**The <main> element**

few important points to remember about the <main> element:

1. The top-level content of a page should be included in the <main> element.
2. The content should be exclusive and unique to it.
3. The <main> element should never be included inside the <header>, <footer>, <nav>, <aside>, or <article> elements.
4. There can only be one <main> element per page.

Consider the following example:

<body>

**<main class="main-container" role="main"> Content goes here**

**</main>**

</body>

**The <article> element**

1. Any self-contained content should be placed inside the <article> element.

*Self-contained* means that if we take the <article> element and everything inside it out into another context, all the content is self-explanatory and does not need anything else around it to be understood.

1. An <article> can be nested inside another <article> element.
2. There can be more than one <article> element in a single page.

Consider the following example:

<body>

<main class="main-container" role="main">

**<article class="article-container flex-container">**

**Content goes here**

**</article>**

</main>

</body>

**The <section> element**

The <section> element can be used to encapsulate a group of related content. This related content doesn't necessarily have to make sense if we take it out of the page's context.

1. A safe and valid way to use the <section> element is to place it inside an <article> element. You can certainly use the <article> element without a <section> element. It's recommended, although not required, to include a heading element (<h1>, <h2>, <h3>, and so on) when using the <section> element.
2. It can be confusing to know when to use the <section> element and when to use the <article> element. If you're in doubt, you can choose either element.
3. There can be more than one <section> in a single page.

<body>

<main class="main-container" role="main">

<article class="article-container flex-container">

**<section class="main-content">**

**<header>**

**<h1>The <code>&lt;main></code> element </h1>**

**</header>**

**<p>As per the MDN definition:</p> <blockquote>**

**<p>The HTML Main Element (<code>&lt;main></code>) represents&hellip;</p>**

**</blockquote>**

**</section>**

</article>

</main>

</body>

**The <aside> element**

• Content that is tangential to the main content can be included in an <aside> element. If this content was to be separated from the main content, it would still make sense on its own.

• There can be more than one <aside> in a single page.

<body>

<main class="main-container" role="main">

<article class="article-container flex-container">

<section class="main-content">

<header><h1>The <code>&lt;main></code> element </h1>

</header>

<p>As per the MDN definition:</p>

<blockquote>

<p>The HTML Main Element (<code>&lt;main></code>)

represents&hellip;</p>

</blockquote>

</section>

**<aside class="side-content" role="complementary">**

**<h2>What Does "Semantic HTML" Mean?</h2>**

**<p>Semantic markup basically means that we use HTML tags**

**to describe what a specific piece of content is.</p>**

**</aside>**

</article>

</main>

</body>

**The <header> element**

*The HTML* <header> *Element represents a group of introductory or navigational aids. It may contain some heading elements but also other elements like a logo, wrapped section's header, a search form, and so on.*

Here are a few important points to remember about the <header> element:

1. . A good rule of thumb is to use a <header> element inside a <section> element.
2. . We can wrap a heading (h1 to h6) inside a <header> element if we think it is necessary, but this is not really a common practice or required.
3. . There can be more than one <header> element in a single page.

<body>

**<header class="masthead" role="banner">**

**<div class="logo">Mastering RWD with HTML5 &amp; CSS3</div>**

**<div class="search" role="search">**

**<form>**

**<label>Search:**

**<input type="text" class="field">**

**<button>Search Now!</button>**

**</label>**

**</form>**

**</div>**

**</header>**

<main class="main-container" role="main">

<article class="article-container flex-container">

<section class="main-content">

**<header>**

**<h1>The <code>&lt;main></code> element</h1>**

**</header>**

<p>As per the MDN definition:</p>

<blockquote>

<p>The HTML Main Element (<code>&lt;main></code>) represents&hellip;</p>

</blockquote>

</section>

<aside class="side-content" role="complementary">

<h2>What Does "Semantic HTML" Mean?</h2>

<p>Semantic markup basically means that we use HTML tags to describe what a specific piece of content is.</p>

</aside>

</article>

</main>

</body>

**The <footer> element**

As per the MDN definition:

*The HTML Footer Element (*<footer>*) represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.*

Here are a few important points to remember about the <footer> element:

1. It should always contain any information about its containing parent element.
2. Although the term *footer* implies the *bottom section* of a page, article, or app, the <footer> element doesn't necessarily have to be at the bottom.
3. There can be more than one <footer> element in a single page.

Consider the following example:

<body>

<header class="masthead" role="banner">

<div class="logo">Mastering RWD with HTML5 &amp; CSS3</div>

<div class="search" role="search">

<form>

<label>Search:

<input type="text" class="field">

<button>Search Now!</button>

</label>

</form>

</div>

</header>

<main class="main-container" role="main">

<article class="article-container flex-container">

<section class="main-content">

<header>

<h1>The <code>&lt;main></code> element</h1>

</header>

<p>As per the MDN definition:</p>

<blockquote>

<p>The HTML Main Element (<code>&lt;main></code>) represents&hellip;</p>

</blockquote>

</section>

<aside class="side-content" role="complementary">

<h2>What Does "Semantic HTML" Mean?</h2>

<p>Semantic markup basically means that we use HTML tags to describe what a specific piece of content is.</p>

</aside>

</article>

**<footer class="main-footer" role="contentinfo">**

**<p>Copyright &copy;</p>**

**<ul class="nav-container" role="navigation">**

**<li><a href="#">Footer Link 1</a></li>**

**<li><a href="#">Footer Link 2</a></li>**

**<li><a href="#">Footer Link 3</a></li>**

**<li><a href="#">Footer Link 4</a></li>**

**<li><a href="#">Footer Link 5</a></li>**

**</ul>**

**</footer>**

</main>

</body>

**The <nav> element**

1. It is used to group a list or collection of links. The links can either point to external resources or to other pages within the site/app.
2. It's common practice to use an unordered list <ul> inside the <nav> element to structure the links, because it's easier to style.
3. Including a <nav> in the <header> element is also a common practice but not required.
4. Not all groups of links have to be inside a <nav> element. If we have a list of links inside a <footer> tag, then its isn't really necessary to include those links in a <nav> as well.
5. There can be more than one <nav> element in a single page, for example, a main navigation, a utility navigation, and a <footer> navigation.

Consider the following example:

<body>

<header class="masthead" role="banner">

<div class="logo">Mastering RWD with HTML5 &amp; CSS3</div>

<div class="search" role="search">

<form>

<label>Search:

<input type="text" class="field">

<button>Search Now!</button>

</label>

</form>

</div>

</header>

**<nav class="main-nav" role="navigation">**

**<ul class="nav-container">**

**<li><a href="#">Link 1</a></li>**

**<li><a href="#">Link 2</a></li>**

**<li><a href="#">Link 3</a></li>**

**<li><a href="#">Link 4</a></li>**

**</ul>**

**</nav>**

<main class="main-container" role="main">

<article class="article-container flex-container">

<section class="main-content">

<header>

<h1>The <code>&lt;main></code> element</h1>

</header>

<p>As per the MDN definition:</p>

<blockquote>

<p>The HTML Main Element (<code>&lt;main></code>) represents&hellip;</p>

</blockquote>

</section>

<aside class="side-content" role="complementary">

<h2>What Does "Semantic HTML" Mean?</h2>

<p>Semantic markup basically means that we use HTML tags to describe what a specific piece of content is.</p>

</aside>

</article>

<footer class="main-footer" role="contentinfo">

<p>Copyright &copy;</p>

<ul class="nav-container" role="navigation">

<li><a href="#">Footer Link 1</a></li>

<li><a href="#">Footer Link 2</a></li>

<li><a href="#">Footer Link 3</a></li>

<li><a href="#">Footer Link 4</a></li>

<li><a href="#">Footer Link 5</a></li>

</ul>

</footer>

</main>

</body>

**The banner role**

Here are a few important points to remember:

1. This role is usually applied to the top <header> of the page.
2. The header region contains the most prominent heading or title of a page.
3. Usually, the content that has role="banner" appears constantly across the site rather than in a single specific page.
4. Only one role="banner" is allowed per page/document.

**<header class="masthead" role="banner">**

<div class="logo">Mastering RWD with HTML5 &amp; CSS3</div>

<div class="search" role="search">

<form>

<label>Search:

<input type="text" class="field">

<button>Search Now!</button>

</label>

</form>

</div>

**</header>**

**The navigation role**

Here are a few important points to remember:

1. This role is usually applied to the <nav> element, but it can also be applied to other containers such as <div> or <ul>.
2. It describes a grou p of navigational elements/links. These links can be either to navigate the site or the page they appear on.
3. There can be more than one role="navigation" per page.

Consider the following example where the role is applied to the main <nav> element:

**<nav class="main-nav" role="navigation">**

<ul class="nav-container">

<li><a href="#">Link 1</a></li>

<li><a href="#">Link 2</a></li>

<li><a href="#">Link 3</a></li>

<li><a href="#">Link 4</a></li>

</ul>

**</nav>**

**The main role**

Here are a few important points to remember:

1. This role is usually applied to the <main> element of the page.
2. The container of the main/central subject of the page should be marked with this role.
3. Only one role="main" is allowed per page/document.
4. <body>
5. **<main class="main-container" role="main">** Content goes here
6. **</main>**
7. </body>

**For Responsive Meta Tag**

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

1. The name="viewport" directive describes the type of meta tag.
   1. The content="width=device-width, initial-scale=1" directive does several things:The width property defines the size of the viewport meta tag. We can also use specific pixel widths, for example, width=960.
   2. The device-width value is the width of the screen at 100 percent zoom in CSS pixels.
   3. The initial-scale value defines the zoom level the page should be shown at when it's first loaded. 1 equals 100 percent zoom and 1.5 equals 150 percent zoom.

**The charset meta tag**

<meta charset="utf-8">

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| Offline | script | Triggers when the document goes offline |
| onabort | script | Triggers on an abort event |
| onafterprint | script | Triggers after the document is printed |
| onbeforeonload | script | Triggers before the document loads |
| onbeforeprint | script | Triggers before the document is printed |
| onblur | script | Triggers when the window loses focus |
| oncanplay | script | Triggers when media can start play, but might has to stop for buffering |
| oncanplaythrough | script | Triggers when media can be played to the end, without stopping for buffering |
| onchange | script | Triggers when an element changes |
| onclick | script | Triggers on a mouse click |
| oncontextmenu | script | Triggers when a context menu is triggered |
| ondblclick | script | Triggers on a mouse double-click |
| ondrag | script | Triggers when an element is dragged |
| ondragend | script | Triggers at the end of a drag operation |
| ondragenter | script | Triggers when an element has been dragged to a valid drop target |
| ondragleave | script | Triggers when an element leaves a valid drop target |
| ondragover | script | Triggers when an element is being dragged over a valid drop target |
| ondragstart | script | Triggers at the start of a drag operation |
| ondrop | script | Triggers when dragged element is being dropped |
| ondurationchange | script | Triggers when the length of the media is changed |
| onemptied | script | Triggers when a media resource element suddenly becomes empty. |
| onended | script | Triggers when media has reach the end |
| onerror | script | Triggers when an error occur |
| onfocus | script | Triggers when the window gets focus |
| onformchange | script | Triggers when a form changes |
| onforminput | script | Triggers when a form gets user input |
| onhaschange | script | Triggers when the document has change |
| oninput | script | Triggers when an element gets user input |
| oninvalid | script | Triggers when an element is invalid |
| onkeydown | script | Triggers when a key is pressed |
| onkeypress | script | Triggers when a key is pressed and released |
| onkeyup | script | Triggers when a key is released |
| onload | script | Triggers when the document loads |
| onloadeddata | script | Triggers when media data is loaded |
| onloadedmetadata | script | Triggers when the duration and other media data of a media element is loaded |
| onloadstart | script | Triggers when the browser starts to load the media data |
| onmessage | script | Triggers when the message is triggered |
| onmousedown | script | Triggers when a mouse button is pressed |
| onmousemove | script | Triggers when the mouse pointer moves |
| onmouseout | script | Triggers when the mouse pointer moves out of an element |
| onmouseover | script | Triggers when the mouse pointer moves over an element |
| onmouseup | script | Triggers when a mouse button is released |
| onmousewheel | script | Triggers when the mouse wheel is being rotated |
| onoffline | script | Triggers when the document goes offline |
| onoine | script | Triggers when the document comes online |
| ononline | script | Triggers when the document comes online |
| onpagehide | script | Triggers when the window is hidden |
| onpageshow | script | Triggers when the window becomes visible |
| onpause | script | Triggers when media data is paused |
| onplay | script | Triggers when media data is going to start playing |
| onplaying | script | Triggers when media data has start playing |
| onpopstate | script | Triggers when the window's history changes |
| onprogress | script | Triggers when the browser is fetching the media data |
| onratechange | script | Triggers when the media data's playing rate has changed |
| onreadystatechange | script | Triggers when the ready-state changes |
| onredo | script | Triggers when the document performs a redo |
| onresize | script | Triggers when the window is resized |
| onscroll | script | Triggers when an element's scrollbar is being scrolled |
| onseeked | script | Triggers when a media element's seeking attribute is no longer true, and the seeking has ended |
| onseeking | script | Triggers when a media element's seeking attribute is true, and the seeking has begun |
| onselect | script | Triggers when an element is selected |
| onstalled | script | Triggers when there is an error in fetching media data |
| onstorage | script | Triggers when a document loads |
| onsubmit | script | Triggers when a form is submitted |
| onsuspend | script | Triggers when the browser has been fetching media data, but stopped before the entire media file was fetched |
| ontimeupdate | script | Triggers when media changes its playing position |
| onundo | script | Triggers when a document performs an undo |
| onunload | script | Triggers when the user leaves the document |
| onvolumechange | script | Triggers when media changes the volume, also when volume is set to "mute" |
| onwaiting | script | Triggers when media has stopped playing, but is expected to resume |

Video

<!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>

Video Attribute Specification

**Autoplay**

**Autobuffer**

**Controls**

**Height**

**Loop**

**Preload**

**Poster**

**Src**

**Width**

<!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>

**Autoplay**

**Autobuffer**

**Controls**

**Loop**

**Preload**

**Src**

## Handling Media Events

**Abort**

**Canplay**

**Ended**

**Error**

**Loadeddata**

**Loadstart**

**Pause**

**Play**

**Progress**

**Ratechange**

**Seeked**

**Seeking**

**Suspend**

**Volumechange**

**Waiting**

<!DOCTYPE HTML>

<html>

<head>

<script type = "text/javascript">

function PlayVideo() {

var v = document.getElementsByTagName("video")[0];

v.play();

}

</script>

</head>

<body>

<form>

<video width = "300" height = "200" src = "/html5/foo.mp4">

Your browser does not support the video element.

</video>

<br />

<input type = "button" onclick = "PlayVideo();" value = "Play"/>

</form>

</body>

</html>