



**University of Babylon**

**College of Information Technology**

**Department of Information Networks**

# DYNAMIC HTML

Lecturer

*Haider M. Habeeb*

Second Year, First Course

2012 - 2013

## Understand Hypertext and Links

Why did the arrival of hypertext links make such a difference? You can understand it better by comparing the Internet to a vast library. What if you went into a library and discovered that the only way you could find anything was by going through multiple lists of items, each one becoming a little more specific, until you narrowed your search down to the information you were looking for? Then, when you found the book you wanted, you had to access the information in it, page by page, until you found what you needed? Thankfully, libraries don't work that way.

The development of HTML and the creation of the World Wide Web made it possible to go right to the document you want, and a lot more. Because pages are linked together, it is possible to go to the specific page you want. Even better, assuming a link is provided, it is possible to go to a precise location on the page you are looking for. And with all those pages linked together around the world...well, you see where the term "web" comes from. And it's all possible because of hypertext.

## Understand URLs

What is hypertext? Even if you've never been on the Web, you've probably used hypertext before. Often the "help" files of computer programs use hypertext to help you find the answer to your question quickly. Computerized encyclopedias are another place you've probably encountered hypertext. The articles in such encyclopedias and dictionaries generally have related subjects listed at the bottom.

If you click on the subject, you are instantly transported to another article for more information. That's hypertext at work. To understand how to use it on the Web, you must learn a little about uniform resource locators (URLs). URLs are more commonly known as Web addresses, Internet addresses.

A URL is made up of several parts, each of which helps identify a specific location where a resource can be found (or sent). For instance, if you want to visit University of Babylon Web site, the address would be <http://www.uobabylon.edu.iq/>. The first part of the address is the protocol, usually <http://>. This stands for Hypertext Transfer Protocol, the specification for transferring documents using hypertext links or hyperlinks. As you'll see in the following, there are other protocols, depending on what you are trying to do on the Net; but the primary one for the Web is <http://>. When an Internet server sees <http://> at the beginning of an address, it knows to expect an HTML document.

The next portion of a URL identifies the host (also called a server). In the case of university of Babylon address, the host portion is [www.uobabylon.edu.iq](http://www.uobabylon.edu.iq/). The host is the system where the file or document you are looking for can be found, or where it is to be sent.

You probably are more familiar with the term domain name, as in [www.domainname.com](http://www.domainname.com). When you type in a domain name, it is translated into what is known as an IP address, which numerically describes the exact location of the server that hosts that particular Web site.

The last part of a URL locates the file on the server. You generally don't have to type in this part, as the browser supplies it for you. If it is displayed, the file location might look something like this:

**<http://www.mysite.com/index.htm>**

No directory is listed in this address because the **index** page usually is located in a site's main, or root, directory. If you are accessing a page that is in a different directory on a Web site, the address might display this way:

**<http://www.mysite.com/vacations/pictures.htm>**

In the case of this address, the file is pictures.htm; it resides in the vacations directory.

### **Link to another Web Site with the Anchor Element, <a> </a>**

Hypertext points to a precise location on the Internet by using a URL to identify where information is and what should be done with it. But how do you create the link? That's where HTML comes in. HTML stands for Hypertext Markup Language. You can create hypertext links or hyperlinks using an element specially designed for that purpose: the anchor element, <a> </a>.

A link is simply a connection between one hypertext document and another, created by enclosing text or an image inside two anchor tags. However, the link must point somewhere, so you also must include the href=" " attribute. Href stands for hypertext reference and is the place where you put the URL or pointer, so your link has a destination. A link that doesn't point anywhere will look like this:

**<a href=" ">This link doesn't go anywhere</a>.**

If you type this link into a Web page, it will appear as the default blue underlined text that characterizes links, but it won't take you anywhere because it doesn't have an address. To give the link somewhere to take you, insert a URL between the quotation marks beside the href attribute. To make the link take a visitor to Babylon University's site, change the link to read:

**<a href="http:// www.uobabylon.edu.iq ">Go to University of Babylon.</a>**

That's all there is to creating a simple link to another Web site.

### **Navigate Your Site with Internal Links**

One of the most common uses for hyperlinks on a Web site is for site navigation. Site navigation is just a fancy term for "helping people find their way around." It's good to keep that in mind as you develop your site. As you develop links to other pages on your Web site, remember that you want to make things as easy for your visitors as possible. The larger and more complex your site becomes, the more links you will find yourself using. If your links are complicated, confusing, or non-functional, visitors will become frustrated and probably will leave.

### **Link to Pages on Your Own Site**

Linking to another page on your own site is even simpler than linking to a different Web site; all you need to include is the name of the file you're linking to and the directory where it is located. If the file happens to be in the same directory, you need only the filename.

### **Link to another Page in the Same Directory**

To link to another page in the same directory, simply include the filename in the href=" " attribute of the opening anchor tag. For example, to add a link to your index.htm page that will open up headings.htm, type

**<a href="headings.htm">Headings Page</a>.**

This will create a link that looks like this:

[Headings Page](#)

This kind of link will work as long as the file is stored in the same directory. However, if you have a large site, you might find it easier to organize your files in different directories, much as you do on your computer. If you do this, you will have to provide the path to your file.

### **Link to another Page in a Different Directory**

You have two options when linking to a file in another directory: absolute and relative URLs.

**Use Absolute URLs for Simplicity** The least complicated option is to use an absolute URL. Although that term might sound intimidating, it simply refers to the exact “path” to the file’s location, including all directories and subdirectories you have to go through to get there. It is called absolute because it specifies a precise address.

**<a href="/biographies/lecturers/haider.htm">Haider M. Habeeb</a>**

This tells the browser to go to the biographies directory, then to go to the lecturers directory, and finally to load the haider.htm file.

**Use Relative URLs for Flexibility** If you want your site to have maximum flexibility and you definitely don’t want to waste your time rewriting links, relative URLs are your solution. Whereas an absolute URL contains the exact address of the file you want the browser to load, a relative URL uses a kind of shorthand to tell the browser to go backward one or more directories. If you want to write the link from washington.htm-to-index.htm using a relative URL, it will look like this:

**<a href=" ../index.htm">Home</a>**

The two dots followed by a slash are the code that instructs the browser to move backward (or up) one directory level. Because you must go up two levels to get to the main (or root) directory and index.htm, you must use the code twice.

What makes relative URLs advantageous is that the link will work even if you move the file to another directory. What if you want to link to a file in a parallel directory; that is, go sideways rather than backward? On our famous persons’ Web site, to link to Mozart’s biography, you’d send the browser to the Musicians directory. You would write a link to tell the browser to go up one directory (to Biographies), to go forward one directory (to Musicians), and to load the file mozart.htm. How would it look?

**<a href=" ../Musicians/mozart.htm">Mozart Biography</a>**

**Confusing?** It can be if you let your Web site structure get complicated. A good rule is to keep the structure of your Web site as simple as you can and still accomplish what you need to accomplish. In the meantime, remember that two dots followed by a slash, ../, take you backward or up one directory. To go forward, give the directory name followed by a slash, directory/.

### **Link to Precise Spots on a Page with Named Anchors**

Another practical use of links is to help visitors navigate a single, long page. Have you ever visited a site where most of the content was contained on a single page? You could scroll down through the entire page, but you might be interested in only one particular topic covered there. If the page’s author included links to named anchors, you will find it much easier to get around.

#### **Create an Anchor with the name=" " Attribute**

A named anchor helps visitors navigate a long page more efficiently by allowing you to create links that point, not to a different page, but to the anchor. You create an anchor with the name=" " attribute. To put a link at the bottom of a page that will take visitors back to the top without using the scroll bar.

- Place an anchor at the top of the page by typing `<a name="top"> </a>`.
- At the bottom of the page make a link to your anchor by typing the following:  
`<a href="#top">Top of Page</a>`
- Your code should look like this:

```
<html>
<head><title>Named Anchors</title></head>
<body>
<a name="top"> </a>
<p><h1>This is the text of your page</h1></p>
<a href="#top">Top of Page</a>.
</body>
</html>
```

### **Create an Anchor with the id=" " Attribute**

As useful as the `name=" "` attribute is, it also is numbered among HTML's deprecated attributes. The HTML 4 specification recommends using the `id=" "` attribute instead. Don't worry, though. It works the same way as `name`, but with a few new wrinkles and a lot more versatility. Keep in mind the following differences:

- ID names must be unique. If you use `id=" "` to create your anchors, you must choose a unique name for each one you make. It can be any name you choose as long as it begins with either a letter or an underline; IDs may not begin with a number. Also, you can use the ID only one time (as an anchor) in a single document. In other words, you can't have several different elements all using the same ID.
- ID names go inside an element tag. Instead of using the anchor, `<a>`, tags, you can place the ID name inside the opening tag of any element you want to link to: `<h1 id="heading">This is a heading with an ID.</h1>`.
- The anchor element, `<a>`, is still used to create the link. You still must use the anchor element when creating your link and the pound sign, `#`, still must precede the ID: `<a href="#heading">Go to Top</a>`.

The code listed here has been modified to use the ID attribute:

```
<html>
<head><title>Named Anchors Using the ID Attribute</title></head>
<body>
<h1 id="header">This is a heading</h1>
<p>This is the text of your page</p>
<p>This is another text of your page</p>
<p>This is another text of your page</p>

<a href="#header">Top of Page</a><br />
<p>The above link will take you to the top of the page.</p>
</body>
</html>
```

### **Use Named Anchors to Link to Precise Locations on Different Pages**

Suppose you are designing a site for a university and you want to create links that will direct online visitors to a brief paragraph describing a certain course, but you want to include all course descriptions in a single document. One way to do it would be to create a named anchor for each course and create a link to each course's description. It might look something like the following code:

- Anchor (on the `courses.htm` page): `<p id="biology103">Biology 103</p>`
- Link (perhaps on a degree program page, listing required courses): `<a href="courses.htm#biology103">Click for a description of BI-103</a>`

## **Open Links in a New Window**

If you are reluctant to include links to other Web sites because you don't want your visitors to leave your site too quickly, there are several steps you can take to encourage them to stay. One of the easiest is to set up your links so that they open a new browser window. That way, visitors can visit the site you're linked to and remain at your site. You can configure a link to open in a new window by using the `target=""` attribute.

### **Use `target=""` to Open New Browser Windows**

When a visitor clicks on a link with the target attribute, the content will open in a new browser window. To indicate a new window with the target attribute, add the value `_blank` between the quotation marks as in the sample here:

```
<a href="http://www.uobabylon.edu.iq/" target="_blank">Visit University of Babylon</a>
```

## **Insert and Modify Graphics on Your Page**

HTML provides a number of ways to insert images and control their placement and appearance on the page. You have access to some basic image controls with HTML elements. These allow you to put images on your page and use them as links, backgrounds, and navigation tools. After you have a feel for using images with HTML, you can expand your design control with Cascading Style Sheets.

### **Insert a Graphic Image**

Once you have your images ready, you want to place them on your page. Inserting an image is easy. You can place a graphic on your page by using the image, `<img />`, element and its attributes. To place an image, follow these steps:

1. Insert the image with the image element: `<img />`.
2. Identify the image's location with the source attribute: `src=""`. The source attribute tells the browser where to find the image. If the graphic is in the same directory as the Web page, you only need to include the file name, as here:

```

```

However, if you have placed your pictures in a different directory, you will need at least to include the directory where the graphic files is located:

```

```

3. Add a description for non-visual browsers with `alt=""`. The alt (alternate text) attribute provides a place for you to include a text description of the image. It creates a pop-up text box with your description in it when the cursor moves over it. But one of alt's most important uses is to provide content for those who might have images turned off on their browsers or those few who might still be using text-only browsers. Wherever you have placed an image, visitors will see a display that tells them an image is present, and your description will be included. The same description will appear when someone passes the mouse cursor over the image.

```

```

## Control Graphic Size with Height and Width

The height and width attributes enable you to specify the amount of space your image takes up on the page. With these attributes, you can scale an image up or down, depending on your needs.

```

```

## Wrap Text and Align Images

To determine where your image appears on a Web page and how it relates to the text around it, you can use the align=" " attribute. Although this attribute has been deprecated in favor of style sheets, it still provides an easy and reliable way to control the positioning of your images.

### Use left and right for Text Wrapping

To control text wrapping, you use the align attribute with left or right as the value.

The picture will appear on one side of the page, with the text on the other.

```





```

## Specify White Space around Your Image

If you want to create a buffer zone of white space around your image, you can create it with the **hspace** and **vspace** attributes. hspace stands for horizontal space; vspace stands for vertical space. Adding hspace will insert space along the horizontal axis and vspace along the vertical axis.

```

```

## Add a Border

To add a border around your image, include the border=" " attribute. You can specify the size (thickness) of the border in pixels. So, border="5" will insert a border 5 pixels in width. Adding a border to an image is a nice way to give a picture a “large image” feel with a “small image” performance.

```

```

## Create Image Links

You can use an image as a link by putting the <img> element between the anchor tags that create a link.

```
<a href="http://www.google.com">  </a>
```