HyperText Markup Language

HTML&CSS

Unit 6

Hyperlinks

Contents

Hyperlinks

What are hyperlinks? The "a" element

Until now, we have worked with such a unit of information as a web page. Today we introduce the concept of a website. In lay terms, a website is a collection of linked pages, united by a common idea. Hyperlinks allow linking pages to each other.

A hyperlink is an active page element that allows you to refer to other html pages and other resources on the network. When you click on a hyperlink, the specified resource is downloaded to the browser.

The "a" element

Hyperlinks are created using the <a> element. In addition to common attributes, element a can have the following attributes:

- *href* is the address of the resource to be referred to.
- *target* specifies the browser window in which the resource is to be uploaded. It is most often used in conjunction with frames (they will be studied in one of the subsequent lessons). It can take the following values:
- *_blank* — the resource is loaded into a new window;
- *_parent* — the resource is loaded in the parent window of the current document;
- *_top* — the resource is loaded in the upper-level window of the frame structure;
- *_self* — the resource is loaded in the current window. It is the default value;

- "window name" — the resource is loaded in the specified window (frame).
- tabindex is the number that determines the sequence of activating the element with the "Tab" key.
- accesskey is a character used as a combination of keys with the Alt key. Pressing a combination activates a link. This attribute has not been widely used.

There are a number of other attributes that have not been widely spread. You can find out about them in the reference books or on the W3C website.

Examples of hypertext links

```
<a href="my-photo.html">My works</a>
```

references to the current document named my-photo. html in the current directory, creating a hypertext link represented by the words "My works";

```
<a href="/photo/my-photo.html">My photo album</a>
```

references to the file my-photo.html in the photo catalog of the root directory and creates a link represented by the text "My photo album";

```
<a href="./">Photo</a>
```

references to an index file of the current directory (it was /photo/ in the previous example);

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

My site references to the resource on a remote server.

```
<a href="my-photo.html" target="_blank">
   Open in a new window</a>
```

Methods of addressing

We will pay special attention to the method of indicating the resource address. Some typical cases are listed below:

- **Absolute local path.** Specifies the resource address on the current computer in the format

```
file://drive:\folder\...\folder\file
```

- **Absolute network path.** Specifies the resource address on a neighboring computer on the network in the format

```
file://\\host\folder\...\folder\file
```

Here, "host" is the name of the computer or its IP address.

- **HTTP link.** Specifies the address of the resource available via HTTP (Hypertext Transfer Protocol). The following format is used:

```
Http://host:port/folder/.../folder/file
```

The "port" component, however, is not mandatory and is used mainly for connection over a secure link, or when the HTTP server does not work on the default port (default is 80).

- **FTP URL.** A link to a resource available via FTP (File Transfer Protocol). Format:

```
Ftp://user:password@host:port/folder/.../folder/file
```

The user name and password are used to log on via the account. When you need to log on to the server anonymously (if anonymous access to the FTP server is allowed), you do not need to specify them. The "port" component is also not required.

- **Mailto hyperlink.** Starts the system mail client dialog box to create a new e-mail message. The default client for MS Windows is MS Outlook. Mailto hyperlink format: mailto:user@host

- **Relative path.** Used to indicate the location of the resource relative to the current directory. The following notation is used:

- / — root directory of the web server or logical drive

- ./ — current directory

- ../ — parent directory

- directory/ — child directory

In the special case, when the addressable resource is in the same directory as the document, it is sufficient to specify the file name as the address. Relative paths are usually used to indicate the addresses of the "internal" resources of the site. This approach allows you to make the website transfer a quick and easy process with no consequences in the form of link reconfiguration.

We will consider this way of addressing in more detail on the example of building a photo gallery at the end of the lesson.

Practical example

So, let's try to practice the theory. Let it be necessary to mark up a document divided into 5 sections in XHTML. Each section is a separate html file. The index (set of links) is created in a separate file — index.html. We are not particularly interested in the contents of the sections themselves. Particular attention will be paid to creating links to them from the index.

```
<h1>Dave Raggett: The Basics HTML</h1>
<h2>Index</h2>

<ul>
   <li>
      <a href="chapter1.html">Start with a title</a>
   </li>

   <li>
      <a href="chapter2.html">Adding images to
         the page</a>
   </li>

   <li>
      <a href="chapter3.html">Three types of lists</a>
   </li>

   <li>
      <a href="chapter4.html">Additional information</a>
   </li>
</ul>

<p style="font-style: italic; font-size: 10pt;
   color: #666666;">
   This guide is about <b>HTML</b>, therefore some
   rules do not fit and even violate the picture we
   draw for you since the beginning of the HTML
```

```
    course. The author of this guide is
    <a href="http://www.w3.org/People/Raggett"
    target="_blank">Dave Reggett</a>.
    Yu can fine detailed information on HTML and XHTML
    on the website <a href="http://www.w3.org"
    target="_blank">W3C</a>.
</p>
```

Note that the links at the bottom of the document (the last paragraph) open in a new window, due to the presence of target="_ blank" attribute. This is usually done with links that point to resources external to your website. Thus, the user will not "lose" your site by clicking on the link.

Image Link

In the "a" element you can put not only text, but also an image. This is how graphic links are created. Image links in some cases look much more aesthetic than text. Here, however, we are trapped in a not very pleasant surprise — the images enclosed in the "a" element acquire a frame. In order to get rid of it, you must use the border attribute of the img element:

```
<a href="...">
    <img src="image.gif" alt="Image-link"
        border="0" />
</a>
```

Once again I remind you that when using images as links, do not forget to specify the alt attribute for them. If the image cannot be displayed, the browser will make its text description a hyperlink.

Example: creating a navigation bar of the website

Now, with a small example, we will gradually design and implement a simple graphical navigation bar of the website. It will look like this:

It seems that everything is simple: we take an image, place it on the page ... What could be simpler? However, various fragments of this image must be hyperlinks to various resources. Thus, we come to the need to cut the image into several parts.

Here's what I got:

As a result, we have three "buttons" — hyperlinks "Back", "Index" and "Forward". Put all this stuff on the page:

```
<div style="text-align: center;">
    <img src="navbar/navbar01.gif" alt="" />
    <img src="navbar/navbar02.gif" alt="Back" />
    ...
    <img src="navbar/navbar09.gif" alt="" />
</div>
```

As they say — "it is the first step that is troublesome!" Between the images there are gaps now, although no one planned it. The thing is that MSIE added a space character between the images. This is because the img elements are separated by the line break in the page code. Knowing this, the state of things is easy to change by removing the whitespace characters (line breaks, tabs...) between the img elements.

```
<img src="navbar/navbar01.gif" alt=""/>
<img src="navbar/navbar02.gif" alt="Back"/>
```

Now everything is normal! Provide the code with hyperlinks. In order not to think up addresses, use "#" as the value of the href attribute:

The result looks clearly like something is wrong... The whole thing is in the frames around the images. We get rid of them by adding the border = "0" attribute to the link images:

```
<a href="#">
    <img src="navbar/navbar02.gif" alt="Back"/>
</a>...
```

Now you can link multiple pages to a simple single navigation bar by placing its code at the beginning of each page and correctly setting up the links.

Unit 6.
Hyperlinks