HTML.COM
Learn HTML Code, Tags & CSS

TAGS ATTRIBUTES TUTORIALS HOSTING GUIDE BLOG ABOUT

HTML / HTML Document Structure Before And After HTML5 – Here’s What Changed

HTML Document Structure Before And After HTML5 – Here’s What Changed

If you want to write semantic markup – and believe us, you do want to write semantic markup – then you need to structure HTML documents properly. The `html`, `head`, and `body` elements have been part of the HTML specification since the mid 1990s, and up until a few years ago they were the primary elements used to give structure to HTML documents. However, the situation has changed dramatically in the last few years as HTML5 has added a slew of new tags that can be used to add rich semantic meaning to the structure of an HTML document.

Contents [hide]

HTML Document Structure Before HTML5

If you’ve been using HTML for any time at all you know that every bit of HTML needs to be wrapped in `html` tags. An opening `<html>` tag should appear first and a closing `</html>` tag should appear at the bottom of the document. Every other bit of HTML should appear between those two tags.

The `head` element is the first element to appear after the opening `html` tag. In the document `head` we place things like the page `title` and `meta` data, we add JavaScript to our page with the `script` tag, and we [`link`] to external stylesheets and other resources.

On most webpages the `head` element is a very busy place. For this reason, we’ve created a tutorial that explains the tags that typically appear in the `head` element and what these tags are used for.

Search HTML.com

Search …

SEARCH

Most Popular

<var> HTML Tag

Build Your Own Computer: 100 Tools, Tips & Resources

Input Pattern: Use It To Add E
Data Validation In HTML5

<section> HTML Tag 71

What Img Srcset Does Ir 62 vi
HTML5: A Quick & Simple
Guide

All of the content that is visible on a web page is nested between opening and closing `body` tags. The body is the primary container of the content that makes up a web page.

Up until HTML5, that was pretty much it for basic HTML document structure. All of our code was dropped in between the `body` tags and styled with CSS. However, now that HTML5 has broad support among modern browsers, it's time to implement the new HTML5 tags that will give our HTML documents a much more meaningful structure.

New Semantic Tags Added by HTML5

In this brief tutorial we'll touch on all of the new tags added as part of HTML5 to define the structure and content of a web page. The elements we're going to cover in this guide include:

- `header`
- `main`
- `nav`
- `article`
- `section`
- `aside`
- `address`
- `footer`

Using these elements isn't as complicated as it might appear at first glance, and most are fairly self-explanatory. We'll make a quick pass over each new element, and then draw up an HTML template you can use these new tags to add rich semantic meaning to your markup.

`<header>`

The `header` element is used to contain the content that appears at the top of every page of your website: the logo, tagline, search prompt, and possibly a navigational menu. In most cases, the `header` element is best positioned as a direct descendant of the `body` element, but it's also ok to place it inside the `main` element if you prefer.

`<main>`

Use the `main` element between `header` and `footer` elements to contain the primary content of your web page. The `main` element cannot be a descendant of an `article`, `aside`, `header`, `footer`, or `nav` element. Instead, it should be a direct descendant of the `body` element. Think of it as the direct replacement for the `div id="main"` you've used in the past to wrap up your entire page contents.

It's also ok to use more than one `main` element on a webpage. For example, if your blog homepage includes your five most recent posts, it would be appropriate to wrap each post in it's own `main` element – or you could wrap each in `article` tags.

`<nav>`

Navigational menus are commonly placed at the top of a web page, in a sidebar, or in the page footer. Wherever you happen to place a navigational menu, wrap it in `nav` tags. Note that you don't need to use `nav` tags for every link, just for blocks of links that provide either sitewide navigation or navigation for a specific part of a website.

`<article>`

If your website includes blog posts, articles, or any other content that could just as well appear on another website as syndicated content, wrap that content in an `article` post. You can use an `article` element just about anywhere other than nested within an `address` element, but in most cases an `article` element will be a direct descendant of a `main` element or of a `section` element that is a direct descendant of a `main` element.

<section>

The `section` element is used to identify content that is a major sub-section of a larger whole. For example, if you've posted a long-form ebook in HTML format, it would be reasonable to wrap each chapter in a `section` element. Likewise, if you have a sidebar (semantically wrapped in `aside` tags) that contains four sections – ads, a search prompt, related posts, and a newsletter signup form – it would be ok to wrap each of these four sections in `section` tags since a written outline of the sidebar contents would include a line item for each of the four sections.

There is some confusion about when to use a `section` and when to use a `div`. Here's a good rule of thumb to help you know when to use each:

- Use a `div` if you're wrapping up some content purely to make it easier to style the content or to make it easier for some JavaScript to get ahold of it.
- Use a `section` if you would list the content as an item when writing out an outline of the document.

<aside>

If your website contains information that isn't directly related to the main content of the page, it would be appropriate to wrap that information in `aside` tags. For example, if you write a post that includes some technical terms, and you add definitions for those terms in a sidebar, it would make sense to wrap those definitions in `aside` tags. It is also common for the entire sidebar of a blog-type website to be wrapped in `aside` tags to make it clear that the sidebar is not part of the primary content of the page.

<address>

The `address` element provides contact information for the nearest parent `article` or `body` element that contains it. Use the `address` element inside an `article` to provide contact information for the article's author. Use it outside of an `article` in the `main` or `footer` elements, or as a direct descendant of the `body` element, to provide contact information for the website's owner.

<footer>

The `footer` appears at the bottom of a section of a document. Typically, the `footer` is a direct descendant of the `body` element, but it can also be used within a `main` element, a `section`, or an `article`. The most common use of the `footer` element is to place it at the bottom of an HTML document to contain things like a copyright notice, links to related content, `address` information about the owner of the website, and links to administrative things like privacy policies and website's terms of service.

You may also use the `footer` element within an `article` to provide metadata about that particular article. For example, if `article` tags have been used to wrap a forum post, it would be appropriate to wrap copyright information and the date and time the post was made in a `footer` element and place it at the bottom of the `article`.

An HTML Document Template

The template below will show you how all of these elements are properly nested together. We invite you to copy it and use it as a boilerplate template for all of your HTML documents.

```
<html>
  <!--Only the head and body elements are supposed to be direct descendants of the
  html element. All others should be descendants of either the head or body-->
  <head>
    <!--The head element must be a direct descendant of the html element-->
    <!--The head element is a very busy place for most websites, so we've created
    a tutorial to walk you through the different elements and tasks accomplished
    in the head element. You can find it at the following address:
    https://html.com/document/metadata/ -->
    <title>Your Webpage Title Goes Here</title>
  </head>
  <body>
    <!--The body element contains the full visible content of the web page-->
    <header>
      <!--The header typically includes your logo, tagline, and may contain a nav
      element-->
      <nav>
        <!--The nav element isn't used for every single link but for navigational
        menus-->
      </nav>
    </header>
    <main>
      <!--The main element cannot be used inside of anything other than the body
      element. It is intended to hold the main content of the page.-->
      <nav>
        <!--You can use a nav element just about anywhere-->
      </nav>
      <article>
        <!--If your web page contains a blog post or news article it makes sense
        to wrap the whole article in article tags-->
        <aside>
          <!--The aside tag can be used within an article or outside of it. It
          is used to mark content that is related but not central to the main
          content of the page-->
        </aside>
        <section>
          <!--Sections are used to seperate major parts of an element, such as
          chapters of an HTML ebook, or to cordone off the comments section
          from the rest of the main element-->
        </section>
        <address>
          <!--An address element inside of an article element is used to provide
          contact info for the author of the article-->
        </address>
      </article>
      <aside>
        <!--The aside element would also be used to mark a sidebar if used
        outside of the main element-->
        <section>
          <!--Within a sidebar you could use section elements to identify the
          different parts of the sidebar. For example, you could put adds in
          one section, related posts in a second section, and a newsletter
          signup form in a third section element.-->
        </section>
      </aside>
    </main>
    <footer>
      <!--The footer typically contains links to things like About Us, Privacy
      Policy, Contact Us and so forth. It may also contain a nav, address,
      section, or aside element.-->
      <address>
        <!--Put an address element in the footer and you're indicating that
        the contact info within the element is for the owner of the website
        rather than the author of the article.-->
      </address>
    </footer>
  </body>
</html>
```

Jon Penland

Jon is a freelance writer, travel enthusiast, husband and father. He writes about web technologies such as WordPress, HTML, and CSS.

Related Elements

Element Name	Attributes	Notes
main		The <main> element is used to denote the content of a webpage that relates to the central topic of that page or application. It should include content that is unique to that page and should not include content that is duplicated across multiple webpages, such as headers, footers, and primary navigation elements.
header		The <header> element is used to identify content that precedes the primary content of the web page and often contains website branding, navigation elements, search forms, and similar content that is duplicated across all or most pages of a website.
footer		The <footer> element is a structural element used to identify the footer of a page, document, article, or section. A <footer> typically contains copyright and authorship information or navigational elements pertaining to the contents of the parent element.
aside		The <aside> element is used to identify content that is related to the primary content of the webpage, but does not constitute the primary content of the page. Author information, related links, related content, and advertisements are exampes of content that may be found in an aside element.
article		The <article> element identifies a self-contained piece of content which

		could theoretically be distributed to other websites and platforms as a stand-alone unit. The <article> element is a good choice to contain entire blog posts, news articles, and similar content.
title		The <title> element is a required HTML element used to assign a title to an HTML document. Page titles are not displayed in the browser window, but they are used as the page name by search engines and displayed by browsers in the title bar, on the page tab, and as the page name of bookmarked webpages.
isindex	action prompt	The <isindex> element was used to create a single line search prompt for querying the contents of the document. Implementation of the element was inconsistent and the functionality duplicated by the <form> and <input> elements. As a result, <isindex> was deprecated in HTML 4.01.
meta	content http-equiv <meta name="">	The <meta> element is used to add machine-readable information to an HTML document. Information added with the <meta> tag is not displayed to website visitors but is provided for use by browsers and web crawlers.
html comment		This element is used to add a comment to an HTML document. An HTML comment begins with <code><!—</code> and the comment closes with <code>—></code>. HTML comments are visible to anyone that views the page source code, but are not rendered when the HTML document is rendered by a browser.
DOCTYPE		The <!DOCTYPE html> declaration is used to inform a website visitor's browser that the document being rendered is an HTML document. While not actually an HTML

		element itself, every HTML document should being with a DOCTYPE declaration to be compliant with HTML standards.
base	target href	The <base> element is used to identify a base URL upon which to build all relative URLs that appear on a webpage. In addition, if the <base> element has a target attribute, the target attribute will be used as the default attribute for all hyperlinks appearing in the document.
body	background bgcolor bgproperties stylesrc text scroll topmargin onUnload onLoad onFocus	The <body> element contains the entire content of a webpage. It must be the second element inside of the parent <html> element, following only the <head> element.
html		The <html> element is used as a container for all of the HTML of an entire document.
head		The <head> element contains information about an HTML document that is used by browsers and web crawlers but is not displayed to website visitors.
div	align	The <div> element defines an arbitrary block of content which can be placed and styled as a single unit.

Tutorials and Resources

What Is Metadata In HTML Documents?: Head Elements Explained