HTML.COM
Learn HTML Code, Tags & CSS

TAGS ▾ ATTRIBUTES TUTORIALS ▾ HOSTING GUIDE ▾ BLOG ▾ ABOUT ▾

HTML / HTML Anchors & Links: The Right Way to Use them In Your HTML Code

HTML Anchors & Links: The Right Way to Use them In Your HTML Code

The anchor element is used to create hyperlinks between a *source anchor* and a *destination anchor*. The source is the text, image, or button that links to another resource and the destination is the resource that the source anchor links to.

Hyperlinks are one of the fundamental technologies that make the web the information superhighway, and understanding how to use anchor elements is one of the first things you need to master when learning HTML.

Contents [hide]

The Anchor Element

The anchor element tag is the letter "a" surrounded by angle brackets like this: `<a>`. Both the opening and closing attributes are required, and all of the content between the tags makes up the anchor source.

```
If we want to use just a single word as an anchor, we
wrap just that one <a>word</a> in anchor tags.



We could also wrap an entire paragraph in anchor
```

Search HTML.com

Search …

SEARCH

Most Popular

How To Share Your Pictures Online
100+ Image Hosting Tools

HTML Comments: How To Use The
In Your HTML Code

How To Use In Your HTML Code

How To Use The <a> To Make Links & Open Them Where You Want! 94

<var> HTML Tag 937

Recommended Reading

```
<!--We can also use an image as an anchor element-->
<a><img src="http://example.com/image-file.png" /></a>
```

While the examples above will produce anchor elements they aren't of much use since we haven't included any additional instructions. Right now these anchor elements link to nothing. To link a source anchor to a destination anchor, we need to apply some additional attributes to the anchor element.

In this short tutorial we'll cover the attributes you can use to add a destination anchor to your hyperlinks, tell the browser what to do with the link, and add semantic meaning to anchor elements for browsers and web crawlers to use.

The Most Important Anchor Attributes

There are three anchor attributes you need to know to create functional hyperlinks. These attributes are `href` , `target` , and `download` .

Specify a Hyperlink Target: `href`

The hypertext reference, or `href` , attribute is used to specify a target or destination for the anchor element. It is most commonly used to define a URL where the anchor element should link to.

```
In this example, the <a href="http://example.com">anchored text</a>
links to the URL <em>www.example.com</em>.
```

An `href` can do a lot more than just link to another website.

- It can be used to link directly to any element on a web page that has been assigned an `id` .
- It can be used to link to a resource using a protocol other than http.
- It can be used to run a script.

The following HTML includes examples of all three of these actions:

```
<a href="#Specify_a_Hyperlink_Target_href">This first anchor element</a> link
s back to the heading
of this section of the tutorial by linking to an <code>id</code> we
assigned to the section heading element.

We can use the <code>mailto</code> protocol to create a link that will
try to launch the default email program on your computer. Let's give it
a try! We'd love for you to <a href="mailto:contact@html.com">
get in touch</a> with us!

Lastly, if you click the link below a complimentary JavaScript alert
window will appear. <a>
Click the link to see what happens</a>.
```

In the box below you'll see that HTML as it renders in the browser. Try each link to see what happens.

This first anchor element links back to the heading of this section of the tutorial by linking to an `id` we assigned to that heading element.

Lastly, if you click the link below a complimentary JavaScript alert window will appear.
Click the link to see what happens.

The `href` element is pretty easy to use, but we can get a lot more mileage out of our anchor elements by getting to know the `target` attribute.

Specify a Location to Open the Link: `target`

Let's look back at the `mailto` link we created in the previous example. Here's that code again:

```
We'd love for you to <a href="mailto:contact@html.com">
get in touch</a> with us!
```

While that link is useful, the way it opens isn't ideal. Depending on how your computer and browser are configured, it's possible that clicking on the link created by that text will redirect your browser window away from this website to a web-based email application such as Gmail or Yahoo! Mail. While we certainly want to encourage our visitors to contact us, we don't want to do so at the expense of sending visitors away from our website.

The answer to this conundrum is the `target` attribute paired with the `_blank` value.

```
We'd love for you to <a href="mailto:contact@html.com" target="_blank" rel="n
oopener">get in touch</a> with us!
```

When we add the `target` attribute and `_blank` attribute to our link we tell the visitors browser to open the link in a new (blank) browser tab or window. If we render that code in the browser, when a visitor clicks the link it will open in a new tab.

We'd love for you to get in touch with us!

Specify a Resource to Download: `download`

Links are also used to tell a browser to start downloading a file. The `download` attribute is used to identify a link that should initiate a download and the value assigned to the `download` attribute is the name of the file to be downloaded.

The `href` attribute also comes into play when setting up an anchor element that initiates a download. While the `download` attribute names the file, the `href` attribute points to the location where the file is hosted.

Here's an example of how a download-initiating anchor element would look:

```
provide a name for the downloaded file.
```

If the `href` in the example above pointed to a real file, when the download was complete the downloaded file would be called *Example_File.ext* where *.ext* would be the format of the file that was downloaded. It isn't necessary to include the file type extension in the value assigned to the `download` attribute. The file extension will be automatically identified when the browser downloads the file, and automatically appended to the downloaded file name.

Internal and External Links

There are two types of resources we can link to using the `a` element: internal and external. Internal links are those that point to other pages of our website. External links point to web pages that aren't part of our website.

Building internal links is important for a few different reasons:

- Internal links are used to create navigation menus that help website visitors navigate our website.
- Internal links are used in the text of website content to help website visitors locate related content.
- Internal links are also used by search engine web crawlers to locate the pages of a website and to share authority (also known as link juice) with the other pages of a website.

When writing internal links make sure you don't overdo it. The link juice of any given web page is shared between the links on the page. So the more links you put on a page, the more diluted the juice passed to each link becomes. A good rule of thumb is to have no more than 100 links per page although there are exceptions in the case of extremely large and complex websites.

External links are also important for a few different reasons:

- External links may be recommended, required, or just best-practice to provide proper attribution to the source of an idea or a resource.
- External links allow us to refer website visitors to useful related content.
- When other websites post external links that point at our website, these links known as backlinks allow link juice to flow to our website and improve our website's position on search engine results pages (SERP).

When writing external links try to avoid linking to direct competitors. You don't want to help their SERP ranking for search terms that you are targeting for your own website. It's also a good idea to use the `target="_blank"` attribute when writing external links. By opening external links in a new tab you keep visitors on your site for longer.

Never Say “Click Here”

Surely you've seen and maybe even created links that look something like this:

```
To learn more <a href="http://example.com">click here</a>!
```

There are at least three good reasons why using generic anchor element text such as “Click Here” is a terrible practice.

- Website visitors who depend on assistive technologies such as screen readers will have a hard

- Website visitors who are scanning a page will have to take several additional seconds to investigate the text around the link to have an idea of what the anchor links to.
- Search engine web crawlers associate anchor element text with the link URL. Properly selected anchor element text helps search engines determine the relevance of a web page to specific keywords. Generic text tells search engines nothing about the linked web page.

Ideal anchor element text is succinct and identifies a keyword or keywords that are relevant to the web page. Here's a sentence that includes two examples of properly selected link text:

```
If you want your website to get more traffic from
<a href="http://www.google.com/" target="_blank" rel="noopener">search engine
s</a>,
check out our <a href="/seo/">SEO tutorial</a> for tips on improving
your website's SERP ranking.
```

The first link tells web crawlers and website visitors that the website `http://www.google.com` has something to do with the keywords "search engines". The second tells visitors and search engines that the page found at the relative URL `/seo/` is related to the keywords "SEO tutorial".

Absolute vs. Relative URLs

There were two links in our last example: `http://www.google.com` and `/seo/`. The first is an example of an *absolute URL*. Absolute URLs are those that include a complete (absolute) description of the link destination. Absolute URLs include the protocol (*http*) and the complete domain name and file path needed to reach the destination anchor.

The second is an example of a *relative URL*. Relative URLs link to a web page by describing the position of the page *relative* to the position of the current page. When writing internal links that point to other pages of the same website we have the option of writing relative URLs rather than absolute URLs.

For example, if we are at this address: `http://example.com/products/red-product/`, and we want to link to the blue product page, we can use either of the following anchor elements:

```
<a href="/products/blue-product/">Go check out the blue product!</a>
<a href="../blue-product/">Go check out the blue product!</a>
```

What the first syntax says is: "look in the lowest level of the file directory for the file products, and look for blue products within that file". What the second syntax says is: "look in the parent folder of the `red-product` directory for a directory with the name `blue-product`. Both of these anchor elements point to the URL: `http://example.com/products/blue-product/`.

Let's say that we want to link to *About Us* – a page that exists as a direct descendant of the homepage. Here's how we could do that:

```
<a href="/about-us/">Learn more about us.</a>
```

That syntax says "go back to the lowest level of the file directory tree and then look for a directory titled *about-us*".

As a final example, let's say we are at this address once again: `http://example.com/products/red-product/`, and that we want to link to a subcategory page that exists within the `/red-products/` category.

What that relative URL will do is tell the browser to look in the current folder for a file named `large.php` and will resolve to this URL: `http://example.com/products/red-product/large.php`.

To summarize, when we write relative URLs we have three options:

- When we start a relative URL with "/" we tell the browser to look for the specified file or folder in the lowest level of the file directory.
- If we start a relative URL with "../" we tell the browser back up in the directory tree one level.
- If we start a relative URL with a file or folder name we tell the browser to look for the specified resources in the current file.

Why Are Relative URLs Used?

Relative URLs are very popular with web developers. It is common for websites to be developed on a staging server with a domain name that is not the same as the eventual permanent domain name. If relative URLs are used, when the site is transferred from the staging server to being live on the web all of the relative URLs will continue to work just fine. However, if absolute URLs are used the development team will have to go through the site fixing all of the URLs.

While the benefit of using relative URLs during the development of a site on a staging server not inconsiderable, it comes at a cost.

While you should know how relative URLs work, many experienced developers and SEO experts recommend sticking to absolute URLs as much as possible.

Other Useful Anchor Attributes

With just the three attributes we've covered so far, you can complete every hyperlinking task you will encounter on the web. However, there are additional attributes that can be used to tell a visitor's web browser and search engines that index our website something about the meaning of the hyperlinks.

Specify the Language of the Anchor Destination: `hreflang`

This is optional tag can be used to tell a web browser the language of the document to which the anchor is linked. For example, if we were linking to a URL and wanted to tell the browser that the destination anchor was in Spanish we would use the appropriate language code in ISO 639-1 format as the value of the `hreflang` attribute.

```
To tell the browser that a <a href="http://example.com" hreflang="es">a link
</a> points to a resource that is in a
different langauge, we can use the <code>hreflang</code>
attribute.
```

Specify the Relationship Between Source and Destination: `rel`

The relationship, or `rel`, attribute is used to describe the relationship between the source anchor and the destination anchor. For example, we could use the following code to identify the destination as being a resource about the author of the document.

There are several different values that can be used to describe the relationship between the source link and the destination link. Here are a few of the most common:

- `rel="nofollow"` : Use this attribute if you're linking to a resource you don't want to lend your site's reputation to. For example, if you link to a spammy site as an example of what *not* to do, you would want to use this value to keep from contributing to the site's search engine visibility.
- `rel="alternate"` : If your site has more than one version – such as a translation into a different language – use this value to identify the linked resource as an alternate version of the current page.
- `rel="bookmark"` : This value is used to identify a URL formulation that is permanent and may be used for bookmarking.
- `rel="help"` : Identify a linked resource as a help file for the current page with this value.
- `rel="license"` : Use this attribute when linking to a recognized copyright license.
- `rel="next"` : If the current document exists in a series of documents, use this value when linking to the next document in the series.
- `rel="prev"` : If the current document exists in a series of documents, use this value when linking to the previous document in the series.
- `rel="noreferrer"` : If you want to link to an external website but avoid letting the destination website know who the referrer is, use this value.

Specify the Internet Media Type: `type`

The `type` attribute is an optional attribute that can be used to identify the Internet media type, or MIME type, of the destination document. For example, if linking to a website rendered in HTML you could add the attribute `type="text/html"` to tell a visitor's browser that the link points to an HTML document.

Right now this attribute doesn't do much of anything, but the thinking is that in the future the information contained in this attribute could be used to somehow communicate to a website visitor the type of content they are about to be linked to before clicking on the link.

The `type` value must be a valid IANA media type. Some of the most commonly used values are: `text/html`, `text/css`, `application/javascript`, and `multipart/form-data`.

A Web of Links

The World Wide Web is best described as a massive library of hyperlinked documents where anchor elements are used to create bridges between related documents. In this definition, anchor elements occupy their rightful place as the glue that ties the web together and the bridges that allow web users to move from one document to a related document.

One common way for links to be used is to build a navigation menu that lays out the logical, hierarchical structure of a website. If you want to learn how to build effective navigational menus read our tutorial on that topic.

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
link	href rel media title type	The <link> element is used to define a relationship between an HTML document and an external resource. This element is most commonly used to define the relationship between a document and one or more external CSS stylesheets.
anchor	hreflang download target title href name	The <a> element, or anchor element, it used to create a hyperlink to another webpage or another location within the same webpage. The hyperlink created by an anchor element is applied to the text, image, or other HTML content nested between the opening and closing <a> tags.
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.

Tutorials and Resources

How to Create Responsive Menus with CSS