioning content", <https://www.w3.org/TR/html53/dom.html#sectioning-content>

article (toc sections)

`article` represents ...

a complete, or self-contained, composition ... that is, in principle, independently distributable or reusable, e.g. in syndication. This could be a forum post, a magazine or newspaper article, a blog entry, a user-submitted comment, an interactive widget or gadget, or any other independent item of content. [Emphasis added].

```
<body>
  <!-- Several newspaper articles -->
  <article>
    <h1>All about Cars</h1>

    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>

    <p>Accusantium amet aspernatur dolorem est facilis .</p>

    <p>Accusamus culpa distinctio excepturi laborum reprehenderit?</p>
  </article>
  <article>
    <h1>All about Apples</h1>

    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>

    <p>Accusantium amet aspernatur dolorem est facilis .</p>

    <p>Accusamus culpa distinctio excepturi laborum reprehenderit?</p>
  </article>
</body>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/sections.html#the-article-element>

An example of nesting articles within a main.

```
<!-- other content -->

<main>

  <h1>Skateboards</h1>
  <p>The skateboard is the way cool kids get around</p>

  <article>
    <h2>Longboards</h2>
    <p>Longboards are a type of skateboard with a longer wheelbase and larger, softer wheels.</p>
    <p>... </p>
    <p>... </p>
  </article>

  <article>
    <h2>Electric Skateboards</h2>
    <p>These no longer require the propelling of the skateboard by means of the feet; rather an electric motor propels the board, fed by an electric battery.</p>
    <p>... </p>
    <p>... </p>
  </article>

</main>

<!-- other content -->
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/grouping-content.html#the-main-element>

If there is only one self-contained composition on the page mark it up with an `article` and a `main`.

```
// Do this
<body>
  <!-- One newspaper article -->
```

```

<main>
  <!-- An article, "sectioning content", makes it clear what h1 ... h6 apply to -->
  <article>
    <h1>All about Cars</h1>

    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>

    <p>Accusantium amet aspernatur dolorem est facilis .</p>

    <p>Accusamus culpa distinctio excepturi laborum reprehenderit?</p>
  </article>
</main>
</body>

// Not this
<body>
  <!-- One newspaper article -->
  <main>
    <h1>All about Cars</h1>

    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>

    <p>Accusantium amet aspernatur dolorem est facilis .</p>

    <p>Accusamus culpa distinctio excepturi laborum reprehenderit?</p>
  </main>
</body>

```

The HTML 5 spec points out that where there is only one `article` it is redundant:

When the main content of the page (i.e. excluding footers, headers, navigation blocks, and sidebars) is all one single self-contained composition, the content should be marked up with a `main` element and the content may also be marked with an `article`, but it is technically redundant in this case (since it's self-evident that the page is a single composition, as it is a single document).

However, rules around `h1 ... h6` entail that they markup the nearest ancestor sectioning content. Sectioning content is specified to apply to [`article`, `section`, `nav`, `aside`], not `main`. Therefore using `article` makes it clear which ancestor our `h1 ... h6` takes effect for. See [h1 ... h6 and sections](#) elsewhere in this document.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/sections.html#the-article-element>

When `article` elements are nested, the inner `article` elements represent articles that are in principle related to the contents of the outer `article`.

"For instance, a blog entry on a site that accepts user-submitted comments could represent the comments as `article` elements nested within the `article` element for the blog entry."

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/sections.html#the-article-element>

section (toc sections)

section represents ...

... a thematic grouping of content [in a] ... document or application.

1. "The theme of each section should be identified, typically by including a heading (h1-h6 element) as a child of the section element."
2. "When an element is needed only for styling purposes or as a convenience for scripting, authors are encouraged to use the `div` element instead. A general rule

is that the `section` element is appropriate only if the element's contents would be listed explicitly in the document's outline."

3. "Authors are encouraged to use the `article` element instead of the `section` element when it would make sense to syndicate the contents of the element."

```
<!-- Examples of sections would be chapters, the various tabbed pages in a tabbed dialog
box, or the numbered sections of a thesis. A Web site's home page could be split into
sections for an introduction, news items, and contact information. -->
```

```
<main>
  <article>
    <header>
      <h2>Apples</h2>
      <p>Tasty, delicious fruit!</p>
    </header>

    <p>The apple is the pomaceous fruit of the apple tree.</p>

    <section>
      <h3>Red Delicious</h3>
      <p>These bright red apples are the most common found in many supermarkets.</p>
    </section>

    <section>
      <h3>Granny Smith</h3>
      <p>These juicy, green apples make a great filling for apple pies.</p>
    </section>
  </article>
</main>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/sections.html#the-section-element>

article V section (toc sections)

Generally, if there are `section` and `article` elements the `section` is nested inside an `article`.

```
<main>
  <article>
    <header>
      <h2>Apples</h2>
      <p>Tasty, delicious fruit!</p>
    </header>

    <p>The apple is the pomaceous fruit of the apple tree.</p>

    <section>
      <h3>Red Delicious</h3>
      <p>These bright red apples are the most common found in many supermarkets.</p>
    </section>

    <section>
      <h3>Granny Smith</h3>
      <p>These juicy, green apples make a great filling for apple pies.</p>
    </section>
  </article>
</main>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/sections.html#article-or-section?>

A section forms part of something else. An article is its own thing. But how does one know which is which? Mostly the real answer is "it depends on author intent".

For example, one could imagine a book with a "Granny Smith" chapter that just said "These juicy, green apples make a great filling for apple pies."; that would be a section because there'd be lots of other chapters on (maybe) other kinds of apples.

On the other hand, one could imagine a tweet or reddit comment or tumblr post or newspaper classified ad that just said "Granny Smith. These juicy, green apples make a great filling for apple pies."; it would then be articles because that was the whole thing.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/sections.html#article-or-section? nav>

nav (toc sections)

nav represents

section of a page that links to other pages or to parts within the page: a section with navigation links.

```
<!-- Note in this example the the nav in the header element contains links to other pages
and the nav in the main element consisting of links to sub sections of the main content. -->

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="UTF-8" />
  <title>HTML 5 - Main</title>
</head>
<body>
  <header>The Lawson Academy:
    <nav>
      <ul>
        <li>
          <a href="courses.html">Courses</a>
        </li>
        <li>
          <a href="fees.html">Fees</a>
        </li>
        <li>
          <a href="graduation.html">Graduation</a>
        </li>
      </ul>
    </nav>
  </header>

  <main>
    <h1>Graduation</h1>

    <nav>
      <ul>
        <li>
          <a href="#ceremony">Ceremony</a>
        </li>
        <li>
          <a href="#graduates">Graduates</a>
        </li>
        <li>
          <a href="#awards">Awards</a>
        </li>
      </ul>
    </nav>

    <h2 id="ceremony">Ceremony</h2>
    <p>Opening Procession</p>
    <p>Speech by Valedictorian</p>
    <p>Speech by Class President</p>
    <p>Presentation of Diplomas</p>
    <p>Closing Speech by Headmaster</p>

    <h2 id="graduates">Graduates</h2>
    <ul>
      <li>Eileen Williams</li>
```

```

    <li>Andy Maseyk</li>
    <li>Blanca Sainz Garcia</li>
  </ul>

  <h2 id="awards">Awards</h2>
  <ul>
    <li>Clara Faulkner</li>
    <li>Eloisa Faulkner</li>
    <li>Blanca Sainz Garcia</li>
  </ul>
</main>

<footer>Copyright 2012 B.lawson</footer>
</body>
</html>

```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/sections.html#the-nav-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\html5-main.xhtml

aside (toc sections)

h1 ... h6 (toc sections)

[h1 . . . h6] represent headings for their sections.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/sections.html#the-h1,-h2,-h3,-h4,-h5,-and-h6-elements>

h1 ... h6 and sections (toc sections)

Basic points:

- h1 . . . h6 is "heading content".
- The first heading content that is a child of sectioning content [article, section, nav, aside], represents the heading for that section.

Explicit sectioning (encouraged):

Authors are ... encouraged to explicitly wrap sections in elements of sectioning content [article, section, nav, aside] instead of relying on the implicit sections generated by having multiple headings in one element of sectioning content.

```

// Encouraged.
<body>
  <article>
    <h1>Apples</h1>
    <p>Apples are fruit.</p>
    <section>
      <h2>Taste</h2>
      <p>They taste lovely.</p>
    <section>
      <h3>Sweet</h3>
      <p>Red apples are sweeter than green ones.</p>
    </section>
  </section>
  <section>
    <h2>Color</h2>
    <p>Apples come in various colors.</p>
  </section>
</article>
</body>

// The following is permitted but harder to read, given the implied sections.

```

```

<body>
  <h4>Apples</h4>
  <p>Apples are fruit.</p>
  <section>
    <h2>Taste</h2>
    <p>They taste lovely.</p>
    <h6>Sweet</h6>
    <p>Red apples are sweeter than green ones.</p>
    <h1>Color</h1>
    <p>Apples come in various colors.</p>
  </section>
</body>

```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/sections.html#headings-and-sections>

Implicit sectioning (discouraged):

Subsequent headings of equal or higher rank start new new (implied) sections, headings of lower rank start subsections that are part of the previous one. In both cases, the element represents the heading of the implied section.

When creating subsections an alternative to incrementing the number of the hN heading for a subsection is to use a hN of an equal rank, but wrapped in sectioning content [article, section, nav, aside]. This is useful for documents where moving the sections or headings around is expected to be a heavy part of the editing.

```

// For documents that have sections moved around a lot do this
<body>
  <h1>Apples</h1>
  <p>Apples are fruit.</p>
  <section>
    <h1>Taste</h1>
    <p>They taste lovely.</p>
  </section>
  <section>
    <h1>Sweet</h1>
    <p>Red apples are sweeter than green ones.</p>
  </section>
  <section>
    <h1>Color</h1>
    <p>Apples come in various colors.</p>
  </section>
</body>

// Instead of this
<body>
  <h1>Apples</h1>
  <p>Apples are fruit.</p>
  <section>
    <h2>Taste</h2>
    <p>They taste lovely.</p>
  </section>
  <section>
    <h3>Sweet</h3>
    <p>Red apples are sweeter than green ones.</p>
  </section>
  <section>
    <h2>Color</h2>
    <p>Apples come in various colors.</p>
  </section>
</body>

```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/sections.html#headings-and-sections>

main (toc grouping content)

main represents ...

the main content of the `body` [element]. ... excludes content that is repeated across a set of documents such as site navigation links, copyright information, site logos ... Authors must not include more than one main element in a document. Authors must not include the main element as a descendant of an `article`, `aside`, `footer`, `header` or `nav` element.

`<!-- Here is a graduation programme the main content section is defined by the use of the main element. Note in this example the main element contains a nav element consisting of links to sub sections of the main content. -->`

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="UTF-8" />
  <title>HTML 5 - Main</title>
</head>

<body>
  <header>The Lawson Academy:
    <nav>
      <ul>
        <li>
          <a href="courses.html">Courses</a>
        </li>
        <li>
          <a href="fees.html">Fees</a>
        </li>
        <li>
          <a href="graduation.html">Graduation</a>
        </li>
      </ul>
    </nav>
  </header>

  <main>
    <h1>Graduation</h1>

    <nav>
      <ul>
        <li>
          <a href="#ceremony">Ceremony</a>
        </li>
        <li>
          <a href="#graduates">Graduates</a>
        </li>
        <li>
          <a href="#awards">Awards</a>
        </li>
      </ul>
    </nav>

    <h2 id="ceremony">Ceremony</h2>
    <p>Opening Procession</p>
    <p>Speech by Valedictorian</p>
    <p>Speech by Class President</p>
    <p>Presentation of Diplomas</p>
    <p>Closing Speech by Headmaster</p>

    <h2 id="graduates">Graduates</h2>
    <ul>
      <li>Eileen Williams</li>
      <li>Andy Maseyk</li>
      <li>Blanca Sainz Garcia</li>
    </ul>

    <h2 id="awards">Awards</h2>
    <ul>
      <li>Clara Faulkner</li>
      <li>Eloisa Faulkner</li>
      <li>Blanca Sainz Garcia</li>
    </ul>
  </main>
</body>
```

```

</main>

<footer>Copyright 2012 B.lawson</footer>

</body>
</html>

```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/grouping-content.html#the-main-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\html5-main.xhtml

Text-Level semantics

a

"If the a element has an href attribute, then it represents a hyperlink (a hypertext anchor) labeled by its contents.

If the a element has no href attribute, then the element represents a placeholder for where a link might otherwise have been placed, if it had been relevant, consisting of just the element's contents."

If a site uses a consistent navigation toolbar on every page, then the link that would normally link to the page itself could be marked up using an a element:

```

<nav>
<ul>
<li> <a href="/">Home</a> </li>
<li> <a href="/news">News</a> </li>
<li> <a>Examples</a> </li>
<li> <a href="/legal">Legal</a> </li>
</ul>
</nav>

```

em

"stress emphasis of its contents".

strong

"strong importance, seriousness, or urgency for its contents".

small

" side comments such as small print ... features disclaimers, caveats, legal restrictions, or copyrights ... the small element should not be used for extended spans of text, such as multiple paragraphs, lists, or sections of text. It is only intended for short runs of text."

```

<dl>
<dt>Single room
<dd>199 € <small>breakfast included, VAT not included</small>
<dt>Double room
<dd>239 € <small>breakfast included, VAT not included</small>
</dl>

```

s

[jlb: 'substituted'] "contents that are no longer accurate or no longer relevant ... The s element is not appropriate when indicating document edits"

```
<p>Buy our Iced Tea and Lemonade!</p>
<p><s>Recommended retail price: $3.99 per bottle</s></p>
<p><strong>Now selling for just $2.99 a bottle!</strong></p>
```

cite

represents a reference to a creative work. It must include the title of the work or the name of the author (person, people or organization) or an URL reference.

```
<!-- Inline citation of author -->
<p>In the words of <cite>Charles Bukowski</cite> - <q>An intellectual says a simple thing in
a hard way. An artist says a hard thing in a simple way.</q></p>

<!-- Blockquote citation of author, with link -->
<blockquote>
The people recognize themselves in their commodities; they find their
soul in their automobile, hi-fi set, split-level home, kitchen equipment.
- <cite><a href="http://en.wikipedia.org/wiki/Herbert_Marcuse">Herbert Marcuse</a></cite>
</blockquote>

<!-- Inline citation of title, example comes from spec -->
<p><cite>Universal Declaration of Human Rights</cite>, United Nations,
December 1948. Adopted by General Assembly resolution 217 A (III).</p>
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantic.html#the-cite-element>

q

Represents content quoted from another source. The source may be fictional, as when quoting characters in a novel or screenplay.

```
<!-- External source -->
<p>The W3C page <cite>About W3C</cite> says the W3C's
mission is <q>To lead the
World Wide Web to its full potential by developing protocols and
guidelines that ensure long-term growth for the Web</q>. I
disagree with this mission.</p>

<!-- Fictional source -->
<p>The man said <q>Things that are impossible just take
longer</q>. I disagreed with him.</p>
```

(W3C HTML5.3, 2017), "4.5.7. The q element", <https://www.w3.org/TR/html5/textlevel-semantic.html#the-q-element>

If there is an address for a source this may be cited in the cite attribute. But the cite attribute is primarily intended for private use (e.g., by server-side scripts collecting statistics about a site's use of quotations), not for readers.

```
<p>The W3C page <cite>About W3C</cite> says the W3C's
mission is <q cite="https://www.w3.org/Consortium/">To lead the
World Wide Web to its full potential by developing protocols and
guidelines that ensure long-term growth for the Web</q>. I
disagree with this mission.</p>
```

(W3C HTML5.3, 2017), "4.5.7. The q element", <https://www.w3.org/TR/html5/textlevel-semantic.html#the-q-element>

The use of q elements to mark up quotations is entirely optional; using explicit quotation punctuation without q elements is just as correct.

```
<p>The man said "Things that are impossible just take
```

```
longer". I disagreed with him.</p>
```

However, there may be CSS reasons to prefer `<q>` over explicit quotations marks (e.g. if internationalisation entails that the quote marks need to change depending on the region of the reader).

(W3C HTML5.3, 2017), "4.5.7. The *q* element", <https://www.w3.org/TR/html5/textlevel-antics.html#the-q-element>

dfn (defining term)

dfn represents the authoritative label for a definition. E.g. It references the "bird" part of a definition including "animal that flies". Some points:

1. The paragraph, description list group, or section that is the nearest ancestor of the dfn element must also contain the definition(s) for the term given by the dfn element.
2. "If the dfn element has a title attribute, then the exact value of that attribute is the term being defined."
3. "Otherwise, if it contains exactly one element child node and no child Text nodes, and that child element is an abbr element with a title attribute, then the exact value of that attribute is the term being defined."

```
<!-- When used in a description list -->
<dl>
  <dt><dfn>Apartment</dfn>, n.</dt>
  <dd>An execution context grouping one or more threads with one or
  more COM objects.</dd>
  <dt><dfn>Flat</dfn>, n.</dt>
  <dd>A deflated tire.</dd>
  <dt><dfn>Home</dfn>, n.</dt>
  <dd>The user's login directory.</dd>
</dl>

<!-- When used in paragraph -->
<p>A <dfn>Bird</dfn> is an animal that flies</p>.

<!-- When used with an abbr (with title attribute). -->
<p>
  The <dfn><abbr title="Garage Door Opener">GDO</abbr></dfn>
  is a device that allows off-world teams to open the iris.
</p>
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-antics.html#the-dfn-element>
See (in this document) [dl](#)

abbr

`abbr` represents and abbreviation or acronym. Some points:

- It may optionally have an expansion in the title attribute.
- The title element, if used, must not contain anything apart from an expansion.

```
<p>
  The
  <dfn id="whatwg">
    <abbr title="Web Hypertext Application Technology Working Group">WHATWG</abbr>
  </dfn>
  is a loose unofficial collaboration of Web browser manufacturers and
  interested parties who wish to develop new technologies designed to
  allow authors to write and deploy Applications over the World Wide
  Web.
</p>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantic.html#the-abbr-element>

acronym (obsolete)

`acronym` is obsolete. Instead use `abbr`.

(Mozilla, 2016) <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/acronym>

data

`data` represents its human readable contents. The `data` element has a `value` attribute which must be present and represent the machine readable equivalent for the human readable contents. Points:

4. For datetime values don't use the `data` element, use the `time` element.
5. There are `data-*` attributes available for custom data stuffing.

```
<!-- The following example displays product names but also associates each name with its UPC
code -->
<p>New Products</p>
<ul>
<li><data value="3967381398">Mini Ketchup</data></li>
<li><data value="3967381399">Jumbo Ketchup</data></li>
<li><data value="3967381400">Mega Jumbo Ketchup</data></li>
</ul>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantic.html#the-data-element>
 Example from: <https://developer.mozilla.org/en/docs/Web/HTML/Element/data>

time (datetime type)

The `time` represents a date, time, datetime, time-zone offset or duration.

Datetime type	Example
Month	<code><time>2011-11</time></code>
Date	<code><time>2011-11-12</time></code>
Yearless date	<code><time>11-12</time></code>
Time	<code><time>14:54</time></code> <code><time>14:54:39</time></code> <code><time>14:54:39.929</time></code>
Date and time	<code><time>2011-11-12T14:54</time></code> <code><time>2011-11-12T14:54:39</time></code> <code><time>2011-11-12T14:54:39.929</time></code> <code><time>2011-11-12 14:54</time></code> <code><time>2011-11-12 14:54:39</time></code> <code><time>2011-11-12 14:54:39.929</time></code>
Timezone offset	<code><time>Z</time></code> <code><time>+0000</time></code> <code><time>+00:00</time></code> <code><time>-0800</time></code> <code><time>-08:00</time></code>

Date, time, and timezone	<pre><time>2011-11-12T14:54Z</time> <time>2011-11-12T14:54+0000</time> <time>2011-11-12T06:54:39-0800</time> <time>2011-11-12T06:54:39-08:00</time> <time>2011-11-12 14:54Z</time> <time>2011-11-12 14:54:39.929+00:00</time> <time>2011-11-12 06:54:39.929-0800</time></pre>
Week	<pre><time>2011-W46</time></pre>
Year	<pre><time>2011</time></pre>
Duration	<pre><time>PT4H18M3S</time> <time>4h 18m 3s</time></pre>

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-time-element>

If the `datetime` attribute is also present the attribute must contain a ISO 8601 formatted machine-readable datetime (date, time, datetime, time-zone offset or duration).

```
<div class="vevent">
  <a class="url" href="http://www.web2con.com/">http://www.web2con.com/</a>
  <span class="summary">Web 2.0 Conference</span>:
  <time class="dtstart" datetime="2005-10-05">October 5</time> -
  <time class="dtend" datetime="2005-10-07">7</time>,
  at the <span class="location">Argent Hotel, San Francisco, CA</span>
</div>
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-time-element>

If the `datetime` attribute is not present then the text value of the element must be in that ISO 8601 format.

```
<div class="vevent">
  <a class="url" href="http://www.web2con.com/">http://www.web2con.com/</a>
  <span class="summary">Web 2.0 Conference</span>:
  <time class="dtstart">2005-10-05</time> -
  <time class="dtend">2005-10-07</time>,
  at the <span class="location">Argent Hotel, San Francisco, CA</span>
</div>
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-time-element>

code

The `code` element represents a string that a computer would recognize. For example, an XML element name, a file name, or the lines of a computer program.

```
<!-- Referencing computer code inline -->
<p>
  The <code>code</code> element represents a fragment of computer
  code.
</p>

<p>
  When you call the <code>activate()</code> method on the
  <code>robotSnowman</code> object, the eyes glow.
</p>
```

```
<p>
  The example below uses the <code>begin</code> keyword to indicate
  the start of a statement block. It is paired with an <code>end</code>
  keyword, which is followed by the <code>.</code> punctuation character
  (full stop) to indicate the end of the program.
</p>

<!-- Referencing computer code as a block -->
<pre><code class="language-pascal">var i: Integer;
  begin
    i := 1;
  end.
</code></pre>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-code-element>

var

var represents a variable. This could be in the context of programming, mathematics, or "a term used as a placeholder in text".

```
<!-- Mathematic variable in prose -->
<p>If there are <var>n</var> pipes leading to the ice
cream factory then I expect at <em>least</em> <var>n</var>
flavors of ice cream to be available for purchase!</p>

<!-- Mathematics variable in prose, referencing variable in mathML element -->
<figure>
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <mi>a</mi>
    <mo>=</mo>
    <msqrt>
      <msup><mi>b</mi><mn>2</mn></msup>
      <mi>+</mi>
      <msup><mi>c</mi><mn>2</mn></msup>
    </msqrt>
  </math>
  <figcaption>
    Using Pythagoras' theorem to solve for the hypotenuse <var>a</var> of
    a triangle with sides <var>b</var> and <var>c</var>
  </figcaption>
</figure>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-var-element>

samp & kbd

samp represents (sample) output from a program.

kbd represents user input, typically keyboard input but it may also be voice input (for example).

When the kbd element is nested inside a samp element, it represents the input as it was echoed by the system

```
<pre><samp>PS <kbd>$env:Path -split ';' | sort</kbd>

C:\Program Files (x86)\Android\android-sdk\PLATFORM-TOOLS
C:\Program Files (x86)\Android\android-sdk\TOOLS
C:\PROGRAM FILES (X86)\FCIV\
C:\Program Files (x86)\GNU\GnuPG\pub
C:\PROGRAM FILES (X86)\INTEL\ICLS CLIENT\
...
</samp>
</pre>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-samp-element>

sub and sup

i

i represents an alternative voice or mode. For example, "a taxonomic designation, a technical term, an idiomatic phrase from another language, transliteration, a thought, or a ship name in Western texts". Use as a last resort, when another element won't do.

```
<p>The <i class="taxonomy">Felis silvestris catus</i> is cute.</p>
<p>The term <i>prose content</i> is defined above.</p>
<p>There is a certain <i lang="fr">je ne sais quoi</i> in the air.</p>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-i-element>

b

The **b** element represents a span of text to which attention is being drawn for utilitarian purposes without conveying any extra importance and with no implication of an alternate voice or mood. [For example] ... as key words in a document abstract, product names in a review, actionable words in interactive text-driven software, or an article lede.

Use as a last resort, when another element won't do.

```
<!-- Markup words without making them important -->
<p>The <b>frobonitor</b> and <b>barbinator</b> components are fried.</p>

<!-- Text based adventure example -->
<p>You enter a small room. Your <b>sword</b> glows
brighter. A <b>rat</b> scurries past the corner wall.</p>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-b-element>

u

The **u** element represents a span of text with an unarticulated, though explicitly rendered, non-textual annotation, such as labeling the text as being a proper name in Chinese text (a Chinese proper name mark), or labeling the text as being misspelt.

Use as a last resort, when another element won't do.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-u-element>

mark

The **mark** element represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.

```
<p>This is the <mark>most sublime</mark> of possibilites.</p>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantics.html#the-mark-element>

ruby, rb, rt, rtc, rp

The [ruby](#) element allows one or more spans of phrasing content to be marked with ruby annotations. Ruby annotations are short runs of text presented alongside base text, primarily used in East Asian typography as a guide for pronunciation or to include other annotations.

bdi (Bi Directional Isolation)

Isolates text from it's surrounding so that it doesn't taken on it's own directionality. Jlb: we can think if "bdi"standing for Bi-Directional Isolation.

```
<!-- Sever side scripting generates usernames with their numbe of posts: -->

<!-- If bdi where not used on the username, Arabic usernames would fuck up the surrounding
text as follows -->
<ul>
  <li>User <bdi>jcranmer</bdi>: 12 posts.</li>
  <li>User <bdi>hober</bdi>: 5 posts.</li>
  <li>User 3: إيانposts.</li>
</ul>

<!-- With bdi an Arabic usernames preserves the order of the surrounding text. -->
<ul>
  <li>User <bdi>jcranmer</bdi>: 12 posts.</li>
  <li>User <bdi>hober</bdi>: 5 posts.</li>
  <li>User <bdi>إيان</bdi>: 3 posts.</li>
</ul>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantic.html#the-bdi-element>

bdo (Bi Directional Override)

Override default directionality for children and expliclity specifiy directionality.

```
<!-- Override default directionality for children and expliclity specifiy directionality -->
<ul>
  <li>User <bdi>jcranmer</bdi>: 12 posts.</li>
  <li>User <bdi>hober</bdi>: 5 posts.</li>
  <li>User <bdi><bdo dir="rtl">إيان</bdo></bdi>: 3 posts.</li>
</ul>

// Ouput (as desired)
...
User [Arabic name]: 3 posts.
```

Without bdo being enclosed by bdi the surrounding text is fucked up.

```
<ul>
  <li>User <bdi>jcranmer</bdi>: 12 posts.</li>
  <li>User <bdi>hober</bdi>: 5 posts.</li>
  <li>User <bdo dir="rtl">إيان</bdo>: 3 posts.</li>
</ul>

// Output (fucked up surrounding text).
User 3: إيان posts.
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantic.html#the-bdo-element>

span

The span element doesn't mean anything on its own, but can be useful when used together with the global attributes, e.g. class, lang, or dir. It represents its children.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantic.html#the-span-element>

br

`br` represents a line break. For backwards compatibility [insert a space](#) before closing the tag.

```
<p>
  330 Sandymount Rd<br />
  Sandymount 9077<br />
  New Zealand
</p>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantic.html#the-br-element>

wbr

`wbr` represents a line break opportunity. That is, a place where a line break should be made if necessary.

```
<p>
  So then he pointed at the tiger and screamed
  "there<wbr />is<wbr />no<wbr />way<wbr />you<wbr />are<wbr />ever<wbr />going<wbr />to<wbr />
  </p>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/text-level-semantic.html#the-wbr-element>

Links

Links are a conceptual connection between two resources, one of which is the current document.

(Web Hypertext Application Technology Working Group (WHATWG), n.d. *HTML Standard, Developer's Edition*)

There are two kinds of links: links to external resources; and hyperlinks.

```
<!-- Links to external resources -->
<link rel="stylesheet" href="html5-reset.org.jlb.css" />

<!-- Hypelink -->
<p>Lorem <a href="html5-play.xhtml" download="download">Play</a></p>
```

(Web Hypertext Application Technology Working Group (WHATWG), n.d. *HTML Standard, Developer's Edition*)

a and area

With the 'download' attribute you can force a resource to download rather than be navigated to.

The attribute may have a value; the value, if any, specifies the default file name that the author recommends for use in labeling the resource in a local file system.


```
<p>Lorem <a href="html5-play.xhtml" download="play.xhtml">Play</a></p>
```

(Web Hypertext Application Technology Working Group (WHATWG), n.d. HTML Standard, Developer's Edition)
<https://html.spec.whatwg.org/dev/links.html#links-created-by-a-and-area-elements>

In cross-origin situations, the `download` attribute has to be combined with the `Content-Disposition` HTTP header, specifically with the `attachment` disposition type, to avoid the user being warned of possibly nefarious activity.

(Web Hypertext Application Technology Working Group (WHATWG), n.d. HTML Standard, Developer's Edition)
<https://html.spec.whatwg.org/dev/links.html#downloading-resources>

The `ping` attribute, if present, gives the URLs of the resources that are interested in being notified if the user follows the hyperlink. The value must be a set of space-separated tokens.

```
// todo: setup example where the reponse page writes to database or otherwise register the ping
<p>Lorem <a href="html5-play.xhtml" ping="response.php?txtCool=PlayPing">Play Ping</a></p>
```

(Web Hypertext Application Technology Working Group (WHATWG), n.d. HTML Standard, Developer's Edition)
<https://html.spec.whatwg.org/dev/links.html#links-created-by-a-and-area-elements>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\html5-links.xhtml

Thus, while it is possible to track users without this feature [ping], authors are encouraged to use the ping attribute so that the user agent can make the user experience more transparent.

(Web Hypertext Application Technology Working Group (WHATWG), n.d. HTML Standard, Developer's Edition)
<https://html.spec.whatwg.org/dev/links.html#downloading-resources>

Other attributes: `rel`; `hreflang`; `type`; `referrerpolicy`.

(Web Hypertext Application Technology Working Group (WHATWG), n.d. HTML Standard, Developer's Edition)
<https://html.spec.whatwg.org/dev/links.html#links-created-by-a-and-area-elements>

Api

<code>hyperlink . toString()</code>	
<code>hyperlink . href</code>	
<code>hyperlink . origin</code>	
<code>hyperlink . protocol</code>	
<code>hyperlink . host</code>	
<code>hyperlink . hostname</code>	
<code>hyperlink . port</code>	
<code>hyperlink . pathname</code>	
<code>hyperlink . search</code>	
<code>hyperlink . hash</code>	

(Web Hypertext Application Technology Working Group (WHATWG), n.d. HTML Standard, Developer's Edition)
<https://html.spec.whatwg.org/dev/links.html#links-created-by-a-and-area-elements>

email linking

For email linking use:

```
<a
href="mailto:john%40cool.org?subject=About%20cool%20stuff&amp;body=First%20line%0D%0ASecond%
20Line">Email with subject and multiline body</a>
```

Use ? & syntax. But encode & as & Url safe encoding %20 for spaces; %0A for new line. "subject=" and "body="

Use the "Basic mailto URI composer" <http://shadow2531.com/js/maillouri.html> with Encode & as &

Edits

ins

ins representation a text insertion into a document. Don't cross implied paragraph boundaries when so inserting ("authors are strongly encouraged to always mark up all paragraphs with the p element").

```
<aside>
  <ins datetime="2005-03-16 00:00Z">
    <p> I like fruit. </p>
  </ins>
  <ins datetime="2005-03-16 00:00Z">
    Apples are <em>tasty</em>.
  </ins>
  <ins datetime="2007-12-19 00:00Z">
    So are pears.
  </ins>
</aside>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/edits.html#the-ins-element>

del

del represents a deletion from the document. Don't cross implied paragraph boundaries when so deleting ("authors are strongly encouraged to always mark up all paragraphs with the p element").

```
<ul>
  <li>Empty the dishwasher</li>
  <li><del datetime="2009-10-11T01:25-07:00">Watch Walter Lewin's lectures</del></li>
  <li><del datetime="2009-10-10T23:38-07:00">Download more tracks</del></li>
  <li>Buy a printer</li>
</ul>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/edits.html#the-del-element>

Embedded content

picture, source, img, Images

picture

The picture element is a container which provides multiple sources to its contained img element to allow authors to declaratively control or give hints to the user agent about which image resource to use, based on the screen pixel density, viewport size, image format, and other factors.

(Web Hypertext Application Technology Working Group (WHATWG), n.d. HTML Standard, Developer's Edition)
<https://html.spec.whatwg.org/dev/embedded-content.html#the-picture-element>

The picture element is somewhat different from the similar-looking video and audio elements. While all of them contain source elements, the source element's src attribute has no meaning when the element is nested within a picture element, and the resource selection algorithm is different. Also, the picture element itself does not display anything; it merely provides a context for its contained img element that enables it to choose from multiple URLs.

(Web Hypertext Application Technology Working Group (WHATWG), n.d. HTML Standard, Developer's Edition)
<https://html.spec.whatwg.org/dev/embedded-content.html#the-picture-element>

source

img

img represents an image.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-img-element>

Except where otherwise specified, the alt attribute must be specified and its value must not be empty

```

```

(W3C HTML5.3, 2017) <https://www.w3.org/TR/html53/semantics-embedded-content.html#the-img-element>
 (Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Examples\CssExamples\imageLoadTimeTest.*

Image techniques

Basics

The figure and figcaption elements can be used to associate the longer text alternative with an image. In this case the alt attribute is supplied with a briefer label.

```
<figure>
  
  <figcaption>
    <b>Figure 1.</b>
    Distribution of Articles by Journal Category. Pie chart: Language=68%,
    Education=14% and Science=18%.
  </figcaption>
</figure>
```

" The advantage of this method over the previous example is that the text alternative is available to all users at all times. It also allows structured mark up to be used in the text alternative, where as a text alternative provided using the alt attribute does not."

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-img-element>

JLB: Explicitly specify the dimensions (width and height) of your images in XHTML not CSS.

```

```

There are three ways you can deal with dimensioning images: not setting them; setting them in CSS; setting them in XHTML.

Setting dimensions in XHTML is the only way to ensure the page layout remains still as downloading occurs.

It is often claimed that explicitly setting the dimensions (in XHTML) gives speedier rendering of the page (eg Castro > P 106.) However, experiment proves this to be negligible.

(Castro, 2003)

(Bentley, n.d.) CSSExamples > *imageLoadTimeTest*.*

Image Types

Logos & Drawing	.gif
Photos	.jpg, .jpeg
Manipulation	.bmp
Save with Transparency	.gif (or png-8)

Image Size

We should aim to get images down to 30KB or less.

Medium: width=285px or height=190px. That is, a minimum dimension of width=285px, height=190px, while preserving aspect ratio. Eg a 200px wide and 500px high image is fine if that what the aspect ratio implies.

So one could aim to have standard images (where image merely complement's text rather than is the centre of attention) with a minimum dimension of width=203px, height=152px, while preserving aspect ratio.

Medium: To good effect the standard JPEG image dimensions on <http://news.bbc.co.uk> are width=203px, height=152px (about 13KB).

<http://www.abc.net.au/news/stories/2007/09/24/2041168.htm>: Width=285px, Height=190px, (about 15 KB).

Large: Width= 670px, height = 500px. Good to aim for about 70 KB.

Compressing JPEG procedure:

1. Use IfranView
2. Image > Resize/Resample > Half image button twice (or other amount). (Ctrl + R)
3. Select Area to crop with mouse.
4. Edit > Crop Selection (Ctrl + Y).
5. Save.
6. Save As:
 - a. Type: JPG
 - b. Save Quality: 85% (Not much loss but lots of compression).
 - c. Save as Progressive

Other tips to compress JPEGs.

- Reducing colors from 16.7 Million colours (24 bit) to 256 colours (8 bit) does not help with JPEGs.

Compressing GIFS.

1. Use Photoshop.
2. Open Image 96 pixels/Inch and at desired dimension.
3. Save as Web Page. GIF settings:
 - a. Colours: 8
 - b. Colour Reduction System: Selective
 - c. Dither Algorithm: Diffusion
 - d. Dither %: 100%
 - e. Transparency: No.
 - f. Interlaced: Yes.

Image Selection

Images can be swamped out depending on the user's environment (screen size, pixel density, etc). Four kinds of bases can be used for image selection:

1. Viewport-based selection.
2. Device-pixel-ration-based selection.
3. Art-direction-based selection.
4. Image-format-based selection.

Fixed width images: **device-pixel-ration-based selection**. Change the image to get the same image with a different pixel density. For when the image will remain that same width regardless of screen size (e.g. site logo or a profile image).

```

```

(Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\image-selection-device-pixel-ratio-based.xhtml

(Jing, 2015. Using Responsive Images (Now)) <https://alistapart.com/article/using-responsive-images-now>

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#embedded-content-introduction>

Change the image size, and/or the image, as the browser viewport size changes using **Viewport-based selection**. How to:

- We use CSS viewport percentage lengths (`vw`, `vh`, `vmin`, `vmax`).
- Alter the size: The `sizes` attribute specifies that for a given viewport size (a media condition) display the image as a percentage of the viewport. E.g. "(max-width: 960px) 75vw" means for viewport sizes up to 960px display the image at 75% of the viewport.
- Change the image: the `srcset` attribute specifies an image to display.

```


```

(Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\image-selection-viewport-based.xhtml

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#embedded-content-introduction>

(Jing, 2015. Using Responsive Images (Now)) <https://alistapart.com/article/using-responsive-images-now>

(W3C, 2016. CSS Values and Units Module Level 3, W3C Candidate Recommendation) <https://www.w3.org/TR/css3-values/#viewport-relative-lengths>

Change the image size, and/or the image, as the browser viewport size changes using **art-direction-base selection**. How to:

- Use a `picture` element with `source` elements and a single `img` element.
- The browser will choose the first `source` element with a matching `media` attribute.
- If the browser does not understand `picture` or `source` then `img` will be used.
- The size of the image can be controlled with CSS media queries.

```

<picture>
  <source media="(min-width: 1100px)" srcset="images/mount-everest-5285w-full-resolution.jpeg" />
  <source media="(min-width: 750px)" srcset="images/mount-everest-1366w.jpg" />
  
</picture>

/* CSS */
img { width: 80vw; height: 100px }
@media (min-width: 750px) { img { width: 90vw; height: 150px } }
@media (min-width: 1100px) { img { width: 95vw; height: 200px } }

```

(Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\image-selection-art-direction-based.xhtml

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#embedded-content-introduction>

Change image based browser image type support: **image-format-based selection**. Details:

- There are new image formats kicking about: WebP; JPEG XR; JPEG 2000;
- Browsers have varying levels of support for these formats.
- We can use `picture`, `source`, and `img` elements with the `type` attribute.
- The browser chooses the first image that matches the `type` attribute.

```

<picture>
  <source type="image/webp" srcset="images/tree.webp" />
  
</picture>

// At 2017-12-11
// Firefox displays mount-everest-640w.jpg
// Chrome displays tree.webp (and it takes on the dimensions specified in the img element

```

(Deveria, n.d.) <https://caniuse.com/#search=image%20format>
 (W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#embedded-content-introduction>
 (Jing, 2015. Using Responsive Images (Now)) <https://alistapart.com/article/using-responsive-images-now>

Requirements for providing text to act as an alternative for images

Except where otherwise specified, the alt attribute must be specified and its value must not be empty; the value must be an appropriate functional replacement for the image.

```

```

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#general-guidelines>

The longdesc attribute can be used to reference a fuller text alternative. It can be useful when working in conjunction with the alt attribute.

```
// "Example 28. In the case where an image repeats the previous paragraph in graphical form.
The alt attribute content labels the image and the longdesc attribute identifies it as a
description."
<p id="graph7">According to a recent study Firefox has a 40% browser share,
Internet Explorer has 25%, Chrome has 25%, Safari has 6% and Opera has 4%.</p>
<p></p>
```

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#graphical-representations-charts-diagrams-graphs-maps-illustrations>

Kinds of longdesc.

```
<!-- The description is somewhere on the same page as the image -->


<!-- The description is a page on its own -->


<!-- The description is one of several within an external page -->


<!-- The description is included in a data: URI -->

```

(W3C, 2015) <https://www.w3.org/TR/html-longdesc/>

iframe, embed, object, param

iframe

iframe ... represents a nested browsing context.

```
<p>We're not scared of you! Here is your content, unedited:</p>
<iframe sandbox src="http://usercontent.example.net/getusercontent.cgi?id=12193"></iframe>
```

!b: iframes generally server (x)html, or xml.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-iframe-element>

The `sandbox` attribute, when specified, enables a set of extra restrictions on any content hosted by the `iframe`

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-iframe-element>

Potentially hostile files should not be served from the same server as the file containing the `iframe` element ... To limit the damage that can be caused by hostile HTML content, it should be served from a separate dedicated domain.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-iframe-element>

embed

`embed` ... provides an integration point for an external (typically non-HTML) application or interactive content.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-embed-element>

Any namespace-less attribute other than `name`, `align`, `hspace`, and `vspace` may be specified on the `embed` element, so long as its name is XML-compatible and contains no uppercase ASCII letters. These attributes are then passed as parameters to the plugin.

```
<!-- Pass thhe plugin a parameter "quality" with the value "high" -->
<embed src="catgame.swf" quality="high" />
```

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-embed-element>

object

`object` ... can represent an external resource, which, depending on the type of the resource, will either be treated as an image, as a nested browsing context, or as an external resource to be processed by a plugin.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-object-element>

Authors who reference resources from other origins that they do not trust are urged to use the `typemustmatch` attribute.

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-object-element>

Examples.

```
<!-- A Java applet -->
<figure>
  <object type="application/x-java-applet">
    <param name="code" value="MyJavaClass" />
    <p>You do not have Java available, or it is disabled.</p>
  </object>
```



```

<figcaption>My Java Clock</figcaption>
</figure>

<!-- Embed HTML -->
<figure>
  <object data="clock.html"></object>
  <figcaption>My HTML Clock</figcaption>
</figure>

<!-- Shockwave Flash. Fallback to video; fallback, further, to a link to the video -->
<p>Look at my video:
  <object type="application/x-shockwave-flash">
    <param name="movie" value="http://video.example.com/library/watch.swf" />
    <param name="allowfullscreen" value="true" />
    <param name="flashvars" value="http://video.example.com/vids/315981" />
    <video controls="controls" src="http://video.example.com/vids/315981">
      <a href="http://video.example.com/vids/315981">View video</a>.
    </video>
  </object>
</p>

```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-object-element>

param

`param` ... defines parameters for plugins invoked by `object` elements. It does not represent anything on its own.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-param-element>

Media Elements (video and audio)

The media elements are `video` and `audio`.

video

Own hosted content

`video` ... is used for playing videos or movies, and audio files with captions.

```

<video src="video/ronaldjenkees-a-fun-beat-from-outer-space.mp4" controls="controls"
width="480" height="360">https://html.spec.whatwg.org/dev/iframe-embed-object.html
  <p>Your browser does not support video playback.</p>
  <a href="video/ronaldjenkees-a-fun-beat-from-outer-space.mp4">Download to view video</a>.
</video>

```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014; Web Hypertext Application Technology Working Group (WHATWG), n.d. *HTML Standard, Developer's Edition*) <https://www.w3.org/TR/html5/embedded-content-0.html#the-video-element>

Content may be provided inside the `video` element; it is intended for older Web browsers which do not support video, so that legacy video plugins can be tried, or to show text to the users of these older browsers informing them of how to access the video contents.

JLB: Include the `controls` attribute, if you aren't going to script your own.

(Bentley, n.d.) 2016-03-25 09:39

(W3C, 2017. *HTML 5.2*) <https://www.w3.org/TR/html52/semantics-embedded-content.html#user-interface>

Youtube

Youtube uses iframes. To obtain the relevant code visit the youtube website, click share, and embed, to get the relevant code. Then ...

1. Remove the frameborder attribute.
2. Add an "allowfullscreen" attribute value (to conform to xml).

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/ompApz4IJMA"
allowfullscreen="allowfullscreen"></iframe>
```

(Bentley, n.d.) 2016-04-18 12:03 C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-example.xhtml

audio

An audio element represents a sound or audio stream.

```
<figure>
  <audio src="audio/FatboySlim-RightHereRightNow-RocknessRemix.mp3" controls="controls" >
    <p>Your browser does not support the <code>audio</code> element.</p>
    <a href="audio/FatboySlim-RightHereRightNow-RocknessRemix.mp3">Download audio
file</a>.
  </audio>
  <figcaption>Fatboy Slim - Right here, right now - Rockness remix</figcaption>
</figure>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-audio-element>
(Mozilla, 2016) https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML5_audio_and_video

track (text track, mainly for video)

Overview

The track element allows authors to specify explicit external timed text tracks for media elements.

track element kind attribute:

kind attribute value	Description
subtitles	Transcription or translation of the dialogue, suitable for when the sound is available but not understood (e.g. because the user does not understand the language of the media resource's audio track). Overlaid on the video.
captions	Transcription or translation of the dialogue, sound effects, relevant musical cues, and other relevant audio information, suitable for when sound is unavailable or not clearly audible (e.g. because it is muted, drowned-out by ambient noise, or because the user is deaf). Overlaid on the video; labeled as appropriate for the hard-of-hearing.
descriptions	Textual descriptions of the video component of the media resource, intended for audio synthesis when the visual component is obscured, unavailable, or not usable (e.g. because the user is interacting with the application without a screen while driving, or because the user is blind). Synthesized as audio.

chapters	Chapter titles, intended to be used for navigating the media resource. Displayed as an interactive (potentially nested) list in the user agent's interface.
metadata	Tracks intended for use from script. Not displayed by the user agent.

```
<video src="video/ronaldjenkees-a-fun-beat-from-outer-space.mp4" controls="controls">
  <track src="html5-track.vtt" kind="captions" default="default" label="English captions for
the hard of hearing" />
  Your browser does not support the <code>video</code> element.
</video>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/embedded-content-0.html#the-track-element>

WebVTT

A track element will refer to a file written in "WebVTT: The Web Video Text Tracks Format".

(W3C, 2016. *WebVTT: The Web Video Text Tracks Format*) <https://w3c.github.io/webvtt/>

A web video text track, WebVTT, formatted file. Save with *.vtt extension.

```
WEBVTT

00:00:00.000 --> 00:00:01.000
Hello Youtubes.

NOTE This is a comment. Precede with blank line. End with blank line.

NOTE
Multi line
comments, cool.

NOTE Identify a cue

mycue
00:00:01.000 --> 00:00:02.000
Everything is goodz.

NOTE Style block and id referencing

STYLE
::cue(#mycue) { color: blue; } /*Not supported yet?*/

NOTE Multiline cue

00:00:02.000 --> 00:00:03.000
- You are great.
- I know I am great.

NOTE voice spans

00:00:03.000 --> 00:00:04.000
<v Ronald>Here is my music.</v>

NOTE hours are optional

00:04.000 --> 00:05.000
Hours are optional
```

(W3C, 2016. *WebVTT: The Web Video Text Tracks Format*) <https://w3c.github.io/webvtt/>
C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\html5-track.vtt

Styling the text in a WebVTT when played through a video.

```

<!doctype html>
<html>
<head>
  <title>Styling WebVTT cues</title>
  <style type="text/css">
    /*  */
    /** Doesn't work in FF at 2016-06-17 19:43.
    See http://caniuse.com/#feat=webvtt */
    video::cue {
      margin-bottom: 3em;
      background-image: linear-gradient(to bottom, dimgray, lightgray);
      color: papayawhip;
    }

    video::cue(b) {
      color: peachpuff;
    }
  /* ]]&gt; */
&lt;/style&gt; &lt;/head&gt;
&lt;body&gt;
  &lt;video src="video/ronaldjenkees-a-fun-beat-from-outer-space.mp4" controls="controls"&gt;
    &lt;track src="html5-track.vtt" kind="captions" default="default" label="English captions
for the hard of hearing" /&gt;
    Your browser does not support the &lt;code&gt;video&lt;/code&gt; element.
  &lt;/video&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>
</div>
<div data-bbox="125 391 419 406" data-label="Text">
<p>Doesn't work in FF at 2016-06-17 19:43.</p>
</div>
<div data-bbox="125 421 733 449" data-label="Text">
<p>(W3C, 2016. <i>WebVTT: The Web Video Text Tracks Format</i>) <a href="https://w3c.github.io/webvtt/#styling">https://w3c.github.io/webvtt/#styling</a><br/>
<a href="http://caniuse.com/#feat=webvtt">http://caniuse.com/#feat=webvtt</a></p>
</div>
<div data-bbox="115 466 171 479" data-label="Section-Header">
<h2>source</h2>
</div>
<div data-bbox="129 496 867 525" data-label="Text">
<p>The source element allows authors to specify multiple alternative media resources for media elements.</p>
</div>
<div data-bbox="125 541 835 596" data-label="Text">
<pre>
&lt;video controls=""&gt;
  &lt;source src="video/ronaldjenkees-a-fun-beat-from-outer-space.ogv" type="video/ogv" /&gt;
  &lt;source src="video/ronaldjenkees-a-fun-beat-from-outer-space.mp4" type="video/mp4" /&gt;
  Your browser does not support the &lt;code&gt;video&lt;/code&gt; element.
&lt;/video&gt;
</pre>
</div>
<div data-bbox="125 609 859 651" data-label="Text">
<p>(W3C HTML5, 2014. <i>HTML5, A vocabulary and associated APIs for HTML and XHTML</i>, W3C Recommendation 28 October 2014) <a href="https://www.w3.org/TR/html5/embedded-content-0.html#the-source-element">https://www.w3.org/TR/html5/embedded-content-0.html#the-source-element</a><br/>
(Mozilla, 2016) <a href="https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML5_audio_and_video">https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML5_audio_and_video</a></p>
</div>
<div data-bbox="115 666 445 683" data-label="Section-Header">
<h2>Media element (video and audio) attributes</h2>
</div>
<div data-bbox="115 697 763 729" data-label="Text">
<p>Media element common attributes <code>src</code>, <code>crossorigin</code>, <code>preload</code>, <code>autoplay</code>, <code>loop</code>, <code>muted</code>, and <code>control</code>.</p>
</div>
<div data-bbox="125 743 828 758" data-label="Text">
<p>(W3C, 2017. <i>HTML 5.2</i>) <a href="https://www.w3.org/TR/html52/semantics-embedded-content.html#the-media-elements">https://www.w3.org/TR/html52/semantics-embedded-content.html#the-media-elements</a></p>
</div>
<div data-bbox="115 773 663 790" data-label="Section-Header">
<h2>AudioTrackList (and AudioTrack) and VideoTrackList (and VideoTrack)</h2>
</div>
<div data-bbox="115 804 764 820" data-label="Text">
<p>A media resource (video or audio) can have multiple audio and video tracks.</p>
</div>
<div data-bbox="125 835 843 901" data-label="Text">
<pre>
&lt;script&gt;
  // tries to load the video, and, once it is loaded, checks to see if there is a sign-
language track available. If there is, it also displays that track. Both tracks are just
placed in the given container
  function loadVideo(url, container) {
    var video = document.createElement('video');
</pre>
</div>
<div data-bbox="115 938 320 955" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div>
<div data-bbox="363 938 663 956" data-label="Page-Footer">John Bentley of <a href="http://www.softmake.com.au">www.softmake.com.au</a></div>
<div data-bbox="702 938 822 956" data-label="Page-Footer">Page 52 of 103</div>
```

```

video.src = url;
video.autoplay = true;
video.controls = true;
container.appendChild(video);
video.videoTracks.onaddtrack = function (event) {
  if (event.track.kind == 'sign') {
    var sign = document.createElement('video');
    sign.src = url + '#track=' + event.track.id;
    sign.autoplay = true;
    container.appendChild(sign);
    return;
  }
};
}
</script>

```

E.g. "foreign-language dubbed dialogs, director's commentaries, audio descriptions, alternative angles, or sign-language overlays."

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#media-resources-with-multiple-media-tracks>

Audio.track and VideoTrack.kind:

Return values for AudioTrack.kind and VideoTrack.kind		
Category	Definition	Applies to...
"alternative"	A possible alternative to the main track, e.g., a different take of a song (audio), or a different angle (video).	Audio and video.
"captions"	A version of the main video track with captions burnt in. (For legacy content; new content would use text tracks.)	Video only.
"descriptions"	An audio description of a video track.	Audio only.
"main"	The primary audio or video track.	Audio and video.
"main-desc"	The primary audio track, mixed with audio descriptions.	Audio only.
"sign"	A sign-language interpretation of an audio track.	Video only.
"subtitles"	A version of the main video track with subtitles burnt in. (For legacy content; new content would use text tracks.)	Video only.
"translation"	A translated version of the main audio track.	Audio only.
"commentary"	Commentary on the primary audio or video track, e.g., a director's commentary.	Audio and video.
"" (empty string)	No explicit kind, or the kind given by the track's metadata is not recognized by the user agent.	Audio and video.

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#audio-tracklist-and-video-tracklist-objects>

AudioTracks and videoTracks can be declaratively selected.

```
<video src="myvideo#track=alternative"></video>
```

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#selecting-specific-audio-and-video-tracks-declaratively>

Image maps (img, maps, area).

An image map allows geometric areas on an image to be associated with hyperlinks.

```

<map name="shapes">
  <area shape="rect" coords="50,50,100,100" />
  <!-- the hole in the red box -->
  <area shape="rect" coords="25,25,125,125" href="red.html" alt="Red box." />
  <area shape="circle" coords="200,75,50" href="green.html" alt="Green circle." />
  <area shape="poly" coords="325,25,262,125,388,125" href="blue.html" alt="Blue triangle."
/>
  <area shape="poly" coords="450,25,435,60,400,75,435,90,450,125,465,90,500,75,465,60"
href="yellow.html" alt="Yellow star." />
</map>
```

(Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\image-map.xhtml
(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#embedded-content-image-maps>

MathML

MathML is cool ...

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>Math ML</title>

  <link rel="stylesheet" href="style/simple.css" />
</head>
<body>
  <h1>Maths ML</h1>

  <p>Some maths:</p>

  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <mi>x</mi>
    <mo>=</mo>
    <mfrac>
      <mrow>
        <mo form="prefix">-</mo> <mi>b</mi>
        <mo>±</mo>
      </mrow>
      <msqrt>
        <msup> <mi>b</mi> <mn>2</mn> </msup>
        <mo>-</mo>
        <mn>4</mn> <mo></mo> <mi>a</mi> <mo></mo> <mi>c</mi>
      </msqrt>
      </mrow>
      <mrow>
        <mn>2</mn> <mo></mo> <mi>a</mi>
      </mrow>
    </mfrac>
  </math>
</body>
</html>
```

(W3C, 2017. HTML 5.2) <https://www.w3.org/TR/html52/semantics-embedded-content.html#mathml>
(Bentley, n.d.) C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\mathML.xhtml

Polyglot: an explicit namespace declaration is required.

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
```

See [Namespaces](#)

MathML conversion tools:

- <https://webdemo.myscript.com/views/math.html>

Further reading:

- "Authoring MathML" <https://developer.mozilla.org/en-US/docs/Web/MathML/Authoring>

Tabular data

table

Content model: ... optionally a `caption` element, followed by zero or more `colgroup` elements, followed optionally by a `thead` element, followed by either zero or more `tbody` elements or one or more `tr` elements, followed optionally by a `tfoot` element

Table example, full featured.

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>Tables - Full Featured</title>
  <style>
    /*  */

    table {
      border-collapse: collapse;
    }

    th, td {
      border: 1px solid black;
    }

    caption {
      text-align: left;
      padding-bottom: 0.5em;
    }

    tr {
      font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans',
      Arial, sans-serif;
      font-size: 0.9em;
    }

    /* The top row, a heading row */
    thead tr {
      text-align: centre;
      background-color: cornsilk;
    }

    /* The top row, a heading row */
    tfoot tr {
      font-weight: bold;
      background-color: ghostwhite;
      border-top-style: double;
      text-align: center;
    }

    /* Styling columns using classes */
    tbody .col02 {
      text-align: right;
      font-style: italic;
    }

    /* ]]&gt; */
  &lt;/style&gt;
</pre>
</div>
<div data-bbox="114 938 321 956" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div>
<div data-bbox="362 938 663 957" data-label="Page-Footer">John Bentley of <a href="http://www.softmake.com.au">www.softmake.com.au</a></div>
<div data-bbox="702 938 822 957" data-label="Page-Footer">Page 55 of 103</div>
```

```

</head>
<body>

  <h1>Tables - Full Featured</h1>

  <table id="table01">
    <caption>
      A <code>Caption</code> in a Table.
    </caption>
    <colgroup span="2" style="width: 200px; background-color: #def0f8;"></colgroup>
    <colgroup>
      <col style="width: 400px; background-color: #b3f8b3" />
    </colgroup>
    <thead>
      <tr>
        <th>th in thead</th>
        <th class="col02">th in thead</th>
        <th>th in thead</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <th>th in tbody</th>
        <td class="col02">td in tbody</td>
        <td>td in tbody</td>
      </tr>
      <tr>
        <th>th in tbody</th>
        <td class="col02">td in tbody</td>
        <td>td in tbody</td>
      </tr>
    </tbody>
    <tfoot>
      <tr>
        <th>th in tfoot</th>
        <td class="col02">td in tfoot</td>
        <td>td in tfoot</td>
      </tr>
    </tfoot>
  </table>

</body>
</html>

```

(W3C HTML5.3, 2017) <https://www.w3.org/TR/html53/tabular-data.html#the-table-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\tables-full-featured.xhtml

Table example, simple.

```

<body>

  <h1>Tables- Simple</h1>

  <style>
    /*  */

    table {
      border-collapse: collapse;
    }

    th, td {
      border: 1px solid black;
      width: 100px;
    }

    /*  */
  </style>
  <table id="table01">
    <tr>
      <th>th</th>
      <th>th</th>
      <th>th</th>
    </tr>
    <tr>

```



```

    <th>th</th>
    <td>td</td>
    <td>td</td>
  </tr>
  <tr>
    <th>th</th>
    <td>td</td>
    <td>td</td>
  </tr>
</table>

</body>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\tables-simple.xhtml

Further examples from the horse's mouth.

(W3C HTML5.3, 2017), "4.9.13. Examples", <https://www.w3.org/TR/html53/tabular-data.html#examples>

caption

When a table element is the only content in a figure element other than the figcaption, the caption element should be omitted in favor of the figcaption.

(W3C HTML5.3, 2017) <https://www.w3.org/TR/html53/tabular-data.html#the-caption-element>

colgroup, col

The styles that apply to colgroup and col are quite limited: border, background, width, and visibility. In practice you probably only want to set, at most width and background-color.

```

<table id="table01">
  <caption>
    A <code>Caption</code> in a Table.
  </caption>
  <colgroup span="2" style="width: 200px; background-color: #def0f8;"></colgroup>
  <colgroup>
    <col style="width: 400px; background-color: #b3f8b3" />
  </colgroup>
  <thead>
  ...

```

<https://www.w3.org/TR/CSS22/tables.html#columns>

Attributes span (for colgroup, col elements); and rowspan, colspan (for th, td elements)

Attributes span (for colgroup, col elements); and rowspan, colspan (for th, td elements) - work as follows:

- The `span` attribute for a `colgroup` just serves to define how many columns within the `colgroup` are included. This is useful only for styling the `colgroup` and, in turn the children `cols`.
- The `span` attribute for a `col` similarly just serves to define how many columns any styling on the `col` should be applied to.
- The `colspan` attribute for a `th` or `td` defines how many cells in adjacent columns should be merged.
- The `rowspan` attribute for a `th` or `td` defines how many cells in adjacent row should be merged.

```
<style>
/*  */

table {
  border-collapse: collapse;
}
th, td {
  border: 1px solid black;
  width: 100px;
}
/* ]]&gt; */
&lt;/style&gt;
&lt;table id="table01"&gt;
  &lt;colgroup span="2" style="background-color: ghostwhite" /&gt;
  &lt;col span="3" style="background-color: lavenderblush" /&gt;
  &lt;tr&gt;
    &lt;th&gt;th&lt;/th&gt;
    &lt;th&gt;th&lt;/th&gt;
    &lt;th colspan="3"&gt;th&lt;/th&gt;
    &lt;th&gt;th&lt;/th&gt;
  &lt;/tr&gt;
  &lt;tr&gt;
    &lt;th rowspan="2"&gt;th&lt;/th&gt;
    &lt;td&gt;td&lt;/td&gt;
    &lt;td colspan="3"&gt;td&lt;/td&gt;
    &lt;td&gt;td&lt;/td&gt;
  &lt;/tr&gt;
  &lt;tr&gt;
    &lt;td&gt;td&lt;/td&gt;
    &lt;td&gt;td&lt;/td&gt;
    &lt;td&gt;td&lt;/td&gt;
    &lt;td&gt;td&lt;/td&gt;
  &lt;/tr&gt;
&lt;/table&gt;</pre>
</div>
<div data-bbox="152 649 332 665" data-label="Text">
<p>The above code produces:</p>
</div>
<div data-bbox="176 678 554 732" data-label="Table">
<table border="1">
<thead>
<tr>
<th>th</th>
<th>th</th>
<th colspan="3">th</th>
<th>th</th>
</tr>
</thead>
<tbody>
<tr>
<th>th</th>
<td>td</td>
<td colspan="3">td</td>
<td>td</td>
</tr>
<tr>
<td></td>
<td>td</td>
<td>td</td>
<td>td</td>
<td>td</td>
<td>td</td>
</tr>
</tbody>
</table>
</div>
<div data-bbox="125 748 840 775" data-label="Text">
<p>(Bentley, n.d.) 2018-02-13, C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\tables-span-attribute.xhtml</p>
</div>
<div data-bbox="115 791 833 807" data-label="Text">
<p>The <code>rowspan</code> attribute may have a zero value. That means: span the remaining rows.</p>
</div>
<div data-bbox="125 820 856 849" data-label="Text">
<p>(W3C HTML5.3, 2017), "4.9.11. Attributes common to <code>td</code> and <code>th</code> elements", <a href="https://www.w3.org/TR/html53/tabular-data.html#attributes-common-to-td-and-th-elements">https://www.w3.org/TR/html53/tabular-data.html#attributes-common-to-td-and-th-elements</a></p>
</div>
<div data-bbox="115 864 164 880" data-label="Text">
<p><b>tbody</b></p>
</div>
<div data-bbox="115 894 163 910" data-label="Text">
<p><b>thead</b></p>
</div>
<div data-bbox="115 938 321 955" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div>
<div data-bbox="363 938 663 956" data-label="Page-Footer">John Bentley of <a href="http://www.softmake.com.au">www.softmake.com.au</a></div>
<div data-bbox="702 938 822 956" data-label="Page-Footer">Page 58 of 103</div>
```

The `<thead>` element contains a block of rows (generally only one row) who, in turn, contain column headers.

Note `th` is used within `thead`.

```
...
</colgroup>
<thead>
  <tr>
    <th>th in thead</th>
    <th class="col02">th in thead</th>
    <th>th in thead</th>
  </tr>
</thead>
<tbody>
...
```

(W3C HTML5.3, 2017), "4.9.6. The thead element", <https://www.w3.org/TR/html53/tabular-data.html#the-thead-element>

tfoot

tr

The `tr` element can be a child of `thead`, `tbody`, `tfoot`; or directly of `table`.

(W3C HTML5.3, 2017), "4.9.8. The tr element", <https://www.w3.org/TR/html53/tabular-data.html#the-tr-element>

td

th

In `th` there is a `scope` attribute which stipulates the cells the `th` applies to. The `scope` attribute may have the values: `row`, `col`, `rowgroup`, `colgroup`.

```
<table>
  <caption>Measurement of legs and tails in Cats and English speakers</caption>
  <thead>
    <tr> <th> ID <th> Measurement <th> Average <th> Maximum
  </thead>
  <tbody>
    <tr> <td> <th scope=rowgroup> Cats <td> <td>
    <tr> <td> 93 <th scope=row> Legs <td> 3.5 <td> 4
    <tr> <td> 10 <th scope=row> Tails <td> 1 <td> 1
  </tbody>
  <tbody>
    <tr> <td> <th scope=rowgroup> English speakers <td> <td>
    <tr> <td> 32 <th scope=row> Legs <td> 2.67 <td> 4
    <tr> <td> 35 <th scope=row> Tails <td> 0.33 <td> 1
  </tbody>
</table>
```

This would result in the following table:

Table 1 Measurement of legs and tails in Cats and English speakers

ID	Measurement	Average	Maximum
	Cats		
93	Legs	3.5	4
10	Tails	1	1

	English speakers		
32	Legs	2.67	4
35	Tails	0.33	1

(W3C HTML5.3, 2017), "4.9.10. The th element", <https://www.w3.org/TR/html53/tabular-data.html#the-th-element>

Attributes common to td and th elements

See [Attributes span \(for colgroup, col elements\); and rowspan, colspan \(for th, td elements\)](#), above.

The `headers` attribute for `td` and `th` may content a list of ids. The ids must reference those of another `th` in the same table.

(W3C HTML5.3, 2017), "4.9.11. Attributes common to td and th elements", <https://www.w3.org/TR/html53/tabular-data.html#attributes-common-to-td-and-th-elements>

Forms

form

Form Template: create forms using `div` enclosing labels and controls. Buttons are wrapped in a `<p class="buttonGroup">` Example XHTML form ...

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>
    Forms - all controls, div enclosing.
  </title>
  <!-- <link rel="stylesheet" href="style/forms-div-enclosing-traditional.css" /> -->
  <link rel="stylesheet" href="style/forms-div-enclosing-grid.css" />
</head>
<body>
  <h1>Forms - all controls, div enclosing.</h1>

  <form method="post" enctype="application/x-www-form-urlencoded"
action="http://localhost:8080/web/Examples/Html5Examples/forms-process.php">

    <section>
      <h2>Input element of various types</h2>

      <fieldset>
        <legend>Textish</legend>
        <input type="hidden" name="SecretValue" value="Kenny Everett" />
        <div>
          <label>Given name (type=text default)</label>
          <input name="FirstName" />
        </div>
        <div>
          <label>Family name (type=text explicit)</label> <input type="text"
name="FamilyName" />
        </div>
        <div>
          <label>Search (type=search)</label>
          <input type="search" name="Search" />
        </div>
      </fieldset>

      ...
      <fieldset>
        <legend>Pizza Toppings (type=checkbox)</legend>
        <div>

```

```

        <input type="checkbox" name="Toppings[]" value="Bacon" />
        <label>Bacon</label>
    </div>
    <div>
        <input type="checkbox" name="Toppings[]" value="Extra cheese" />
        <label>Extra Cheese</label>
    </div>
    <div>
        <input type="checkbox" name="Toppings[]" value="onion" />
        <label>Onion</label>
    </div>
</fieldset>

...

    <p class="buttonGroup">
        <input type="submit" />
        <input type="reset" />
        <input type="image" name="ImageButton" src="images/beautiful-scene.jpg"
alt="Beautiful scene of mountains" />
    </p>

</fieldset>
</section>

<section>
    <h2>Non-input elements</h2>

    ...

    <div>
        <label>Delivery instructions (textarea)</label>
        <textarea name="deliveryInstructions"></textarea>
    </div>

    <p class="buttonGroup">
        <button type="submit">Submit order</button>
        <button type="reset">Reset</button>
    </p>
</section>
</form>
</body>
</html>

```

Wrapping the label and control in a div permits CSS styling in all sorts of ways. Namely, using grids or using traditional techniques. It also beautifies (be subject to automatic reformatting in an IDE) readily.

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing-the-template.xhtml

In the source code the label and control:

- Can be on different lines; or
- If they are on the same line then space separate them.

```

<div>
    <label for="FirstName" >Given name (type=text default)</label>
    <input name="FirstName" id="FirstName" />
</div>
<div>
    <label>Family name (type=text explicit)</label> <input type="text" name="FamilyName" />
</div>

```

This keeps the rendered output consistent.

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

CSS for forms ...

(Bentley, 2018. *CascadingStyleSheets-Css-Reference.docx*,
<file:///Atlas/Users/John/Documents/Sda/Info/Web/KB/Css/Reference/CascadingStyleSheets-Css-Reference.docx>,
<file:///Atlas/Users/John/Documents/Sda/Info/Web/KB/Css/Reference/CascadingStyleSheets-Css-Reference.docx>,
 GridLayout > Template Examples > For a Form.,

(Discouraged) An alternative form technique is to `label` wrap controls.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
  <head>
    <meta charset="utf-8" />
    <title>
      Forms - all controls, label enclosing
    </title>
    <meta name='viewport' content='width=device-width, initial-scale=1.0' />
    <link rel="stylesheet" href="style/forms-label-enclosing.css" />
  </head>
  <body>
    <h1>Forms - all controls, label enclosing</h1>

    <form method="post" enctype="application/x-www-form-urlencoded" action=
      "http://localhost:8080/web/Examples/Html5Examples/forms-process.php">

      <section>
        <h2>Input element of various types</h2>

        <fieldset>
          <legend>Textish</legend>
          <input type="hidden" name="SecretValue" value="Kenny Everett" />
          <!-- Each element on own line -->
          <label>
            <span>Given name (type=text default):</span>
            <input name="FirstName" />
          </label>
          <!-- For the styling to be consistent with "Each element own own line" put a space
between span and input -->
          <label><span>Family name (type=text explicit):</span> <input type="text"
name="FamilyName" /></label>
          <label><span>Search (type=search):</span> <input type="search" name="Search"
/></label>
        </fieldset>

        ...

        <fieldset>
          <legend>Clickables</legend>
          <label><span>What colour would you like your pizza (type=color)?:</span> <input
type="color" name="WhatColor" /></label>
          <label><span>Photo for your pizza box:</span> <input type="file"
name="PhotoForBox" value="Upload photo ..." /></label>

          <input type="submit" />
          <input type="reset" />
          <input type="image" name="ImageButton" src="images/beautiful-scene.jpg" />
        </fieldset>
      </section>

      <section>
        <h2>Non-input elements</h2>

        <label>
          <span>Dessert Size (select):</span>
          <select name="DessertSize">
            <!-- Returns "Small" to the server -->
            <option>Small</option>
            <!-- Returns "Medium Dessert Size" to the server -->
            <option value="Medium dessert size">Medium</option>
            <!-- Superfluous value attribute -->
            <option value="Large">Large</option>
          </select>
        </label>

        ...
```

```

        <button type="submit">Submit order</button>
        <button type="reset">Reset</button>
    </section>
</form>
</body>
</html>

```

This is discouraged because the CSS styling options are relatively limited.

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-label-enclosing.xhtml

label

If you want to explicitly associate the `label` with the control you can use the a label's `for` attribute and the control's `id` attribute.

```

<div>
  <label for="FirstName" >Given name (type=text default)</label>
  <input name="FirstName" id="FirstName" />
</div>

```

(W3C HTML5.3, 2017), "The label element", <https://www.w3.org/TR/html53/sec-forms.html#the-label-element>

input

Input types overview

Input Type	
Hidden state (type=hidden)	
Text (type=text) state and Search state (type=search)	
Telephone state (type=tel)	
URL state (type=url)	
E-mail state (type=email)	
Password state (type=password)	
Date state (type=date)	
Time state (type=time)	
Number state (type=number)	
Range state (type=range)	
Color state (type=color)	
Checkbox state (type=checkbox)	
Radio Button state (type=radio)	
File Upload state (type=file)	
Submit Button state (type=submit)	
Image Button state (type=image)	
Reset Button state (type=reset)	

Button state (type=button)	
----------------------------	--

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html5/Overview.html#contents>

Kinds of types

Textish types

```
<input type="hidden" name="SecretValue" value="Kenny Everett" />
<div>
  <label for="FirstName" >Given name (type=text default)</label>
  <input name="FirstName" id="FirstName" />
</div>
<div>
  <label>Family name (type=text explicit)</label> <input type="text" name="FamilyName" />
</div>
<div>
  <label>Search (type=search)</label>
  <input type="search" name="Search" />
</div>
```

(W3C HTML5.3, 2017), "States of the type attribute", <https://www.w3.org/TR/html53/sec-forms.html#sec-states-of-the-type-attribute>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

Special types

```
<div>
  <label>Mobile (type=tel)</label>
  <input type="tel" name="Mobile" id="Mobile" />
</div>
<div>
  <label>Favourite Website (type=url)</label>
  <input type="url" name="FavouriteWebsite" />
</div>
<div>
  <label>Email (type=email)</label>
  <input type="email" name="Email" />
</div>
<div>
  <label>Password (type=password)</label>
  <input type="password" name="Password" />
</div>
```

(W3C HTML5.3, 2017), "States of the type attribute", <https://www.w3.org/TR/html53/sec-forms.html#sec-states-of-the-type-attribute>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

datetime types

```
<div>
  <label>Preferred delivery date (type=date)</label>
  <input type="date" name="PrefferredDeliveryDate" />
</div>
<div>
  <label>Your month (type=month)</label>
  <input type="month" name="YourMonth" />
</div>
<div>
  <label>Your week (type=week)</label>
  <input type="week" name="YourWeek" />
</div>
<div>
  <label>Preferred delivery time (type=time)</label>
  <input type="time" name="PreferredDeliveryTime" min="11:00" max="21:00" step="900" />
</div>
```



```
<div>
  <label>Your local date and time (type=datetime-local)</label>
  <input type="datetime-local" name="YourDateAndTimeLocal" />
</div>
```

(W3C HTML5.3, 2017), "States of the type attribute", <https://www.w3.org/TR/html53/sec-forms.html#sec-states-of-the-type-attribute>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

Number types

```
<div>
  <label>Scoops of Gelato (type=number)?</label>
  <input type="number" name="Scoops" min="1" max="3" /> </div>
<div>
  <label>How many free items (type=range)?</label>
  <input type="range" name="FreeItems" min="0" max="10" step="2" value="2" />
</div>
```

(W3C HTML5.3, 2017), "States of the type attribute", <https://www.w3.org/TR/html53/sec-forms.html#sec-states-of-the-type-attribute>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

clickable types

```
<div>
  <label>What colour would you like your pizza (type=color)?</label>
  <input type="color" name="WhatColor" />
</div>
<div>
  <label>Photo for your pizza box</label>
  <input type="file" name="PhotoForBox" />
</div>

<p class="buttonGroup">
  <input type="submit" />
  <input type="reset" />
  <input type="image" name="ImageButton" src="images/beautiful-scene.jpg" alt="Beautiful scene of mountains" />
</p>
```

(W3C HTML5.3, 2017), "States of the type attribute", <https://www.w3.org/TR/html53/sec-forms.html#sec-states-of-the-type-attribute>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

Particular type detail

type=hidden

```
<input type="hidden" name="SecretValue" value="Kenny Everett" />
```

(W3C HTML5.3, 2017), "States of the type attribute", <https://www.w3.org/TR/html53/sec-forms.html#sec-states-of-the-type-attribute>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

type=text

If the type attribute missing then type=text is the default.

```
<div>
  <label for="FirstName" >Given name (type=text default)</label>
  <input name="FirstName" id="FirstName" />
</div>
<div>
  <label>Family name (type=text explicit)</label> <input type="text" name="FamilyName" />
</div>
```

```
</div>
```

(W3C HTML5.3, 2017), "States of the type attribute", <https://www.w3.org/TR/html53/sec-forms.html#sec-states-of-the-type-attribute>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

type=checkbox

Handle checkboxes by appending "[]" after the name so that they are passed along as an array.

Example Markup:

```
<form method="post"
  enctype="application/x-www-form-urlencoded"
  action="forms-process.php">
  <label>Customer name: <input name="customerName" /></label>
  <fieldset>
    <legend> Pizza Toppings </legend>
    <label> <input type="checkbox" name="toppings[]" value="bacon" /> Bacon </label>
    <label> <input type="checkbox" name="toppings[]" value="cheese" /> Extra Cheese </label>
    <label> <input type="checkbox" name="toppings[]" value="onion" /> Onion </label>
  </fieldset>
  <label><button>Submit order</button></label>
</form>
```

Example Php:

```
<?php
/*
Summary: Output variables pushed to the page. Handle arrays that have been sent.
Remarks: $_REQUEST handles posts or gets.
*/
echo '<pre>';
foreach($_REQUEST as $name => $value){
  if (is_array($value)) {
    echo "$name:<br />";

    // Assign array to something more mnemonic
    $items = $value;
    foreach ($items as $item) {
      echo "  $item<br />";
    }
  } else {
    echo "$name: $value<br />";
  }
}
echo '</pre>';
?>
```

Example Output:

```
customerName: John
toppings:
  bacon
  cheese
```

(Bentley, n.d.), 2018-02-19, C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-w3c-pizza-example.xhtml

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-process.php

(Stack Exchange Inc, 2015) <https://stackoverflow.com/a/48864706/872154>

(W3C HTML5.3, 2017) <https://www.w3.org/TR/html53/sec-forms.html#sec-forms>

Default checkedness state: add the checked attribute for checked; remove the checked attribute for unchecked.

```
<div>
  <!-- Unchecked by default -->
  <input type="checkbox" name="debug" />
  <label>Debug</label>
</div>
```

```
<div>
  <!-- Checked by default -->
  <input type="checkbox" name="developerNotes" checked="checked" />
  <label>Developer Notes</label>
</div>
```

The checked content attribute is a boolean attribute that gives the default checkedness of the input element. When the checked content attribute is added, if the control does not have dirty checkedness, the user agent must set the checkedness of the element to true; when the checked content attribute is removed, if the control does not have dirty checkedness, the user agent must set the checkedness of the element to false.

(Web Hypertext Application Technology Working Group (WHATWG) The latest, "HTML the Living Standard", <https://html.spec.whatwg.org/multipage/> <https://html.spec.whatwg.org/multipage/input.html#attr-input-checked>

type=radio

```
<div>
  <input type="radio" name="Size" value="small" />
  <label>Small</label>
</div>
<div>
  <input type="radio" name="Size" value="medium" />
  <label>Medium</label>
</div>
<div>
  <input type="radio" name="Size" value="large" />
  <label>Large</label>
</div>
```

(W3C HTML5.3, 2017), "States of the type attribute", <https://www.w3.org/TR/html53/sec-forms.html#sec-states-of-the-type-attribute>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

type=image

The element is a button, specifically a Submit Button.

```
<input type="image" name="ImageButton" src="images/beautiful-scene.jpg" />
```

(W3C The Latest, "HTML 5.3, the Latest", <https://www.w3.org/TR/html53/>), "4.10.5.1.19. Image Button state (type="image")", <https://www.w3.org/TR/html53/sec-forms.html#image-button-state-typeimage>

This button can be used to:

- Pass x and y coordinates of the image to the server, in addition to signalling a submit; or

```
ImageButton_x: 136
ImageButton_y: 105
```

- Just signal a submit.

<file:///C:/Users/John/Documents/Sda/Code/web/Examples/Html5Examples/forms-w3c-pizza-example.xhtml>

non-input controls

button

```
<p class="buttonGroup">
  <button type="submit">Submit order</button>
  <button type="reset">Reset</button>
  <button type="button">Do something</button>
```

```
</p>
```

(W3C HTML5.3, 2017), "The button element", <https://www.w3.org/TR/html53/sec-forms.html#the-button-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

select

```
<div>
  <label>Dessert Size (select)</label>
  <select name="DessertSize">
    <!-- Returns "Small" to the server -->
    <option>Small</option>
    <!-- Returns "Medium Dessert Size" to the server -->
    <option value="Medium dessert size">Medium</option>
    <!-- Superflous value attribute -->
    <option value="Large">Large</option>
  </select>
</div>
```

(W3C HTML5.3, 2017), "The select element", <https://www.w3.org/TR/html53/sec-forms.html#the-select-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

Handle multiple values by using the multiple attribute and naming the control with a "[]" suffix.

```
<div>
  <label>Colors (select multiple)</label>
  <select name="Colors[]" multiple="multiple">
    <option>Black</option>
    <option>White</option>
    <option>Yellow</option>
    <option>Green</option>
    <option>Blue</option>
  </select>
</div>
```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-process.php

datalist

```
<div>
  <label>Chili strength (list attribue and datalist element)</label>
  <input name="sex" list="chili-strength" />
  <datalist id="chili-strength">
    <option>Mild</option>
    <option>Medium</option>
    <option>Hot</option>
  </datalist>
</div>
```

(W3C HTML5.3, 2017), "The datalist element", <https://www.w3.org/TR/html53/sec-forms.html#the-datalist-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

optgroup

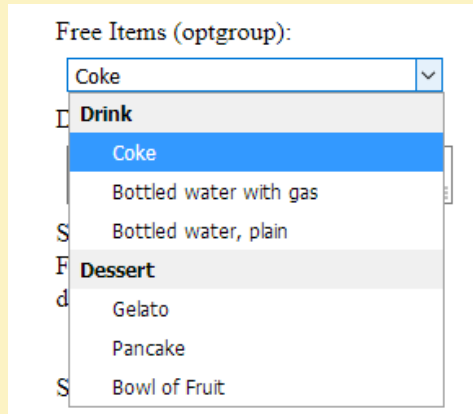
The optgroup is just a way of heading some of the options in a select.

```
<div>
  <label>Free Items (optgroup)</label>
  <select name="free-item">
    <optgroup label="Drink">
      <option value="Lolly water">Coke</option>
      <option value="Aqua minerale carbonated">Bottled water with gas</option>
      <option value="Aqua minerale plain">Bottled water, plain</option>
    </optgroup>
  </select>
</div>
```

```

<optgroup label="Dessert">
  <option value="Gelato">Gelato</option>
  <option value="Pancake">Pancake</option>
  <option value="Bowl of fruit">Bowl of Fruit</option>
</optgroup>
</select>
</div>

```



"The optgroup element", <https://www.w3.org/TR/html53/sec-forms.html#the-optgroup-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

option

The option element is used in select, or optiongroup, as above.

textarea

```

<div>
  <label>Delivery instructions (textarea)</label>
  <textarea name="deliveryInstructions"></textarea>
</div>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml
 (W3C HTML5.3, 2017), "The textarea element", <https://www.w3.org/TR/html53/sec-forms.html#the-textarea-element>

output

the [output] element's value itself is not submitted when the form is submitted.

```

<div>
  <label>Script target (output)</label>
  <output name="SomeOutput">To override</output>
</div>

```

(W3C HTML5.3, 2017), " 4.10.12. The output element", <https://www.w3.org/TR/html53/sec-forms.html#the-output-element>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

progress

Progress bars are determinate or indeterminate.

```

<div>
  <label>Form complete (progress, determinate)</label>
  <progress max="100" value="75" >Determinate ..</progress>
</div>

```

```
<div>
  <label>Sending ... (progress, indeterminate)</label>
  <progress id="indeterminateProgress" >Indeterminate ...</progress>
</div>
```

(W3C HTML5.3, 2017), "The progress element", <https://www.w3.org/TR/html53/sec-forms.html#the-progress-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

meter

```
<div>
  <label>Fun factor (meter)</label>
  <meter min="0" max="10" optimum="8" value="5">Meter ...</meter>
</div>
```

(W3C HTML5.3, 2017), "The meter element", <https://www.w3.org/TR/html53/sec-forms.html#the-meter-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

fieldset

```
<fieldset>
  <legend>Textish</legend>
  <input type="hidden" name="SecretValue" value="Kenny Everett" />
  <div>
    <label for="FirstName" >Given name (type=text default)</label>
    <input name="FirstName" id="FirstName" />
  </div>
  <div>
    <label>Family name (type=text explicit)</label> <input type="text" name="FamilyName" />
  </div>
  <div>
    <label>Search (type=search)</label>
    <input type="search" name="Search" />
  </div>
</fieldset>
```

</fieldset> (W3C HTML5.3, 2017), "The feildset element", <https://www.w3.org/TR/html53/sec-forms.html#the-fieldset-element>
 C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

legend

The legend element can be the first child of a fieldset, as above.

Form child element attributes

name attribute

See (Bentley, 2016. *Web Code Convention - Naming and Style - General*,
 file://Atlas/Users/John/Documents/Sda/Info/Web/System/Conventions/Web Code Convention - Naming and Style -
 General.docx) <file://Atlas/Users/John/Documents/Sda/Info/Web/System/Conventions/Web Code Convention - Naming and Style - General.docx>, "Markup (xml or xhtml) form field names"

Input attributes

`minlength` and `maxlength` restrict the characters that can be input into the element.

```
<div>
  <label>minlength and maxlength</label>
  <input name="MinLengthAndMaxLength" minlength="5" maxlength="10" />
</div>
```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml (W3C HTML5.3, 2017), "The maxlength and minlength attributes", <https://www.w3.org/TR/html53/sec-forms.html#the-maxlength-and-minlength-attributes>

`size` sets the visual size of the input control, unless css styling overrides this.

```
<input name="Size" id="Size" size="50" />
```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml (W3C HTML5.3, 2017), "The size attribute", <https://www.w3.org/TR/html53/sec-forms.html#the-size-attribute>

`multiple` is a boolean indicating whether the user can set more than one value. This also works with a select element. This doesn't work on an input of `type="text"`.

```
<div>
  <label>multiple input (list and datalist)</label>
  <input type="email" name="MultipleInputWithList" list="EmailsToSend" multiple="multiple" />
  <datalist id="EmailsToSend">
    <option>hedral@example.cat</option>
    <option>pillar@example.com</option>
    <option>astrophy@cute.example</option>
    <option>astronomy@science.example.org</option>
  </datalist>
</div>
```

(W3C HTML5.3, 2017), "The multiple Attribute", <https://www.w3.org/TR/html53/sec-forms.html#the-multiple-attribute>
C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

`pattern` is a javascript regex pattern, matching the start and end of the string.

```
<div>
  <label>Rego</label>
  <input name="Rego" pattern="[A-Z]{2}\d{2}[A-Z]{2}"
        title="Two capital letters, two digits, two capital letters"
        placeholder="Two capital letters, two digits, two capital letters" />
</div>

// Matches
AA44EE

// Doesn't match
AA44EEE
```

(W3C HTML5.3, 2017), "The pattern attribute", <https://www.w3.org/TR/html53/sec-forms.html#the-pattern-attribute>
C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

`min` and `max` restrict the range of number or datetime input controls.

```
<fieldset>
  <legend>min and max</legend>
  <div>
    <label>Max date</label>
    <input name="bday" type="date" max="1979-12-31" />
  </div>
  <div>
    <label>Min number. Integer greater than zero</label>
    <input name="quantity" required="required" type="number" min="1" value="1" />
  </div>
  <div>
    <label>Ranged time. Hours between 21:00 and 06:00, default midnight</label>
    <input name="sleepStart" type="time" min="21:00" max="06:00" step="60" value="00:00" />
  </div>
</fieldset>
```

(W3C HTML5.3, 2017), "The min and max attributes", <https://www.w3.org/TR/html53/sec-forms.html#the-min-and-max-attributes>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

`step` controls the granularity of the value (or values).

```
<input name="sleepStart" type="time" min="21:00" max="06:00" step="60" value="00:00" />

<!-- values in the range 0..1, and allows 256 steps in that range -->
<input name="opacity" type="range" min="0" max="1" step="0.00392156863" />

<!-- Normally, time controls are limited to an accuracy of one minute. The following control
allows any time in the day to be selected, with any accuracy (e.g., thousandth-of-a-second
accuracy or more) -->
<input name="favtime" type="time" step="any">
```

(W3C HTML5.3, 2017), "The step attribute", <https://www.w3.org/TR/html53/sec-forms.html#the-step-attribute>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

`list` identifies the ID of a `datalist` element holding options. The options in the `datalist` become, in effect, suggestions.

```
<div>
  <label>multiple input</label>
  <input name="MultipleInputWithList" list="PlanetsToVisit" multiple="multiple" />
  <datalist id="PlanetsToVisit">
    <option>Venus</option>
    <option>Earth</option>
    <option>Mars</option>
    <option>Saturn</option>
    <option>Jupiter</option>
  </datalist>
</div>

<label>
  Homepage:
  <input name="hp" type="url" list="hpurls">
</label>
<datalist id="hpurls">
  <option value="https://www.example.com/" label="Example">
  <option value="https://w3.org/" label="W3C">
</datalist>
```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

(W3C HTML5.3, 2017), "The list attribute", <https://www.w3.org/TR/html53/sec-forms.html#the-list-attribute>

`placeholder` represents a short hint to the user. Don't use as a substitute for a label.

```
<div>
  <label>State</label>
  <input type="text" name="State" placeholder="NSW or VIC etc." />
</div>
```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

(W3C HTML5.3, 2017), "The placeholder attribute", <https://www.w3.org/TR/html53/sec-forms.html#the-placeholder-attribute>

`autocapitalize` is an enumerated attribute with the following values and consequences (doesn't seem to be supported at 2018-05-24):

- `characters`. "SIMPLE EXAMPLE. FOUR WORDS.";
- `words`. "Simple Example. Four Words.";
- `sentences`. "Simple example. Four words."
- (Any other value). (As entered by user).

```
<div>
  <label>Given Name (autocapitalize="words")</label>
  <input type="text" name="GivenName" autocapitalize="words" />
</div>
```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml
(W3C HTML5.3, 2017), "The autocapitalize attribute", <https://www.w3.org/TR/html53/sec-forms.html#the-autocapitalize-attribute>

capture "requests a specific microphone or camera for the media capture mechanism, for example the user-facing and outward facing cameras on smartphones. It has one of the two values `user` or `environment`."

(W3C HTML5.3, 2017), "The capture attribute", <https://www.w3.org/TR/html53/sec-forms.html#the-capture-attribute>

Other input control attributes: `readonly`, `required`.

Other attributes common to many controls

`minlength` and `maxlength` are common to all controls. (See above for details).

`disabled` is common to many controls.

(W3C HTML5.3, 2017), "Enabling and disabling form controls: the disabled attribute", <https://www.w3.org/TR/html53/sec-forms.html#enabling-and-disabling-form-controls-the-disabled-attribute>

`autofocus` on a control specifies it should receive autofocus on page load. Not recommended as using this fucks with accessibility.

```
<div>
  <label>Given Name (autocapitalize="words")</label>
  <input type="text" name="GivenName" autocapitalize="words" autofocus="autofocus" />
</div>
```

(W3C HTML5.3, 2017), "Autofocusing a form control: the autofocus attribute", <https://www.w3.org/TR/html53/sec-forms.html#autofocusing-a-form-control-the-autofocus-attribute>

Form submission attributes

Form v submit buttons

Form submission attributes can occur either on the `form` element or on submit buttons (either an `input` or `button` element with `type="submit"`). When on a submit button the attribute is prefixed with "form" (e.g. the form attribute "action" is named "formaction" on a submit button).

```
<form method="post" action="http://localhost:8080/web/Examples/Html5Examples/forms-process.php" enctype="application/x-www-form-urlencoded" novalidate="novalidate">
  <!-- or -->
  <p class="buttonGroup">
```

```
<button type="submit" formmethod="post"
formaction="http://localhost:8080/web/Examples/Html5Examples/forms-process.php"
formenctype="application/x-www-form-urlencoded" formnovalidate="formnovalidate">Submit
order</button>
</p>
```

(W3C HTML5.3, 2017), "Form submission", <https://www.w3.org/TR/html53/sec-forms.html#form-control-infrastructure-form-submission>

action (and formaction)

`action` (and `formaction`) specifies a URL.

```
<form method="get" action="/search">
<form method="post" action="http://localhost:8080/web/Examples/Html5Examples/forms-
process.php">
<button type="submit" formaction="http://localhost:8080/web/Examples/Html5Examples/forms-
process.php">Submit order</button>
```

(W3C HTML5.3, 2017), "Form submission", <https://www.w3.org/TR/html53/sec-forms.html#form-control-infrastructure-form-submission>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

Form submission attributes: send via email example.

```
<form method="post" enctype="text/plain" action="mailto:jrlbentley@yahoo.com.au">
```

Form submission attributes: send via script example.

```
<form method="post"
action="http://www.site.org/cgi-bin/display.pl">
```

method (and formmethod)

`method` (and `formmethod`) specifies one of: `get`; `post`; `dialog`

```
<form method="get" action="/search">
// Dialog
<dialog id="ship">
  <form method="dialog">
    <p>A ship has arrived in the harbour.</p>
    <button type="submit" value="board">Board the ship</button>
    <button type="submit" value="call">Call to the captain</button>
  </form>
</dialog>
<script>
  var ship = document.getElementById('ship');
  ship.showModal();
  ship.onclose = function(event) {
    if (ship.returnValue == 'board') {
      // ...
    } else {
      // ...
    }
  };
</script>
```

(W3C HTML5.3, 2017), "Form submission", <https://www.w3.org/TR/html53/sec-forms.html#form-control-infrastructure-form-submission>

If all else is equal use `post` rather than `get` as `get` limits the amount of data you can send.

(Castro, 2003, p.269)

enctype (and formenctype)

`enctype` (and `formenctype`) specifies one of:

- `application/x-www-form-urlencoded`;
- `multipart/form-data`;
- `text/plain`.

```
<form method="post" enctype="application/x-www-form-urlencoded"
action="http://localhost:8080/web/Examples/Html5Examples/forms-process.php">
```

(W3C HTML5.3, 2017), "Form submission", <https://www.w3.org/TR/html53/sec-forms.html#form-control-infrastructure-form-submission>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing.xhtml

novalidated (and formnovalidate)

`novalidated` (and `formnovalidate`) indicate the form is not to be submitted during submission. Useful for save or cancel operations.

```
<form action="editor" method="post">
  <div>
    <label for="fname">Name:</label>
    <input id="fname" name="fn" required>
  </div>
  <div>
    <label for="essay">Essay:</label>
    <textarea id="essay" name="essay" required></textarea>
  </div>
  <div>
    <input type="submit" name="submit" value="Submit essay">
    <input type="submit" formnovalidate="formnovalidate" name="save" value="Save essay">
    <input type="submit" formnovalidate="formnovalidate" name="cancel" value="Cancel">
  </div>
</form>
```

(W3C HTML5.3, 2017), "Form submission", <https://www.w3.org/TR/html53/sec-forms.html#form-control-infrastructure-form-submission>

target

`target` exists.

Form autofilling with the `autocomplete` attribute.

Some user agents (e.g. some browsers like Chrome) provide help with autofilling forms. This is facilitated with the `autocomplete` attribute.

(W3C HTML5.3, 2017), "Autofill", <https://www.w3.org/TR/html53/sec-forms.html#sec-autofill>

The `autocomplete` attribute allows up to three kinds of tokens: a section identifier; `[shipping|billing]`; and field type.

```
<fieldset>
  <legend>Ship the blue gift to...</legend>
  <div>
    <label>
```

```

    Address:
    <input name="ba" autocomplete="section-blue shipping street-address">
  </label>
</div>
...
<fieldset>
  <legend>Ship the red gift to...</legend>
  <div>
    <label>
      Address:
      <input name="ra" autocomplete="section-red shipping street-address">
    </label>
  </div>
  <div>
    <label>
      City:
      <input name="rc" autocomplete="section-red shipping address-level2">
    </label>
  </div>

```

(W3C HTML5.3, 2017), "Autofill", <https://www.w3.org/TR/html53/sec-forms.html#sec-autofill>

The meanings of the `autocomplete` field types are given in (W3C HTML5.3, 2017), "Autofill", <https://www.w3.org/TR/html53/sec-forms.html#sec-autofill>.

In particular there are the following meanings for tricky address fields

autocomplete field type	Meaning	Example
"street-address"	Street address (multiple lines, newlines preserved)	32 Vassar Street MIT Room 32-G524
"address-line1"	Street address (one line per field)	32 Vassar Street
"address-line2"		MIT Room 32-G524
"address-line3"		
"address-level4"	The most fine-grained administrative level, in addresses with four administrative levels	
"address-level3"	The third administrative level, in addresses with three or more administrative levels	
"address-level2"	The second administrative level, in addresses with two or more administrative levels; in the countries with two administrative levels, this would typically be the city, town, village, or other locality within which the relevant street address is found	Cambridge, Mosman
"address-level1"	The broadest administrative level in the address, i.e., the province within which the locality is found; for example, in the US, this would be the state; in Switzerland it would be the canton; in the UK, the post town	MA, NSW

"country"	Country code. Valid ISO 3166-1-alpha-2 country code [ISO3166]	US , AU
"country-name"	Country name	United States, Australia
"postal-code"	Postal code, post code, ZIP code, CEDEX code (if CEDEX, append "CEDEX", and the dissement, if relevant, to the address-level2 field)	02139

(W3C HTML5.3, 2017), "Autofill", <https://www.w3.org/TR/html53/sec-forms.html#sec-autofill>

Autofiling with `autocomplete` example (play with in chrome).

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>
    Forms - Autocomplete
  </title>
  <link rel="stylesheet" href="style/forms-div-enclosing-grid.css" />
</head>
<body>
  <h1>Forms - Autocomplete</h1>

  <form method="post" action="http://localhost:8080/web/Examples/Html5Examples/forms-
process.php" enctype="application/x-www-form-urlencoded">

  <section>
    <h2>Input element of various types</h2>

    <fieldset>
      <legend>Name</legend>
      <div>
        <label for="GivenName">Given name</label>
        <input name="GivenName" autocomplete="given-name"/>
      </div>
      <div>
        <label>Family name</label>
        <input type="text" name="FamilyName" autocomplete="family-name" />
      </div>
      <div>
        <label>Email</label>
        <input type="email" name="Email" autocomplete="email" />
      </div>
      <div>
        <label>Mobile</label>
        <input type="tel" name="Mobile" id="Mobile" autocomplete="mobile tel" />
      </div>
    </fieldset>

    <fieldset>
      <legend>Shipping Address</legend>
      <div>
        <label>AddressLine1</label>
        <input name="ShippingAddressLine1" autocomplete="shipping address-line1" />
      </div>
      <div>
        <label>AddressLine2</label>
        <input name="ShippingAddressLine2" autocomplete="shipping address-line2"/>
      </div>
      <div>
        <label>AddressLine3</label>
        <input name="ShippingAddressLine3" autocomplete="shipping address-line3" />
      </div>
      <div>
        <label>City/Suburb</label>
        <input name="ShippingCitySuburb" autocomplete="shipping address-level2" />
      </div>
      <div>
        <label>State</label>
```

```

        <input name="ShippingState" autocomplete="shipping address-level1" />
    </div>
    <div>
        <label>Postcode</label>
        <input name="ShippingPostCode" autocomplete="shipping postal-code"/>
    </div>
</fieldset>

<fieldset>
    <legend>Billing Address</legend>
    <div>
        <label>AddressLine1</label>
        <input name="BillingAddressLine1" autocomplete="billing address-line1" />
    </div>
    <div>
        <label>AddressLine2</label>
        <input name="BillingAddressLine2" autocomplete="billing address-line2"/>
    </div>
    <div>
        <label>AddressLine3</label>
        <input name="BillingAddressLine3" autocomplete="billing address-line3" />
    </div>
    <div>
        <label>City/Suburb</label>
        <input name="BillingCitySuburb" autocomplete="billing address-level2" />
    </div>
    <div>
        <label>State</label>
        <input name="BillingState" autocomplete="billing address-level1" />
    </div>
    <div>
        <label>Postcode</label>
        <input name="BillingPostCode" autocomplete="billing postal-code"/>
    </div>
</fieldset>

<p class="buttonGroup">
    <button type="submit">Submit order</button>
    <button type="reset">Reset</button>
</p>
</section>
</form>
</body>
</html>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-autocomplete.xhtml
(W3C HTML5.3, 2017), "Autofill", <https://www.w3.org/TR/html53/sec-forms.html#sec-autofill>
C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-autocomplete.xhtml

Forms processing with PHP

Forms processing with PHP.

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
    <meta charset="utf-8" />
    <title>Forms Process</title>
    <link rel="stylesheet" href="style/html5-basic.css" />
</head>
<body>
    <h1>Forms Process</h1>
    <?php
    /*
    Summary: Output variables pushed to the page. Handle arrays that have been sent.
    Remarks: $_REQUEST handles posts or gets.
    Example Markup:
        <form method="post"
            enctype="application/x-www-form-urlencoded"
            action="forms-process.php">
            <label>Customer name: <input name="customerName" /></label>
            <label>Mobile: <input type="tel" name="mobile" /></label>
            <fieldset>

```

```

        <legend> Pizza Toppings </legend>
        <label> <input type="checkbox" name="toppings[]" value="bacon" /> Bacon </label>
        <label> <input type="checkbox" name="toppings[]" value="cheese" /> Extra Cheese
</label>
        <label> <input type="checkbox" name="toppings[]" value="onion" /> Onion </label>
        <div>
            <label>Colors (select multiple)</label>
            <select name="Colors[]" multiple="multiple">
                <option>Black</option>
                <option>White</option>
                <option>Yellow</option>
                <option>Green</option>
                <option>Blue</option>
            </select>
        </div>
    </fieldset>
</form>
*/
echo '<pre>';
foreach($_REQUEST as $name => $value){
    if (is_array($value)) {
        echo "$name:<br />";

        // Assign array to something more mnemonic
        $items = $value;
        foreach ($items as $item) {
            echo " $item<br />";
        }
    } else {
        echo "$name: $value<br />";
    }
}
echo '</pre>';
?>
</body>
</html>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-process.php

Interactive elements

details and summary

The `details` element represents a unit which, by default is initially hidden from the user. If the user wants further information or to reveal controls they can click on the element. The `summary` element is a child of details. The `summary` element: expresses a summary of the hidden stuff; and may be clicked to reveal the stuff.

```

<style>
    /*  */
    details {
        display: inline-block;
        outline: 1px solid green;
    }
    summary {
        cursor: pointer; /* i.e. hand */
    }
    summary:hover {
        outline: 1px solid red;
    }
    /*  */
</style>
</head>
<body>
<h1>Details and Summary Elements</h1>
<!-- <details open="open"> -->
<details>

```

```

<summary>Latin has a long history ...</summary>

<p>Lorem ipsum dolor sit amet.</p>

<p>Beatae recusandae distinctio ad labore.</p>

<p>Enim itaque magnam culpa aperiam.</p>
</details>
</body>
</html>

```

(W3C HTML5.3, 2017), "Interactive elements", <https://www.w3.org/TR/html53/interactive-elements.html#interactive-elements>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\details-summary-elements.xhtml

The `details` element can, by default, reveal all the content it contains through the `open` boolean attribute. Content is revealed either if `open` is set to "open" or a zero length string "".

```

// Reveals content by default
<details open="open">

// Reveals content by default
<details open="">

// Hidden content by default
<details>

```

(W3C HTML5.3, 2017), "Interactive elements", <https://www.w3.org/TR/html53/interactive-elements.html#interactive-elements>

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\details-summary-elements.xhtml

Commands

A command is an abstraction of a link or button. Commands have the following facets:

- Label.
- Access Key.
- Hidden State.
- Disabled State.
- Action. A scripted event handler; a url to navigate to; or a form submission.

(W3C HTML5.3, 2017), "Commands", <https://www.w3.org/TR/html53/interactive-elements.html#commands>

dialog

The `dialog` element does what the name suggests ...

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>Dialog</title>

  <link rel="stylesheet" href="style/html5-basic.css" />
  <script>
    /*  */
    function showDialog () {
      var dialogElement = document.getElementById("myDialog");
      dialogElement.show();
    }
  &lt;/script&gt;
</pre>
</div>
<div data-bbox="114 938 321 955" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div>
<div data-bbox="363 938 663 956" data-label="Page-Footer">John Bentley of <a href="http://www.softmake.com.au">www.softmake.com.au</a></div>
<div data-bbox="703 938 822 956" data-label="Page-Footer">Page 80 of 103</div>
```



```

function closeDialog () {
    var dialogElement = document.getElementById("myDialog");
    dialogElement.close();
}
/* ]]> */
</script>
</head>
<body>
<h1>Dialog</h1>

<p>Lorem ipsum dolor sit amet.</p>

<button type="button" onclick="showDialog();">Show Dialog</button>

<dialog id="myDialog">
  <h2>Add to Wallet</h2>
  <div>
    <label for="MarlinSaves">How many marlin have you saved from being sport
fished?</label>
    <input type="number" name="MarlinSaves" id="MarlinSaves" min="0" />
  </div>
  <div>
    <input type="checkbox" name="Reward" id="Reward" value="onion" />
    <label for="Reward">Do you want a reward?</label>
  </div>
  <p class="buttonGroup">
    <button type="button" onclick="closeDialog();">Close</button>
  </p>
</dialog>
</body>
</html>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\dialog.xhtml
(W3C HTML5.3, 2017), "The dialog element", <https://www.w3.org/TR/html53/interactive-elements.html#the-dialog-element>

At the time of writing it is not widely supported. See <https://caniuse.com/#feat=dialog>

Scripting

script

The `script` element can be placed in `head` or `body`. It can contain script as child content or reference an external file through the `src` attribute.

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>Scripting - script</title>

  <link rel="stylesheet" href="style/html5-basic.css" />

  <script src="script/hello-world.js" />

  <script type="text/ecmascript">
    /*  */
    addLoadEvent(alert("hi"));
    /* ]]]&gt; */
  &lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
  &lt;h1&gt;Scripting - script&lt;/h1&gt;

  &lt;p&gt;Lorem&lt;/p&gt;
  &lt;script async="async"&gt;
    /* <![CDATA[ */
    addLoadEvent(alert("there"));
    /* ]]]&gt; */
  &lt;/script&gt;
</pre>
</div>
<div data-bbox="114 938 320 955" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div>
<div data-bbox="363 938 663 956" data-label="Page-Footer">John Bentley of <a href="http://www.softmake.com.au">www.softmake.com.au</a></div>
<div data-bbox="702 938 822 956" data-label="Page-Footer">Page 81 of 103</div>
```

```

</script>
</body>
</html>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\scripting.xhtml
(W3C HTML5.3, 2017), "The Script Element", <https://www.w3.org/TR/html53/semantics-scripting.html#the-script-element>

The script element shouldn't contain a type element for the JavaScript Mime type. Just depend on that being the default.

```

<!-- do this -->
<script>
  /*  */
  addLoadEvent(alert("hi"));
  /* ]]&gt; */
&lt;/script&gt;

&lt;!-- not this --&gt;
&lt;script type="text/ecmascript"&gt;
  /* <![CDATA[ */
  addLoadEvent(alert("hi"));
  /* ]]&gt; */
&lt;/script&gt;
</pre>
</div>
<div data-bbox="125 390 853 430" data-label="Text">
<p>C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\scripting.xhtml<br/>
(W3C HTML5.3, 2017), "The Script Element", <a href="https://www.w3.org/TR/html53/semantics-scripting.html#the-script-element">https://www.w3.org/TR/html53/semantics-scripting.html#the-script-element</a></p>
</div>
<div data-bbox="115 445 800 476" data-label="Text">
<p>To have the script element interpret the referenced JavaScript "according to the JavaScript Module top-level production" set <code>type</code> to "module".</p>
</div>
<div data-bbox="115 492 182 507" data-label="Section-Header">
<h2>noscript</h2>
</div>
<div data-bbox="115 521 569 537" data-label="Text">
<p>Polyglot: don't use <code>noscript</code> (xml doesn't allow it).</p>
</div>
<div data-bbox="151 553 885 607" data-label="Text">
<p>The noscript element is only effective in the HTML syntax, it has no effect in the XHTML syntax. This is because the way it works is by essentially "turning off" the parser when scripts are enabled, so that the contents of the element are treated as pure text and not as real elements. XML does not define a mechanism by which to do this.</p>
</div>
<div data-bbox="125 621 879 649" data-label="Text">
<p>(W3C HTML5.3, 2017), "The noscript element", <a href="https://www.w3.org/TR/html53/semantics-scripting.html#the-noscript-element">https://www.w3.org/TR/html53/semantics-scripting.html#the-noscript-element</a></p>
</div>
<div data-bbox="115 665 869 695" data-label="Text">
<p>Instead of <code>noscript</code> you can use a pseudo-noscript block to warn users that javascript is turned off. Do this by using <code>script</code> to hide or remove the pseudo-noscript block.</p>
</div>
<div data-bbox="125 710 670 905" data-label="Text">
<pre>
&lt;!DOCTYPE html&gt;
&lt;html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en"&gt;
&lt;head&gt;
  &lt;meta charset="utf-8" /&gt;
  &lt;title&gt;Noscript Substitute&lt;/title&gt;

  &lt;link rel="stylesheet" href="style/html5-basic.css" /&gt;
  &lt;style&gt;
    /* <![CDATA[ */
    #psuedoNoscript{
      color: red;
      outline: 1px solid;
      padding: 1em;
    }
    /* ]]&gt; */
  &lt;/style&gt;

  &lt;script&gt;
</pre>
</div>
<div data-bbox="115 938 320 955" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div>
<div data-bbox="363 938 663 956" data-label="Page-Footer">John Bentley of <a href="http://www.softmake.com.au">www.softmake.com.au</a></div>
<div data-bbox="703 938 822 956" data-label="Page-Footer">Page 82 of 103</div>
```

```

/*  */
window.onload = function() {
    disapperPsuedoNoscript();
}

function disapperPsuedoNoscript () {
    var element = document.getElementById("psuedoNoscript");
    element.hidden = true;
}
/*  */
</script>
</head>
<body>
<div id="psuedoNoscript">
    You have your javascript turned off. This page will still work, but with reduced
    functionality.
</div>
<h1>Noscript Substitute</h1>
<p>Lorem</p>
</body>
</html>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\noscript-substitute.xhtml

template

The template element is used to declare fragments of HTML that can be cloned and inserted in the document by script.

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<meta charset="utf-8" />
<title>Template Element</title>

<link rel="stylesheet" href="style/html5-basic.css" />
<script>
/*  */
function main() {
    // Data is hard-coded here, but could come from the server
    var data = [
        { name: 'Pillar', color: 'Tabby', sex: 'Female', legs: 3 },
        { name: 'Hedral', color: 'Tuxedo', sex: 'Male', legs: 4 },
    ];

    var template = document.querySelector('#row');

    for (var i = 0; i &lt; data.length; i += 1) {
        var cat = data[i];
        var clone = template.content.cloneNode(true);
        var cells = clone.querySelectorAll('td');
        cells[0].textContent = cat.name;
        cells[1].textContent = cat.color;
        cells[2].textContent = cat.sex;
        cells[3].textContent = cat.legs;
        template.parentNode.appendChild(clone);
    };

    window.onload = main;
/*  */
</script>
</head>
<body>
<h1>Template Element</h1>
<table>
<thead>
<tr>
<th>Name</th> <th>Color</th> <th>Sex</th> <th>Legs</th>
</tr>
</thead>
<tbody>
<template id="row">

```

```

    <tr>
      <td></td> <td></td> <td></td> <td></td>
    </tr>
  </template>
</tbody>
</table>
</body>
</html>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\template.xhtml
(W3C HTML5.3, 2017), "The template element", <https://www.w3.org/TR/html53/semantics-scripting.html#the-template-element>

slot

Works with templates.

Mozilla MDN web docs > Using templates and slots > https://developer.mozilla.org/en-US/docs/Web/Web_Components/Using_templates_and_slots

Scripting animations

css

See (Bentley, 2018. CascadingStyleSheets-Css-Reference.docx, file://Atlas/Users/John/Documents/Sda/Info/Web/KB/Css/Reference/CascadingStyleSheet-Css-Reference.docx) <..\..\Css\Reference\CascadingStyleSheets-Css-Reference.docx>

svg

See [ScalableVectorGraphics-SVG.docx](#).

canvas

Canvas static

The canvas element graphics library specifications:

(W3C, 2015. HTML Canvas 2D Context W3C Recommendation 19 November 2015, <https://www.w3.org/TR/2dcontext/>)

The canvas element provides scripts with a resolution-dependent bitmap canvas, which can be used for rendering graphs, game graphics, art, or other visual images on the fly.

(W3C HTML5.3, 2017), "The Canvas Element", <https://www.w3.org/TR/html53/semantics-scripting.html#the-canvas-element>

When authors use the canvas element, they must also provide content that, when presented to the user, conveys essentially the same function or purpose as the canvas's bitmap. ... The contents of the canvas element... are the element's fallback content.

```

<body>
  <h1>Canvas</h1>

  <canvas id="myCanvas">
    
  </canvas>
</body>

```

(W3C HTML5.3, 2017), "The Canvas Element", <https://www.w3.org/TR/html53/semantics-scripting.html#the-canvas-element>

The canvas element can contain visual media that is: interactive; or non-interactive (static).

(W3C HTML5.3, 2017), "The Canvas Element", <https://www.w3.org/TR/html53/semantics-scripting.html#the-canvas-element>

The width attribute defaults to 300, and the height attribute defaults to 150.

(W3C HTML5.3, 2017), "The Canvas Element", <https://www.w3.org/TR/html53/semantics-scripting.html#the-canvas-element>

Canvas non-interactive example.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>Canvas</title>

  <link rel="stylesheet" href="style/html5-basic.css" />
  <script>
    /*  */

    function main () {
      var canvas = document.getElementById("myCanvas");
      var context = canvas.getContext("2d");

      // Create a circle
      context.beginPath();
      context.arc(95, 50, 40, 0, 2 * Math.PI);
      context.stroke();

      // Add some text
      context.font = "30px Arial";
      context.fillText("Hello World", 50, 130);

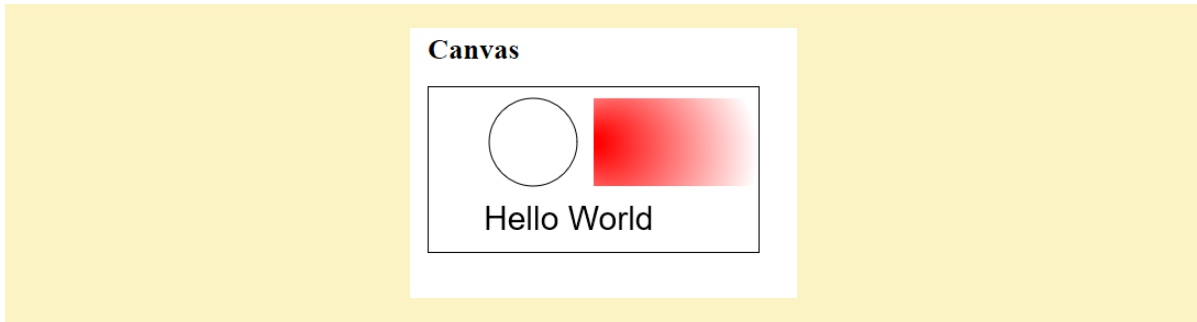
      // Create gradient
      var gradient = context.createRadialGradient(150, 50, 5, 200, 60, 100);
      gradient.addColorStop(0, "red");
      gradient.addColorStop(1, "white");

      // Fill with gradient
      context.fillStyle = gradient;
      context.fillRect(150, 10, 150, 80);
    }

    window.onload = main;

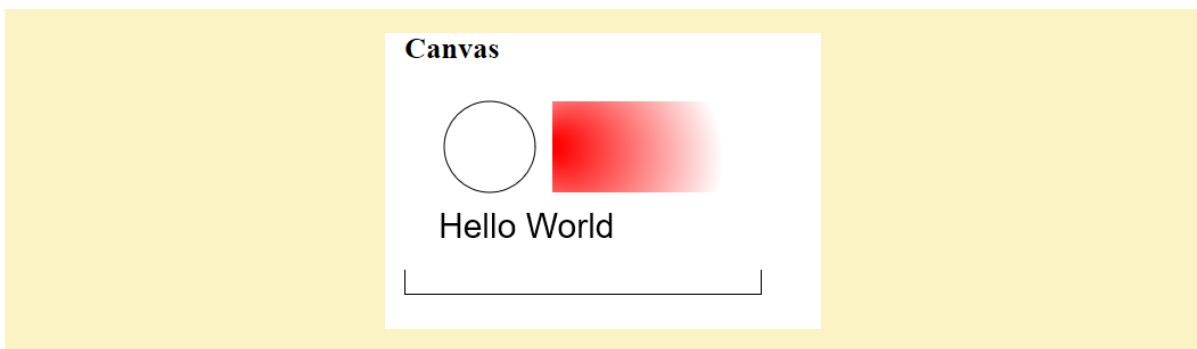
    /* ]]&gt; */
  &lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
  &lt;h1&gt;Canvas&lt;/h1&gt;

  &lt;canvas id="myCanvas"&gt;
    &lt;img src="canvas-fallback.png" alt="A circle, 'Hello World', and a Gradient" /&gt;
  &lt;/canvas&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="115 938 320 955" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div><div data-bbox="363 938 663 956" data-label="Page-Footer">John Bentley of www.softmake.com.au</div><div data-bbox="703 938 822 956" data-label="Page-Footer">Page 85 of 103</div>
```



C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\canvas.xhtmll
 (W3C HTML5.3, 2017), "The Canvas Element", <https://www.w3.org/TR/html53/semantics-scripting.html#the-canvas-element>

In production don't use a border around canvas to avoid fallback ugliness when javascript is disabled:



C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\canvas.xhtmll

The canvas element has a context: none, "2d", "webgl", or "webgl2".

```
context = canvas . getContext(contextId [, ... ] )
```

```
function main () {
  var canvas = document.getElementById("myCanvas");
  var context = canvas.getContext("2d");

  // Create a circle
  context.beginPath();
```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\canvas.xhtmll
 (W3C HTML5.3, 2017), "The Canvas Element", <https://www.w3.org/TR/html53/semantics-scripting.html#the-canvas-element>

Canvas animations

Probably the biggest limitation is, that once a shape gets drawn, it stays that way. If we need to move it we have to redraw it and everything that was drawn before it.

(Mozilla 2018, "Canvas Tutorial: Basic Animations", https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations) https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations

Canvas animations are done by:

1. Clearing the canvas. Everything needs to be redrawn in a new position and so you need to clear the prior image(s) to avoid ghosting.
2. Save canvas state.
3. Draw the shape(s) in a new position.
4. Restore canvas state.

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>Canvas - With Animations</title>

  <style>
    /*  */
    canvas {
      /* Don't use a border in production to prevent fallback ugliness */
      border: 1px solid black;
    }
    /* ]]&gt; */
  &lt;/style&gt;

  &lt;script&gt;
    /* <![CDATA[ */
    // User input
    var startX = 50;
    var endX = 200;
    var totalTime = 2000; // Milliseconds

    // Scaffold variables
    var output = null;
    var distanceX = endX - startX;
    var elapsedTime = 0;
    var stopped = false;
    var requestId = 0;
    var startTime = 0;
    var canvas = null;
    var context = null;

    function drawCircle(x){
      // Animations require everything to be redrawn. So let's clear any prior image.
      context.clearRect(0, 0, 300, 150);
      context.save();
      context.beginPath();
      context.arc(x, 50, 40, 0, 2 * Math.PI);
      context.stroke();
      context.restore();
    }

    function main() {
      canvas = document.getElementById("myCanvas");
      context = canvas.getContext("2d");

      // Create a circle
      drawCircle(startX);
    }
    window.onload = main;

    // The callback function
    function animate(time) {
      if (!stopped) {
        elapsedTime = (Date.now() - startTime);
        position = startX + (elapsedTime/totalTime) * distanceX;
        output.textContent = position;
        drawCircle(position);
        if (position &gt;= endX ) { stopped = true };
        // Recursive callback
        requestId = window.requestAnimationFrame(animate);
      }
    }

    function start() {
      // Get the first rect.
      output = document.getElementsByTagName("output")[0];
      startTime = Date.now();
</pre>
</div>
<div data-bbox="114 938 320 955" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div>
<div data-bbox="363 938 663 956" data-label="Page-Footer">John Bentley of www.softmake.com.au</div>
<div data-bbox="703 938 822 956" data-label="Page-Footer">Page 87 of 103</div>
```

```

        stopped = false;
        requestId = window.requestAnimationFrame(animate);
    }

    function stop () {
        if (requestId) {
            window.cancelAnimationFrame(requestId);
        }
        stopped = true;
    }

    function reset () {
        stop();
        position = startX;
        output.textContent = position;
        drawCircle(position);
    }
    /* ]]> */
</script>
</head>
<body>
    <h1>Canvas - With Animations</h1>

    <canvas id="myCanvas">
        
    </canvas>

    <button onclick="start();" type="button">Start</button>
    <button onclick="stop();" type="button">Stop</button>
    <button onclick="reset();" type="reset">Reset</button>
    <output></output>

</body>
</html>

```

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\canvas-020-with-animations.xhtml
 (Mozilla 2018, "Canvas Tutorial: Basic Animations", https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations) https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial/Basic_animations

WebGL

WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 3D and 2D graphics within any compatible web browser without the use of plug-ins. WebGL does so by introducing an API that closely conforms to OpenGL ES 2.0 that can be used in HTML5 <canvas> elements.

Mozilla, WebGL: 2D and 3D graphics for the web, https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

The Specs.

<https://www.khronos.org/registry/webgl/specs/latest/1.0/>
<https://www.khronos.org/registry/webgl/specs/latest/2.0/>

WebGL example: <http://webgl-samples.org/aquarium/aquarium.html>

WebGL is a somewhat complicated API requiring specialisation.

(Bentley, n.d.)

Custom elements

Common idioms without dedicated elements

Disabled elements

Attributes

Global

See also [Attribute Datatypes](#).

Global attributes:

Common	Event Handlers	Aria	Custom data	
accesskey class contenteditable dir hidden id lang spellcheck style tabindex title translate	onabort onblur# oncancel oncanplay oncanplaythrough onchange onclick oncuechange ondblclick ondurationchange onemptied onended onerror# onfocus# oninput oninvalid onkeydown onkeypress onkeyup onload# onloadeddata onloadedmetadata onloadstart onmousedown onmouseenter onmouseleave	onmousemove onmouseout onmouseover onmouseup onmousewheel onpause onplay onplaying onprogress onratechange onreset onresize# onscroll# onseeked onseeking onselect onshow onstalled onsubmit onsuspend ontimeupdate ontoggle onvolumechange onwaiting	role aria-*	data-*

" The attributes marked with an [hash #] have a different meaning when specified on body elements as those elements expose event handlers of the Window object with the same names."

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html/dom.html#global-attributes>

Common

title

The `title` attribute contains advisory information for the element. `Link`, `abbr`, and `input` have additional semantics for `title`. Otherwise advisory information could include, for example:

- On an image, image credit or image description;

- On a paragraph, a footnote or commentary on the text;
- On a citation, further information about the source;
- On interactive content, a label for, or instructions for, the use of an element.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html/dom.html#the-title-attribute>

Relying on the title attribute is discouraged.

Many user agents do not expose the title attribute accessibly. E.g. While on the desktop a mouse pointer might expose a title on a field as a tooltip, mobile users don't have such pointers.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html/dom.html#the-title-attribute>

translate

The `translate` attribute determines whether the element's attribute values and text node children are to be translated. Possible values: empty string, yes, and no.

```
<p>When the game launches, the first thing you should do is type
<kbd translate="no">eat honey</kbd>.</p>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html/dom.html#the-translate-attribute>

id

The id attribute automatically turns the element into an anchor.

(Castro, 2003, p.67)

type

The `type` attribute can be used in `link` and `style` elements.

```
<link rel="stylesheet" href="mystyle.css" type="text/css">
<!-- type attribute not necessary for css -->
<style type="text/css">
  /*  */

  h1 {
    color: blue;
  }
  /*  */
</style>
```

For style blocks use of `type="text/css"`, for css, is discouraged (by the W3C validator):

```
<!-- Encouraged -->
<style>
  /*  */

  h1 {
    color: blue;
  }
  /*  */
</style>

<!-- Discouraged -->
```

```
<style type="text/css">
  /*  */

  h1 {
    color: blue;
  }
  /* ]]&gt; */
&lt;/style&gt;</pre>
</div>
<div data-bbox="125 187 871 215" data-label="Text">
<p>The HTML spec stipulates that in the style element the type attribute defaults to "text/css" if the type attribute is omitted:</p>
</div>
<div data-bbox="115 229 820 245" data-label="Text">
<p>| The default value for the type attribute, which is used if the attribute is absent, is "text/css". [RFC2318]</p>
</div>
<div data-bbox="125 260 733 275" data-label="Text">
<p>The HTML spec does not specifically discourage the use of type="text/css" in style blocks.</p>
</div>
<div data-bbox="125 289 880 318" data-label="Text">
<p>However the validator (<a href="https://validator.w3.org/nu/">https://validator.w3.org/nu/</a>) warns against its use with "Warning: The type attribute for the style element is not needed and should be omitted".</p>
</div>
<div data-bbox="125 332 859 360" data-label="Text">
<p>(W3C HTML5, 2014. <i>HTML5, A vocabulary and associated APIs for HTML and XHTML</i>, W3C Recommendation 28 October 2014) <a href="https://www.w3.org/TR/html5/document-metadata.html#the-style-element">https://www.w3.org/TR/html5/document-metadata.html#the-style-element</a></p>
</div>
<div data-bbox="115 375 848 391" data-label="Text">
<p>For the link element the use of <code>type="text/css"</code>, for css, is discouraged (by JLB).</p>
</div>
<div data-bbox="125 406 599 461" data-label="Text">
<pre>&lt;!-- Encouraged --&gt;
&lt;link rel="stylesheet" href="mystyle.css"&gt;

&lt;!-- Discouraged --&gt;
&lt;link rel="stylesheet" href="mystyle.css" type="text/css"&gt;</pre>
</div>
<div data-bbox="125 477 826 505" data-label="Text">
<p>The HTML spec doesn't encourage or discourage the use of <code>type="text/css"</code> in the link element. However, the HTML spec makes it clear it is optional. It has an example as follows:</p>
</div>
<div data-bbox="175 520 586 553" data-label="Text">
<pre>&lt;link rel="stylesheet" href="A" type="text/plain"&gt;
&lt;link rel="stylesheet" href="B" type="text/css"&gt;
&lt;link rel="stylesheet" href="C"&gt;</pre>
</div>
<div data-bbox="125 568 865 596" data-label="Text">
<p>The validator (<a href="https://validator.w3.org/nu/">https://validator.w3.org/nu/</a>) doesn't provide any warning if <code>type="text/css"</code> is link element.</p>
</div>
<div data-bbox="125 611 873 640" data-label="Text">
<p>However, for consistency with the rule on its use in the style block I discourage its use on the link block, when <code>rel="stylesheet"</code> is also used (not that you'd used it when <code>rel="stylesheet"</code> is not used).</p>
</div>
<div data-bbox="125 654 872 682" data-label="Text">
<p>(W3C HTML5.3, 2017), "The link element", "Processing the type attribute", <a href="https://www.w3.org/TR/html5/document-metadata.html#processing-link-type">https://www.w3.org/TR/html5/document-metadata.html#processing-link-type</a></p>
</div>
<div data-bbox="115 727 277 745" data-label="Section-Header">
<h2>Custom Data</h2>
</div>
<div data-bbox="125 762 858 821" data-label="Text">
<p>A custom data attribute is an attribute in no namespace whose name starts with the string "data-", has at least one character after the hyphen, is XML-compatible, and contains no uppercase ASCII letters. ... Every HTML element may have any number of custom data attributes specified, with any value.</p>
</div>
<div data-bbox="125 835 624 880" data-label="Text">
<pre>&lt;ol&gt;
  &lt;li data-length="2m11s" data-fuck="true"&gt;Beyond The Sea&lt;/li&gt;
  ...
&lt;/ol&gt;</pre>
</div>
<div data-bbox="115 938 320 955" data-label="Page-Footer">SaveDate: 2021-07-07 00:21</div>
<div data-bbox="363 938 663 956" data-label="Page-Footer">John Bentley of <a href="http://www.softmake.com.au">www.softmake.com.au</a></div>
<div data-bbox="702 938 822 956" data-label="Page-Footer">Page 91 of 103</div>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html/dom.html#custom-data-attribute>

`element.dataset`. Returns a `DOMStringMap` object for the element's `data-*` attributes. Hyphenated names become camel-cased. For example, `data-foo-bar= " "` becomes `element.dataset.fooBar`.

```
<div class="spaceship" data-ship-id="92432"
    data-weapons="laser 2" data-shields="50%"
    data-x="30" data-y="10" data-z="90">
  <button class="fire"
    onclick="spaceships[this.parentNode.dataset.shipId].fire()">
    Fire
  </button>
</div>
```

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html/dom.html#custom-data-attribute>

Custom data attributes are intended to store custom data private to the page or application, for which there are no more appropriate attributes or elements. ... These attributes are intended for use by the site's own scripts, and are not a generic extension mechanism for publicly-usable metadata.

(W3C HTML5, 2014. *HTML5, A vocabulary and associated APIs for HTML and XHTML*, W3C Recommendation 28 October 2014) <https://www.w3.org/TR/html/dom.html#custom-data-attribute>

Aria

Additional Semantics

Overview

Candidate mechanisms for semantics over and above that defined in HTML5 are divided into two types:

1. Public use extensions; and
2. Private use extensions.

Public use extension candidates

Meta element

Using the meta element with extensions, possibly in conjunction with the link element using [Other link types](#).

```
<!-- Link type extensions
http://microformats.org/wiki/existing-rel-values#HTML5 link type extensions -->
<link rel="schema.dc" href="http://purl.org/dc/elements/1.1/" />

<!-- Example meta name extensions, using dublin core.
See: https://wiki.whatwg.org/wiki/MetaExtensions -->
<meta name="dc.title" content="HTML 5 - Polyglot - Template - Example" />
<meta name="dc.creator" content="John Bentley" />
```

<https://www.w3.org/TR/html5/document-metadata.html#the-meta-element>,
<https://wiki.whatwg.org/wiki/MetaExtensions>.

Microformats2

```
<div class="h-card">
  
  <a class="p-name u-url"
    href="http://blog.lizardwrangler.com/"
  >Mitchell Baker</a>
  (<a class="u-url"
    href="https://twitter.com/MitchellBaker"
  >@MitchellBaker</a>)
  <span class="p-org">Mozilla Foundation</span>
  <p class="p-note">
    Mitchell is responsible for setting the direction and scope of the Mozilla Foundation
    and its activities.
  </p>
  <span class="p-category">Strategy</span>
  <span class="p-category">Leadership</span>
</div>
```

"Microformats2 replaces and supersedes both classic microformats, as well as incorporates lessons learned from microdata and RDFa." <http://microformats.org/wiki/microformats2>

<https://en.wikipedia.org/wiki/Microformat> , <http://microformats.org/wiki/microformats2>

Microdata (Deprecated).

```
<!-- Microdata, in this example using schema.org microdata -->
<section itemscope itemType="http://schema.org/Person">
  Hello, my name is
  <span itemprop="name">John Doe</span>,
  I am a
  <span itemprop="jobTitle">graduate research assistant</span>
```

```

at the
<span itemprop="affiliation">University of Dreams</span>.
My friends call me
<span itemprop="additionalName">Johnny</span>.
You can visit my homepage at
<a href="http://www.JohnnyD.com" itemprop="url">www.JohnnyD.com</a>.
<section itemprop="address" itemscope itemType="http://schema.org/PostalAddress">
  I live at
  <span itemprop="streetAddress">1234 Peach Drive</span>,
  <span itemprop="addressLocality">Warner Robins</span>,
  <span itemprop="addressRegion">Georgia</span>.
</section>
</section>

```

"The W3C HTML Working Group failed to find an editor for the specification and terminated its development with a 'Note'."

https://en.wikipedia.org/wiki/Microdata_%28HTML%29

Private use extensions: See [Custom Data](#).

Dublic Core

The basic history of Dublin Core is:

- Dublin Core Terms (title, issued, etc) are defined in one of two namespaces (old and deprecated) <http://purl.org/dc/elements/1.1/> and (new) <http://purl.org/dc/terms/>
- Terms from <http://purl.org/dc/elements/1.1/> are deprecated.

https://en.wikipedia.org/wiki/Dublin_Core#Qualified_Dublin_Core_..28deprecated_in_2012.5B17.5D.29

- In (X)HTML you must declare a namespace when using a dublic core term.
- For example, in the (X)HTML head you use:

```

<link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
<meta name="DC.title" content="Services to Government" />

```

- The prefixes "DC" or "DCTERMS" have no fixed meaning under Dublic Core ...

It is important to note that the use of prefixes such as "DC" or "DCTERMS" is simply a community convention. There is no requirement that these prefixes are associated with the namespace URIs <http://purl.org/dc/elements/1.1/> and <http://purl.org/dc/terms/>. See <http://dublincore.org/documents/dc-html/>

- ... but under <https://wiki.whatwg.org/wiki/MetaExtensions> (Part of HTML5 conventions) the registered prefixes are restricted to be associated as:
 - dc: <http://purl.org/dc/elements/1.1/>
 - dcterms: <http://purl.org/dc/terms/>

Therefore a contemporary use of Dublin Core, conforming to HTML5 as partly determined by <https://wiki.whatwg.org/wiki/MetaExtensions>, goes something like

```

<link rel="schema.DCTERMS" href="http://purl.org/dc/terms" />
<meta name="DCTERMS.title" content="Services to Government" />
<meta name="DCTERMS.modified" scheme="DCTERMS.W3CDTF" content="2016-06-10T20:00:09+1000" />
<!-- or -->

```

```
<link rel="schema.DCTERMS" href="http://purl.org/dc/terms" />
<link rel="schema.XSD" href="http://www.w3.org/2001/XMLSchema#" />
<meta name="DCTERMS.title" content="Services to Government" />
<meta name="DCTERMS.modified" scheme="XSD.date" content="2007-07-22" />
```

(Dublin Core Metadata Initiative, 2008. *Expressing Dublin Core metadata using HTML/XHTML meta and link elements*) <http://dublincore.org/documents/dc-html/> (derived from Example 24 and *fix).

The list of contemporary Dublic Core terms:

<http://dublincore.org/documents/2012/06/14/dcmi-terms/>

(Dublic Core Metadata Initiative, 2012. *DCMI Metadata Terms*)

Google's JSON-LD with Schema.org

Google is promoting JSON-LD for metadata in (X)HTML.

```
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "Organization",
  "url": "http://www.your-company-site.com",
  "contactPoint": [{
    "@type": "ContactPoint",
    "telephone": "+1-401-555-1212",
    "contactType": "customer service"
  }]
}
</script>
```

"You provide structured data markup in your HTML ... pages... JSON-LD is the recommended format. Google is in the process of adding JSON-LD support for all markup-powered features. The table below lists the exceptions to this. We recommend using JSON-LD where possible."

<https://developers.google.com/search/docs/guides/intro-structured-data>

Google notes that with JSON-LD you typically use:

"the schema.org vocabulary – an open community effort to promote standard structured data in a variety of online applications."

<https://developers.google.com/search/docs/guides/intro-structured-data>

Schema.org appears to enforce datetimes in long form only:

```
// http://www.bbc.com/news/science-environment-36505748
"datePublished": "2016-06-11T09:05:50+01:00"
```

"A combination of date and time of day in the form [-]CCYY-MM-DDThh:mm:ss[Z|(+|-)hh:mm] (see Chapter 5.4 of ISO 8601)."

<https://schema.org/DateTime>

Templates

Minimum, must

JLB Polyglot: the absolute minimum required to conform to the specifications of polyglot markup. In other words, the minimum to conform to the MUST specifications of polyglot markup.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="" xml:lang="">
<head>
  <title>Title</title>
</head>
<body>
</body>
</html>
```

Language: Providing attributes with blank contents has meaning under the HTML5 spec, section "3.2.5.3. The lang and xml:lang attributes", "unknown", and seems to be a MUST (written as "must") if a language value is not provided ...

"The lang attribute (in no namespace) specifies the primary language for the element's contents and for any of the element's attributes that contain text. Its value must be a valid BCP 47 language tag, or the empty string. Setting the attribute to the empty string indicates that the primary language is unknown."

Title: The Polyglot spec lists its template with an empty <title> element. However, by HTML 5 <title> can't contain inter-element whitespace (which entails an empty element). <https://validator.w3.org/nu/> enforces that rule.

C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-010-minimum-must.xhtml

(W3C Polyglot, 2015. Polyglot Markup: A robust profile of the HTML5 vocabulary, W3C Candidate Recommendation) <http://www.w3.org/TR/html-polyglot/#minimal-polyglot-html-document>

(W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/document-metadata.html#the-title-element>

<https://www.w3.org/TR/html/dom.html#the-lang-and-xml:lang-attributes>

[default.html5. Polyglot Markup. #237. Comment 2017-01-27 14:59](<https://github.com/jgm/pandoc-templates/issues/237#issuecomment-275120456>)

Minimum, should

JLB Polygot: the minimum required to conform to the SHOULD specifications of polyglot markup.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <meta charset="utf-8" />
  <title>Title</title>
</head>
<body>
</body>
</html>
```

Polyglot "4.8.2 Language attributes": "The root element SHOULD always specify the language". See [Language Declaration](#).

Polyglot recommends you specify the character encoding. See [Character Encoding](#).

The Polyglot spec lists its template with an empty <title> element. However, by HTML 5 <title> can't contain inter-element whitespace (which entails an empty element). <https://validator.w3.org/nu/> enforces that rule.

C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-020-minimum-should.xhtml
 (W3C Polyglot, 2015. Polyglot Markup: A robust profile of the HTML5 vocabulary, W3C Candidate Recommendation) <https://www.w3.org/TR/html-polyglot/#language-attributes>
 (W3C HTML5, 2014. HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014) <http://www.w3.org/TR/html5/document-metadata.html#the-title-element>

Example

JLB: a basic file to be used for examples.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
  <!--
    File Extension and MimeType.
    This file has an extension .xhtml in order to be served with a Mime Type (Content Type):
    application/xhtml+xml. Under HTML 5 the only permitted value for content="",
    when <meta http-equiv="content-type" ...>, is "text/html". Therefore do not delete this
    comment to, instead, express mime type with
    <meta http-equiv="content-type" content="application/xhtml+xml" />
  -->
  <meta charset="utf-8" />

  <!-- Make page mobile friendly -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <title>Title</title>

  <link rel="stylesheet" href="style/html5-basic.css" />
  <script>*<![CDATA[*/*]]>*</script>
</head>
<body>
  <h1>Title</h1>
  <p>Lorem</p>
</body>
</html>
```

C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-030-example.xhtml

Forms

See [form](#) above.

C:\Users\John\Documents\Sda\Code\web\Examples\Html5Examples\forms-all-controls-div-enclosing-the-template.xhtml

Project

JLB Polyglot: a basic file to be used as the default for project files (production websites with multiple pages).

```
<!DOCTYPE html>
<html xmlns='http://www.w3.org/1999/xhtml' lang='en' xml:lang='en'>
<head>
  <!--
    File Extension and MimeType.
    This file has an extension .xhtml in order to be served with a Mime Type (Content Type):
    application/xhtml+xml. Under HTML 5 the only permitted value for content='',
    when <meta http-equiv='content-type' ...>, is 'text/html'. Therefore do not delete this
    comment to, instead, express mime type with
    <meta http-equiv='content-type' content='application/xhtml+xml' />
  -->
```

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

<!-- Make page mobile friendly -->
<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Title</title>

<meta name='description' content='Ensure this is informative for the user.
160 characters max. Displays in search engine results. Gimpe dit molestias
nihil numquam odio qui voluptate, voluptatum.' />
<meta name='author' content='John Bentley' />

<link rel='icon' href='images/favicon/my-favicon.png' />

<link rel='stylesheet' href='style/html5-reset.org.jlb.css' />
<link rel='stylesheet' href='style/html5-project.css' />
<script>*<![CDATA[*/*
/*]]>*</script>
</head>

<body>

<div id='logo'>
  <a href="html5-polyglot-template-040-project.xhtml">Website Name</a>
</div>

<nav>
  <ul>
    <li>
      <a href='#'>Lorem</a>
    </li>
    <li>
      <a href='#'>Dolor</a>
    </li>
    <li>
      <a href='#'>Consectetur</a>
    </li>
  </ul>
</nav>

<main>
  <article>

    <header>
      <h1>Heading 1 Level Title</h1>

      <div>
        <address>Author: Lisa Smith</address>

        <p>
          Date published:
          <time>2016-09-07 18:47</time>
        </p>

        <p id="abstract">Abstract: All about lorem. utem, neque maiores. Aut quisquam
          quidem, maiores eaque ratione sed accusamus velit atque
          perspiciatis iusto dolore laudantium aliquam harum corporis
          cupiditate aspernatur voluptatibus dolor? </p>
        </div>
      </header>

      <section>
        <!-- An alternative would be to use H1, given it is a child of a section,
        but the default recommendation is to step down the h number -->
        <h2>Lorem</h2>

        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Accusantium
        minus, hic, libero quaerat enim aperiam illum. Laudantium
        ea nemo perspiciatis voluptatibus error debitis dicta, iste
        architecto incidunt maxime tenetur sint necessitatibus deleniti
        ipsa deserunt repudiandae harum facere quo praesentium suscipit
        placeat saepe.</p>
      </section>

      <section>
        <h2>Ipsum</h2>
      </section>
    </article>
  </main>
</body>
```

```
<p>Hic fuga, temporibus laudantium earum animi nulla cupiditate  
    ipsum optio amet explicabo mollitia at sequi soluta porro,  
    repudiandae reiciendis quos officia vero placeat! </p>  
</section>  
  
</article>  
</main>  
  
</body>  
</html>
```

C:\Users\John\Documents\Sda\Code\web\Libraries\Html5Library\html5-polyglot-template-040-project.xhtml

Abbreviations

JLB
jlb

Stipulations of a John Bentley convention (above what Polyglot markup or the HTML spec provides).

References, Zotero

Bentley, J., (2015-Jun-09). *XML Quick Reference*. [Electronic Source].

Bentley, J., (2016-Mar-16). *Web Code Convention - Naming and Style - General*. [Electronic Source]
Available at: [file:///Atlas/Users/John/Documents/Sda/Info/Web/System/Conventions/Web Code Convention - Naming and Style - General.docx](file:///Atlas/Users/John/Documents/Sda/Info/Web/System/Conventions/Web%20Code%20Convention%20-%20Naming%20and%20Style%20-%20General.docx).

Bentley, J., (2018). *CascadingStyleSheets-Css-Reference.docx*. [Electronic Source]
Available at:
<file:///Atlas/Users/John/Documents/Sda/Info/Web/KB/Css/Reference/CascadingStyleSheets-Css-Reference.docx>.

Bentley, J., (2018). *HTML Polyglot Markup Rules*. [Electronic Source]
Available at: [\\Atlas\Users\John\Documents\Sda\Info\Web\KB\Xhtml\Reference\Html-PolyglotMarkupRules.docx](file:///Atlas/Users/John/Documents/Sda/Info/Web/KB/Xhtml/Reference/Html-PolyglotMarkupRules.docx).

Bentley, J., n.d. *Experimentally Verified*.

Castro, E., 2003. *HTML for the world wide web with XHTML and CSS*. 5th ed. Peachpit Press.

Deveria, A., n.d. *Can I use*. [Online] Available at: <https://caniuse.com/>.

Dublic Core Metadata Initiative, (2012-06-14). *DCMI Metadata Terms*. [Online]
Available at: <http://dublincore.org/documents/2012/06/14/dcmi-terms/>.

Dublin Core Metadata Initiative, (2008-08-04). *Expressing Dublin Core metadata using HTML/XHTML meta and link elements*. [Online]
Available at: <http://dublincore.org/documents/dc-html/>.

Google Webmaster Central Blog, 2009. *Google does not use the keywords meta tag in web ranking*.
Available at: <https://webmasters.googleblog.com/2009/09/google-does-not-use-keywords-meta-tag.html>.

Jing, C.H., (2015-Oct-06). *Using Responsive Images (Now)*. [Online]
Available at: <https://alistapart.com/article/using-responsive-images-now>.

Mozilla, (2016). [Online]
Available at: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML>.

Stack Exchange Inc, 2015. *Stackoverflow*. [Online] Available at: <http://stackoverflow.com/>.

W3C HTML5.3, 2017. *HTML5.3*. [Online] (First Public Working Draft) Available at:
<https://www.w3.org/TR/html53/>.

W3C HTML5, (2014-Oct-28). *HTML5, A vocabulary and associated APIs for HTML and XHTML, W3C Recommendation 28 October 2014*. [Online]
Available at: <https://www.w3.org/TR/html/>.

W3C Polyglot, (2015-Sep-29). *Polyglot Markup: A robust profile of the HTML5 vocabulary, W3C Candidate Recommendation*. [Online]
Available at: <http://www.w3.org/TR/html-polyglot/>
[Accessed 2016-Mar-17].

W3C, (2015-Nov-19). *HTML Canvas 2D Context* W3C Recommendation 19 November 2015. [Electronic Source]

Available at: <https://www.w3.org/TR/2dcontext/>.

W3C, 2015. *HTML5 Image Description Extension (longdesc)*. W3C Recommendation 26 February 2015.

[Online] Available at: <https://www.w3.org/TR/html-longdesc/>.

W3C, (2016-Sep-26). *CSS Values and Units Module Level 3*, W3C Candidate Recommendation. [Online]

Available at: <https://www.w3.org/TR/css3-values/>.

W3C, (2016-Jun-07). *WebVTT: The Web Video Text Tracks Format*. [Online]

Available at: <https://w3c.github.io/webvtt/>.

W3C, (2017-Nov-02). *HTML 5.2*. [Online]

Available at: <https://www.w3.org/TR/html52/>.

Web Hypertext Application Technology Working Group (WHATWG), (2016-Jun-01). *MetaExtensions*. [Online]

Available at: <https://wiki.whatwg.org/wiki/MetaExtensions>.

Web Hypertext Application Technology Working Group (WHATWG), (n.d.). *HTML Standard, Developer's Edition*. [Online]

Available at: <https://html.spec.whatwg.org/dev/>.

webplatform.org, (2016). [Online]

Available at: http://docs.webplatform.org/wiki/Main_Page.

Document Licence

[Xhtml - Reference](#) © 2021 by [John Bentley](#) is licensed under [Attribution-NonCommercial-ShareAlike 4.0 International](#)

