Chapter 6: Anchors and Hyperlinks

Parameter Details

href	Specifies the destination address. It can be an absolute or relative URL, or the name of an anchor. An absolute URL is the complete URL of a website like http://example.com/. A relative URL points to another directory and/or document inside the same website, e.g. /about-us/ points to the directory "about-us" inside the root directory (/). When pointing to another directory without explicitly specifying the document, web servers typically return the document "index.html" inside that directory.
hreflang	Specifies the language of the resource linked by the href attribute (which must be present with this one). Use language values from BCP 47 for HTML5 and RFC 1766 for HTML 4.
rel	Specifies the relationship between the current document and the linked document. For HTML5, the values must be defined in the specification or registered in the Microformats wiki.
target	Specifies where to open the link, e.g. in a new tab or window. Possible values are _blank, _self, _parent, _top, and framename (deprecated). Forcing such behaviour is not recommended since it violates the control of the user over a website.
title	Specifies extra information about a link. The information is most often shown as a tooltip text when the cursor moves over the link. This attribute is not restricted to links, it can be used on almost all HTML tags.
download	Specifies that the target will be downloaded when a user clicks on the hyperlink. The value of the attribute will be the name of the downloaded file. There are no restrictions on allowed values, and the browser will automatically detect the correct file extension and add it to the file (.img, .pdf, etc.). If the value is omitted, the original filename is used.

Anchor tags are commonly used to link separate webpages, but they can also be used to link between different places in a single document, often within table of contents or even launch external applications. This topic explains the implementation and application of HTML anchor tags in various roles.

Section 6.1: Link to another site

This is the basic use of the `<a>` (anchor element) element:

```
<a href="http://example.com/">Link to example.com</a>
```

It creates a hyperlink, to the URL http://example.com/ as specified by the href (hypertext reference) attribute, with the anchor text "Link to example.com". It would look something like the following:

> Link to example.com

To denote that this link leads to an external website, you can use the `external` link type:

```
<a href="http://example.com/" rel="external">example site</a>
```

You can link to a site that uses a protocol other than HTTP. For example, to link to an FTP site, you can do,

```
<a href="ftp://example.com/">This could be a link to a FTP site</a>
```

In this case, the difference is that this anchor tag is requesting that the user's browser connect to `example.com` using the File Transfer Protocol (FTP) rather than the Hypertext Transfer Protocol (HTTP).

This could be a link to a FTP site

Section 6.2: Link to an anchor

Anchors can be used to jump to specific tags on an HTML page. The `<a>` tag can point to any element that has an id attribute. To learn more about IDs, visit the documentation about Classes and IDs. Anchors are mostly used to jump to a subsection of a page and are used in conjunction with header tags.

Suppose you've created a page (page1.html) on many topics:

```
<h2>First topic</h2>
<p>Content about the first topic</p>
<h2>Second topic</h2>
<p>Content about the second topic</p>
```

Once you have several sections, you may want to create a Table of Contents at the top of the page with quick-links (or bookmarks) to specific sections.

If you gave an id attribute to your topics, you could then link to them

```
<h2 id="Topic1">First topic</h2>
<p>Content about the first topic</p>
<h2 id="Topic2">Second topic</h2>
<p>Content about the second topic</p>
```

Now you can use the anchor in your table of contents:

```
<h1>Table of Contents</h1>
 <a href='#Topic1'>Click to jump to the First Topic</a>
 <a href='#Topic2'>Click to jump to the Second Topic</a>
```

These anchors are also attached to the web page they're on (`page1.html`). So you can link across the site from one page to the other by referencing the page *and* anchor name.

```
 Remember, you can always <a href="page1.html#Topic1">look back in the First Topic</a>
for supporting information.
```

Section 6.3: Link to a page on the same site

You can use a relative path to link to pages on the same website.

```
<a href="/example">Text Here</a>
```

The above example would go to the file `example` at the root directory (`/`) of the server.

If this link was on http://example.com, the following two links would bring the user to the same location

```
<a href="/page">Text Here</a>
<a href="http://example.com/page">Text Here</a>
```

Both of the above would go to the `page` file at the root directory of `example.com`.

Section 6.4: Link that dials a number

If the value of the href-attribute begins with tel:, your device will dial the number when you click it. This works on mobile devices or on computers/tablets running software – like Skype or FaceTime – that can make phone calls.

```
<a href="tel:11234567890">Call us</a>
```

Most devices and programs will prompt the user in some way to confirm the number they are about to dial.

Section 6.5: Open link in new tab/window

```
<a href="example.com" target="_blank">Text Here</a>
```

The `target` attribute specifies where to open the link. By setting it to `_blank`, you tell the browser to open it in a new tab or window (per user preference).

> **SECURITY VULNERABILITY WARNING!**
>
> Using `target="_blank"` gives the opening site partial access to the `window.opener` object via JavaScript, which allows that page to then access and change the `window.opener.location` of *your* page and potentially redirect users to malware or phishing sites.
>
> Whenever using this for pages you do not control, add `rel="noopener"` to your link to prevent the window.opener object from being sent with the request.
>
> Currently, Firefox does not support noopener, so you will need to use `rel="noopener noreferrer"` for maximum effect.

Section 6.6: Link that runs JavaScript

Simply use the `javascript:` protocol to run the text as JavaScript instead of opening it as a normal link:

```
<a href="javascript:myFunction();">Run Code</a>
```

You can also achieve the same thing using the `onclick` attribute:

```
<a href="#" onclick="myFunction(); return false;">Run Code</a>
```

The `return false;` is necessary to prevent your page from scrolling to the top when the link to # is clicked. Make sure to include all code you'd like to run before it, as returning will stop execution of further code.

Also noteworthy, you can include an exclamation mark `!` after the hashtag in order to prevent the page from scrolling to the top. This works because any invalid slug will cause the link to not scroll *anywhere* on the page, because it couldn't locate the element it references (an element with `id="!"`). You could also just use any invalid slug (such as #scrollsNowhere) to achieve the same effect. In this case, `return false;` is not required:

```
<a href="#!" onclick="myFunction();">Run Code</a>
```

> **Should you be using any of this?**
>
> The answer is almost certainly ***no***. Running JavaScript inline with the element like this is fairly bad practice. Consider using pure JavaScript solutions that look for the element in the page and bind a function to it instead. Listening for an event
>
> Also consider whether this element is really a *button* instead of a *link*. If so, you should use `<button>`.

Section 6.7: Link that runs email client

Basic usage

If the value of the `href`-attribute begins with `mailto:` it will try to open an email client on click:

```
<a href="mailto:example@example.com">Send email</a>
```

This will put the email address `example@example.com` as the recipient for the newly created email.

Cc and Bcc

You can also add addresses for cc- or bcc-recipients using the following syntax:

```
<a href="mailto:example@example.com?cc=john@example.com&bcc=jane@example.com">Send email</a>
```

Subject and body text

You can populate the subject and body for the new email as well:

```
<a href="mailto:example@example.com?subject=Example+subject&body=Message+text">Send email</a>
```

Those values must be URL encoded.

Clicking on a link with mailto: will try to open the default email client specified by your operating system or it will ask you to choose what client you want to use. Not all options specified after the recipient's address are supported in all email clients.