## The main Element

The <main> element should contain the main content for your web page. All of this content should be unique to the individual page, and should not appear elsewhere on the site. Any content that is repeated on multiple pages (logos, search boxes, footer links, etc.) should not be placed within the <main> element.

The example below uses a <main> element to represent the main content for the page

<body>

<header>

<div id="logo">Rocking Stone</div>

<nav>...</nav>

</header>

<main role="main">

<h1>Guitars</h1>

<p>The greatest guitars ever built.</p>

<article>

<h2>Gibson SG</h2>

<p>...</p>

</article>

<article>

<h2>Fender Telecaster</h2>

<p>...</p>

</article>

</main>

</body>

<canvas> element

Description:

The HTML5 Canvas element is an HTML tag similar to the <div>, <a>, or <table> tag, with the exception that its contents are rendered with JavaScript.  In order to leverage the HTML5 Canvas, we'll need to place the canvas tag somewhere inside the HTML document, access the canvas tag with JavaScript, create a context, and then utilize the HTML5 Canvas API to draw visualizations.

When using canvas, it's important to understand the difference between the canvas element and the canvas context, as often times people get these confused.  The canvas element is the actual DOM node that's embedded in the HTML page.  The canvas context is an object with properties and methods that you can use to render graphics inside the canvas element.  The context can be 2d or webgl (3d).

Each canvas element can only have one context.  If we use the getContext() method multiple times, it will return a reference to the same context object.

Example Program:

<!DOCTYPE HTML>

<html>

<head>

<style>

body {

margin: 0px;

padding: 0px;

}

</style>

</head>

<body>

<canvas id="myCanvas" width="578" height="200"></canvas>

<script>

var canvas = document.getElementById('myCanvas');

var context = canvas.getContext('2d');

// do cool things with the context

context.font = '40pt Calibri';

context.fillStyle = 'blue';

context.fillText('Hello World!', 150, 100);

</script>

</body>

</html>

<video> and <audio> elements

HTML5 features, include native audio and video support without the need for Flash.

The HTML5 <audio> and <video> tags make it simple to add media to a website. You need to set **src** attribute to identify the media source and include a controls attribute so the user can play and pause the media.

## Embedding Video

Here is the simplest form of embedding a video file in your webpage −

<video src="foo.mp4" width="300" height="200" controls>

Your browser does not support the <video> element.

</video>

The current HTML5 draft specification does not specify which video formats browsers should support in the video tag. But most commonly used video formats are −

* **Ogg** − Ogg files with Thedora video codec and Vorbis audio codec.
* **mpeg4** − MPEG4 files with H.264 video codec and AAC audio codec.

You can use <source> tag to specify media along with media type and many other attributes. A video element allows multiple source elements and browser will use the first recognized format −

<!DOCTYPE HTML>

<html>

<body>

<video width="300" height="200" controls autoplay>

<source src="/html5/foo.ogg" type="video/ogg" />

<source src="/html5/foo.mp4" type="video/mp4" />

Your browser does not support the video element.

</video>

</body>

</html>

## Embedding Audio

HTML5 supports <audio> tag which is used to embed sound content in an HTML or XHTML document as follows.

<audio src="foo.wav" controls autoplay>

Your browser does not support the <audio> element.

</audio>

The current HTML5 draft specification does not specify which audio formats browsers should support in the audio tag. But most commonly used audio formats are **ogg, mp3** and**wav**.

You can use <source> tag to specify media along with media type and many other attributes. An audio element allows multiple source elements and browser will use the first recognized format −

<!DOCTYPE HTML>

<html>

<body>

<audio controls autoplay>

<source src="/html5/audio.ogg" type="audio/ogg" />

<source src="/html5/audio.wav" type="audio/wav" />

Your browser does not support the audio element.

</audio>

</body>

</html>

## The article Element

The HTML <article> tag is used to represent an article. More specifically, the content within the <article> tag is independent from the other content on the site (even though it could be related). By "independent" I mean that its contents could stand alone, for example in syndication.

Examples of article content could include a forum post, a newspaper article, a blog entry, or a user-submitted comment.

The <article> tag was introduced in HTML 5.

<article>

<h4>Environmentally Friendly City</h4>

<p>A <a href="http://www.natural-environment.com/blog/2008/12/14/masdar-city-the-worlds-first-zero-carbon-zero-waste-city/" target="\_blank">brand new city</a> is being built in Abu Dhabi in the United Arab Emirates which, when finished, will be the world's first zero carbon, zero waste city.</p>

<p>Masdar City, a completely self sustaining city, will be powered by renewable energy and all waste will be recycled or reused.</p>

</article>

The <article> element should contain a piece of self-contained content that could be distributed outside the context of the page. This includes things like news articles, blog posts, or user comments.

<article>

<header>

<h1>Blog Post Title</h1>

<p>Posted 13th February 2014</p>

</header>

<p>

...

</p>

</article>

You can nest <article> elements within one another. In this case it’s implied that the nested elements are related to the outer <article> element.

<article>

<header>

<h1>Blog Post Title</h1>

<p>Posted 13th February 2014</p>

</header>

<p>...</p>

<p>...</p>

<p>...</p>

<section>

<h2>Comments</h2>

<article>

<footer>

<p>Posted by: Joe Balochio</p>

</footer>

<p>This was a great article</p>

</article>

<article>

<footer>

<p>Posted by: Casey Brock</p>

</footer>

<p>How do you think this applies to the plan for world domination?</p>

</article>

</section>

</article>

In this example we’ve used <article> elements to mark up the blog post, and each of the comments. This nesting pattern implies that the comments are related to the topic of the main blog post.

## The section Element

The <section> element is used to represent a group of related content. This is similar to the purpose of an <article> element with the main difference being that the content within a <section> element doesn’t necessarily need to make sense out of the context of the page.

It’s advisable to use a heading element (<h1> – <h6>) to define the topic for the section.

Using this blog post as an example, you could have <section> elements that represent each of the individual parts within the post.

<article>

<h1>How to use HTML5 Sectioning Elements</h1>

<p>...</p>

<section>

<h2>The <main> Element</h2>

<p>...</p>

</section>

<section>

<h2>The <article> Element</h2>

<p>...</p>

</section>

<section>

<h2>The <section> Element</h2>

<p>...</p>

</section>

...

</article>

Here we’ve used an <article> element to represent the post as a whole, and then multiple <section> elements to represent each of the sub-topics discussed in the post.

If you just need to group content together for styling purposes you should use a <div> element rather than a <section>.

## The nav Element

The <nav> element is used to mark up a collection of links to external pages or sections within the current page. As well as being used for the main website navigation, the <nav> element is also a good fit for things like a table of contents, or a blogroll.

<nav>

<ul>

<li><a href="#chapter-one">Chapter One</a>

<li><a href="#chapter-two">Chapter Two</a>

<li><a href="#chapter-three">Chapter Three</a>

</ul>

</nav>

Marking up your links within a list will often make your navigation easier to use, however this is not a requirement when using a <nav> element.

## The header Element

The <header> element is used to represent the introductory content to an article or web page. This will usually contain a heading element as well as some metadata that’s relevant to the content, such as the post date of a news article for example. It could also contain a table of contents (within a <nav> element) for a longer document.

A <header> element will be associated with the nearest sectioning element, usually it’s direct parent in the page structure.

<header>

<h1>Google buys Nest</h1>

<p>Posted at 11:34am 13th January 2014</p>

</header>

In this example the <header> element contains the title and posted date for a news article.

## The footer Element

The <footer> element is used to represent information about a section such as the author, copyright information, or links to related web pages.

<footer>

Copyright Matt West 2014

</footer>

As with <header>, the <footer> element is associated with the nearest sectioning element.

## The address Element

The <address> element is one of the most commonly misunderstood HTML elements. This element is not for marking up postal address, but rather for representing the contact information for an article or web page. This could be a link to the author’s website or their email address.

<address>

Contact <a href="mailto:matt@example.com">Matt West</a>

</address>

This element is often used within the <footer> for an <article>.

<article>

<header>

<h1>Google buys Nest</h1>

<p>Posted at 11:34am 13th January 2014</p>

</header>

<p>...</p>

<p>...</p>

<footer>

<address>

By <a href="mailto:matt@example.com">Matt West</a>

</address>

<p>Copyright Matt West 2014</p>

</footer>

</article>

# **<embed> Tag**

The HTML <embed> tag is used for embedding an external application or interactive content into an HTML document.

Note that the <embed> element is an empty element (no closing tag is used).

The <embed> tag was introduced in HTML 5.

**Example:**

<p>Click and drag your mouse to look up:</p>

<embed type="video/quicktime" src="/web\_design/paris\_vegas.mov" style="width:100%;">

**<track> Tag**

The HTML 5 <track> tag is used to specify external timed text tracks for media elements (i.e. the [<video>](http://www.quackit.com/html_5/tags/html_video_tag.cfm) element and the [<audio>](http://www.quackit.com/html_5/tags/html_audio_tag.cfm) element). The text tracks specified with the <track> tag could include subtitles, captions, descriptions, chapters, and metadata.

Note that the <track> element was first introduced into HTML5 in late 2010 and therefore, browser support for this element may be limited for some time.

**Example:**

<video src="/video/pass-countdown.ogg" width="300" height="150" controls>

<track src="/video/countdown\_en.vtt" kind="subtitles" srclang="en" label="English">

<track src="/video/countdown\_es.vtt" kind="subtitles" srclang="es" label="Spanish">

<track src="/video/countdown\_it.vtt" kind="subtitles" srclang="it" label="Italian">

<p>If you are reading this, it is because your browser does not support the HTML5 video element.</p>

</video>

**<datalist> Tag**

HTML5 Datalist can be used to create a simple Autocomplete feature for a webpage.

**Example:**

<label>Enter Country name:</label>

<input type="text" list="countries" />

<datalist id="countries">

<option value="India">

<option value="United Kingdom">

<option value="United States">

<option value="Germany">

<option value="France">

<option value="Zimbabwe">

</datalist>

**<keygen> Tag**

The HTML <keygen> tag is used to process Web forms with certificate management systems. The element generates a secure key and submits the public key.

### **Example:**

<!DOCTYPE html>

<html>

<head>

<title>HTML keygen Tag</title>

</head>

<body>

<form>

<keygen name="random\_key" challenge="0987654321">

<input name="firstname" value="first name">

</form>

</body>

</html>

**<output> Tag**

The <output> element represents the result of a calculation. Typically this element defines a region that will be used to display text output from some calculation that is usually performed by a client-side script (usually JavaScript).

**Example:**

* <form oninput="result.value=parseInt(a.value)+parseInt(b.value)">
* <input type="range" id="a" value="50"> +
* <input type="number" id="b" value="100"> =
* <output name="result" for="a b"></output>
* </form>

**<aside> Tag**

The HTML <aside> tag is used to represent content that is related to the surrounding content within an article or web page, but could still stand alone in its own right. This type of content is often represented in sidebars.

An example is a "pull quote" from a longer article. A pull quote (also known as a lift-out quote or a call-out) is a quotation or edited excerpt from an article that is placed in a larger typeface on the same page, serving to lead readers into an article and to highlight a key topic.

**Example:**

<div style="font:0.8em/1.2em Arial, Helvetica, Sans-Serif;">

<aside style="font-size:larger;font-style:italic;color:green;float:right;width:35%;padding-left:5px;">

70% of the world's reefs will be destroyed over the next 40 years.

</aside>

<p>Global warming is affecting coral reefs all over the world. At the current rate, 70% of the world's reefs will be destroyed over the next 40 years.</p>

<p>As hopeless as this may sound, there are things we can do to help. By developing greener habits, we can all do our part in reducing global warming. For example, here are <a href="http://www.natural-environment.com/blog/2008/01/29/5-easy-ways-to-reduce-greenhouse-gas/" target="\_blank">5 ways to reduce greenhouse gases</a>. And here are some simple steps you can take to <a href="http://www.natural-environment.com/sustainable\_living/sustainable\_habits.php" target="\_blank">live sustainably</a>.</p>

</div>

**<bdi> Tag**

The HTML <bdi> tag is used on a span of text that is to be isolated from its surroundings for the purposes of bidirectional text formatting.

This can be useful when displaying right-to-left text (such as Arabic) inside left-to-right text (such as English) when the text-direction is unknown. The <bdi> element allows you to honor the correct directionality of text when this is unknown (such as in the case with user-generated content).

Example:

**<p>And the top five contributors are:</p>**

**<ol>**

**<li>User <bdi>David</bdi>: 1601 posts</li>**

**<li>User <bdi>Lisa</bdi>: 335 posts</li>**

**<li>User <bdi>إيان</bdi>: 195 posts</li>**

**<li>User <bdi>Sam</bdi>: 6 posts</li>**

**<li>User <bdi>Joe</bdi>: 2 posts</li>**

**</ol>**

**<command> Tag**

The <command> tag defines a command button, like a radiobutton, a checkbox, or a button.

The command element is only visible if it is inside a menu element. If not, it will not be displayed, but can be used to specify a keyboard shortcut.

**Example:**

<menu>  
<command onclick="alert('Hello World')">  
Click Me!</command>  
</menu>

**<details> and <summary> Tag**

The <details> and <summary> elements are used in combination to create a UI widget that allows the user to find out more information about a topic by clicking on a summary.

The <details> element is responsible for marking up all of the content relevant to the particular topic. The <summary> element is used to specify a short piece of text that describes the rest of the content in the <details>element.

Example:

<details>

<summary>Summary text</summary>

Content goes here...

</details>

**<dialog> Tag**

The HTML <dialog> tag indicates a part of an application that the user can interact with. Examples of dialog could include a dialog box, inspector, or window.

The <dialog> element accepts a boolean attribute called open that sets the element to "active" and allows users to interact with it. If the attribute is omitted, you will need to use a script (such as [JavaScript](http://www.quackit.com/javascript/)) to enable the dialog to open and close as required.

Example:

<div>

<dialog id="myFirstDialog" style="width:50%;background-color:#F4FFEF;border:1px dotted black;">

<p><q>The world is full of magical things patiently waiting for our wits to grow sharper.</q> - <cite>Bertrand Russell</cite></p>

<button id="hide">Close</button>

</dialog>

<!-- "Show" button -->

<button id="show">Show Dialog</button>

</div>

<!-- JavaScript to provide the "Show/Close" functionality -->

<script type="text/JavaScript">

(function() {

var dialog = document.getElementById('myFirstDialog');

document.getElementById('show').onclick = function() {

dialog.show();

};

document.getElementById('hide').onclick = function() {

dialog.close();

};

})();

</script>

**<figure> Tag**

The HTML <figure> tag is used for annotating illustrations, diagrams, photos, code listings, etc.

You can use the <figure> element to associate a caption together with some embedded content, such as a graphic or video.

You can use the <figure> element in conjunction with the [<figcaption>](http://www.quackit.com/html_5/tags/html_figcaption_tag.cfm) element to provide a caption for the contents of your <figure> element.

**Example:**

**<p><a href="#1">Figure 1</a> provides the JavaScript code for creating an alert box:</p>**

**<figure id="1">**

**<figcaption>Figure 1. JavaScript Alert Box.</figcaption>**

**<pre><code>alert('Hello!');</code></pre>**

**</figure>**

**<mark> Tag**

The HTML <mark> tag is used for indicating text as marked or highlighted for reference purposes, due to its relevance in another context.

Example:

<p>Despite the stockmarket crash in 2008, the value of my share portfolio <mark>increased by 100 percent</mark>. I must be doing something right.</p>

<p>I should point out that the value only increased because I kept pumping more money in!</p>

<meter> Tag

The <meter> element represents a scalar measurement within a known range, or a fractional value. This is also known as a gauge.

Example:

<p>Disk Usage: <meter value="0.8">80%</meter></p>

<p>Total Score: <meter value="6" min="0" max="10">6 out of 10</meter></p>

<p>Pollution Level: <meter low="60" high="80" max="100" value="85">Very High</meter></p>

<progress> Tag

The <progress> element represents the completion progress of a task.

This element normally used to indicate how much of a task has been completed, such as loading something on a page or registration process. It is typically displayed as a progress bar and often marked as a percentage from 0 to 100%.

Example:

<h2>Task Progress</h2>

<p>Progress: <progress id="bar" value="0" max="100"><span>0</span>%</progress></p>

<script type="text/javascript">

var i = 0;

var progressBar = document.getElementById("bar");

function countNumbers(){

if(i < 100){

i = i + 1;

progressBar.value = i;

// For browsers that don't support progress tag

progressBar.getElementsByTagName("span")[0].textContent = i;

}

// Wait for sometime before running this script again

setTimeout("countNumbers()", 500);

}

countNumbers();

</script>

<ruby> Tag

The HTML <ruby> tag is used for specifying Ruby annotations, which is commonly used in East Asian typography.

Ruby (also spelt *rubi*) characters are small, annotative glosses that can be placed above or to the right of a character when writing logographic languages such as Chinese or Japanese to show the pronunciation. Ruby annotations, are usually used as a pronunciation guide for relatively obscure characters.

Example:

<p lang="zh-CN">...

<ruby>

汉 <rt> hàn </rt>

字 <rt> zì  </rt>

</ruby>

...</p>

<rt> Tag

The HTML <rt> tag marks the ruby text component of a ruby annotation.

Ruby annotations are used in East Asian typography.

Ruby (also spelt *rubi*) characters are small, annotative glosses that can be placed above or to the right of a Chinese character when writing logographic languages such as Chinese or Japanese to show the pronunciation. Ruby annotations, are usually used as a pronunciation guide for relatively obscure characters.

Example:

<p lang="zh-CN">...

<ruby>

汉 <rt> hàn </rt>

字 <rt> zì  </rt>

</ruby>

...</p>

<rp> Tag

The HTML <rp> is used in ruby annotations for the benefit of browsers that don't support ruby annotations.

Ruby annotations are used in East Asian typography.

Ruby (also spelt *rubi*) characters are small, annotative glosses that can be placed above or to the right of a Chinese character when writing logographic languages such as Chinese or Japanese to show the pronunciation. Ruby annotations, are usually used as a pronunciation guide for relatively obscure characters.

Example:

<p lang="ja">...

<ruby>

漢 <rp>(</rp><rt>かん</rt><rp>)</rp>

字 <rp>(</rp><rt>じ</rt><rp>)</rp>

</ruby>

...</p>

<time> Tag

The <time> element represents a time and/or date.

This element is intended as a way to encode dates and times in a machine-readable format, so that user agents can offer to add event reminders such as birthdays and anniversaries, or any event scheduling for user's calendar.

Example:

<p>The library opens at <time>8:30 am</time> every morning.</p>

<p>The concert took place on <time datetime="2016-12-31 12:00">31 Dec</time>.</p>

<wbr> Tag

The HTML <wbr> tag is used for specifying a line break opportunity.

The <wbr> tag can be used on very long words or other long strings of text with no spaces. Without the <wbr> tag, these long strings of text could either wrap in strange place (making it difficult to read), or not wrap at all - inadvertently pushing the page layout to the side (again, making it difficult to read and view the document as intended).

The difference between the <wbr> tag and the [<br>](http://www.quackit.com/html_5/tags/html_br_tag.cfm) tag is that the [<br>](http://www.quackit.com/html_5/tags/html_br_tag.cfm) tag *forces*a line break. The <wbr> tag, on the other hand, simply represents a line break*opportunity* - the browser should only render a line-break if necessary.

Example:

<p>And the world record for the longest place name in an English-speaking country is...<br>

<i>Taumata<wbr>whakatangihanga<wbr>koauau<wbr>o<wbr>tamatea<wbr>turi<wbr>pukakapiki<wbr>maunga<wbr>horo<wbr>nuku<wbr>pokai<wbr>whenua<wbr>kitanatahu</i></p>

<p>This is the name of a hill in New Zealand.</p>

<p>Here's what it looks like without using the <code>wbr</code> tag...

<i>Taumatawhakatangihangakoauauotamateaturipukakapikimaungahoronukupokaiwhenuakitanatahu</i></p>