Web Site Construction III:

The Web Site

David Gerbing
School of Business Administration
Portland State University
# Table of Contents

Typeface Conventions.................................................................................................................. 3

## SECTION III

**CONSTRUCTING THE WEB SITE** ........................................ 5

Displaying the Complete File Name with Windows or Macintosh OS ........................................... 6
Web Directory Structure and Pathnames ......................................................................................... 6
Saving Excel Files as a Web Page ................................................................................................... 8
Displaying an Access DBMS Report as a Web Page ....................................................................... 8
Moving the Web Site to a Web Server ............................................................................................. 9
Accessing Web Pages on a Web Server .......................................................................................... 11
Checking for Missing Links ........................................................................................................... 12
Optional: Web Standards ............................................................................................................... 13
Optional: Character Set Standards ................................................................................................. 14

## SECTION V

**ADDENDUMS** ......................................................................................... 15

More Information ......................................................................................................................... 16
On-line HTML & CSS tutorials ....................................................................................................... 16
Aid for Building Web Pages ........................................................................................................... 16
Examples ........................................................................................................................................ 16
Books ............................................................................................................................................ 16
Placement of CSS Style Rules within a Document ......................................................................... 17
  Internal style sheets ..................................................................................................................... 17
  In-line styles .............................................................................................................................. 17

**INDEX** ................................................................................................................. 19

© David W. Gerbing, 2016
Typeface Conventions

<table>
<thead>
<tr>
<th>Typeface</th>
<th>Indicates …</th>
</tr>
</thead>
<tbody>
<tr>
<td>literal</td>
<td>a literal computer keyword, entered exactly as shown</td>
</tr>
<tr>
<td>example</td>
<td>a specific example of a word or phrase the user is free to choose</td>
</tr>
<tr>
<td>choice</td>
<td>the name of the attribute for which the user is free to choose his or her own word or phrase</td>
</tr>
</tbody>
</table>

Consider the following actual code,

```
<img src="images/jacket.jpg" alt="jacket">.
```

Enter the `<img src=""` exactly as shown. Similarly for the `alt=""` and `">`. The phrase `images/jacket.jpg` is based on names supplied by the user, in this case a directory called "images" and a file within that directory called "jacket.jpg".

In contrast to the actual code displayed above, the more general definition displays as

```
<img src="name" alt="description" />
```

Here the user specifies his or her own chosen name, such as `images/jacket.jpg` from the above example code.
This section shows how to put the different individual web pages together to form the complete web site.
Each web site consists of at least the web pages that define the site plus associated graphics and images. More complex sites contain other types of information such as Flash files for animated effects and Javascript for specific actions such as a slideshow or a picture fade out. Only the basic structure is considered here.

**Displaying the Complete File Name with Windows or Macintosh OS**

Each file name on any Windows, Macintosh, or Unix computer system includes a filetype extension. This extension identifies the application program that generated the file. Examples of these extensions appear below.

<table>
<thead>
<tr>
<th>Extension</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>.doc</td>
<td>Word</td>
</tr>
<tr>
<td>.xls</td>
<td>Excel</td>
</tr>
<tr>
<td>.txt</td>
<td>NotePad</td>
</tr>
<tr>
<td>.html</td>
<td>HTML</td>
</tr>
<tr>
<td>.jpg</td>
<td>JPEG graphic</td>
</tr>
<tr>
<td>.gif</td>
<td>GIF graphic</td>
</tr>
</tbody>
</table>

By default Windows and Macintosh do not display the extension of a filename. For example, is a file named `image1` a JPEG file (`image1.jpg`) or a GIF file (`image1.gif`) or neither? The problem is that the full file name must be known before web pages can reference and display the file.

To display the complete file name using Windows XP, double-click on the My Computer icon on the desktop. Go to the Tools menu and select Folder Options and then select the View tab. For Windows 7, click the Start button and then click the Folder Options link that appears. For Windows 8, open Windows Explorer and go to View, click Change folder and search options. Then, for all versions of Windows, Uncheck the following option:

- [ ] Hide file extensions for known file types

To display the complete file name using Macintosh OS X, go to the Preferences… menu option under the Finder menu. Click on the Always show file extensions checkbox.

**Web Directory Structure and Pathnames**

Files are organized on computer systems into directories and subdirectories. This file organization within directories is called the *directory structure*. The directory structure for a web site follows a specific form. The most basic instance of this form follows.
public_html  [directory: contains all web files; use exactly this name]
styles.css  [file: cascading style sheet]
index.htm  [file: first web page; use exactly this name]
page2.htm  [file: second web page; page2 is any name you wish]
page3.htm  [file: third web page; page3 is any name you wish]
... and so on  [rest of the web pages]
images  [directory: contains all images files, .jpg or .gif; use any name]

Specify the entries, public_html and index.htm, exactly as shown. Also, some web servers require that the web files end in .html instead of .htm. The naming convention is just that, a naming convention. Whether a file ends in .html or .htm, it is still an HTML file.

Almost all web pages reference other files on the site, either images or other web pages. A crucial aspect of web site development is to reference these other files with the correct pathname. The pathname describes what directory the file is in, as well as the name of the file and must exactly match the actual name and directory location of the file. The backslash, /, indicates a directory.

Some people struggle for hours trying to get a web site working when the only mistake is to refer to nonexistent files. If you wish to link to a file called page2.html with the anchor element, then there really must be a file called page2.html – not Page2.html, or page 2.html, or page2.htm. The same principle applies to linking to image files with the img element.

A pathname can be specified as an absolute (or full) pathname or as a relative pathname. An absolute pathname describes the exact position and name of a file from the topmost directory. For example, consider a Windows system in which the public_html directory is located within a directory called documents that is on the C: drive. The absolute pathname of a graphic called jacket.jpg in the images directory on this system is

        C:/documents/public_html/images/jacket.jpg

Web sites are usually developed on a local computer – not the web server that will eventually serve the site over the Internet or a company intranet. The web server application, for example, may be on Unix system and so not even have a drive called C. As such, absolute pathnames would generate a missing link instead of a valid file reference.
Express the file references on a web site, such as from image and anchor elements, as *relative* path names.

A relative pathname describes some other file's location relative to the current working directory. The working or current directory is the directory in which you are currently working, which for web site development, begins with `public_html`.

The pathname of a file in the current working directory is just the name of the file by itself. For example, in the web site directory structure listed above, the home web page, `index.htm`, is at the same directory level in the `public_html` folder as is the second web page, called `page2.htm`. So, relative to `index.htm`, the relative path name of `page2.htm` is just `page2.htm`.

In the web site structure previously specified, all graphics are stored in an images directory within the `public_html` folder. Presume that the home page should display a graphic called `jacket.jpg`. The file reference for this image is the relative pathname `images/jacket.jpg`.

**Saving Excel Files as a Web Page**

Web sites contain HTML files and graphic files referenced by these HTML files. Most modern applications, such as the MS Office suite, can save the content of a file as an HTML file, which then can display as a web page when opened by a browser. For Excel, open the content of interest and then from the File menu, select *Save as Web File*... . Place the resulting HTML file in the web site folder and then link to it, such as from the home page of the web site.

**Displaying an Access DBMS Report as a Web Page**

Access reports can be saved in a variety of formats, including HTML. A report saved in the HTML format is a complete web page that with little or no modification can be integrated into an existing web site. To save a report, first define the report. Then go to the External Data tab and select the second More button, about half-way across the window. Choose the HTML document option.
Save the resulting file to your website folder. Then simply link to the produced report to display on your website.

**Moving the Web Site to a Web Server**

A web site is located on a web server that allows other Internet users access to the site. The web server serves the content of the site to the user’s web browser that interprets the incoming HTML file and formats the corresponding screen display. Fortunately, the web site can be developed on a personal computer using editing tools for which you are already familiar. Once the web site is developed the entire site and directory structure must be moved (uploaded) to a web server.

A standard means of transferring files on the Internet is to use the File Transfer Protocol or FTP, or the corresponding secure version, SFTP, which Portland State University uses. Some people routinely use E-mail attachments to move files from one place to another, but FTP is a more elegant and more general solution than E-mail for file transfer. Just as HTTP for serving web pages is a client-server protocol, FTP is another client-server protocol that operates over any TCP/IP network including the Internet.

Many different FTP server programs and client programs exist for all the primary operating systems. An FTP server is needed to allow the developer to move files to the web server.
Virtually all computers that run a web server (HTTP) program also run a FTP server program allowing the contents of the web site to be uploaded from the developer’s personal computer to the web server computer.

To access an FTP or SFTP server you first need an FTP client application such as the free FileZilla client. Obtain this software from https://filezilla-project.org. (Only download from the official site as some 3rd part sites contain malware.)

To login to the PSU server on your account, you need to enter the host to which you wish to login, your PSU account ID and password.

```
Host: | Username: | Password: |
```

Enter the following host information: sftp://sftp.myfiles.pdx.edu
Then follow with your Username and Password.

For off-campus FTP access, PSU now requires two-factor authentication over a VPN (virtual private network).

1. On your computer where your website files are stored, install the Cisco AnyConnect VPN Client
2. On your smart phone, install the free Duo Mobile app from, for example, the Apple Store
3. Enroll your smartphone into the Duo Verification system
4. Connect remotely via the VPN client, which then pings your Duo Mobile app, from which you then approve the connection.

Once logged in to the PSU Unix system, you have two lists of files. The list on the left is from your own computer. The list on the right is from the server to which you are logged in.

```
Local site: /Users/gerbing/Dropbox/BA325/ | Remote site: /
```

Once logged onto a remote system, transfer files from your local hard drive to the remote computer.

The file directory on the left side applies to the local PC and the file directory on the right side applies to the remote system. The single forward slash at the Remote site indicates that you are at the top of the (Unix) file directory.
Identify the relevant directories on the PC and the remote system for moving from and to. The `public_html` folder in which a web site is stored is in the folder called Unix—Web-Folder on the Remote site.

Use the `..` icon to move up the respective directory structure. To move down, double-click on a folder. To find the `public_html` folder, double click on Unix-Web-Folder.

To move files from your computer to the web server, simply drag the contents of your web site to the `public_html` folder on your web server. Do not drag the folder itself over to the server’s `public_html` folder. For example, the file that corresponds to your home page, index.html, should be at the first level of the server’s `public_html` folder, not buried within another folder inside of `public_html`.

**Caution →** The web server will only find web site files by default that are at the first-level in a `public_html` directory, so make sure that all your web files on the web server are directly stored in this directory and not nested within folders.

You can move the entire web site with a single move: from within your web site folder on your local hard drive to a `public_html` directory on the web server. Open your web site folder on your local drive and then drag the contents of the folder – not the folder itself – over to `public_html`.

Before you move your web site, however, Odin users should first delete the token site that is already pre-installed when the account was first created. To delete a file or folder, select the file or folder and then click on the **Delete** key. Open up the Odin `public_html` folder and delete the `index.html` file. Open the `img` folder and delete the resulting image. Then move back to the `public_html` folder with the green arrow and then delete the `img` folder.

**Accessing Web Pages on a Web Server**

Most web servers have the `public_html` directory and proper privileges already in place, so just FTP your site to the directory. The corresponding URL (web address) for an Odin account is `web.pdx.edu/~userid` where you provide your own Odin userid.

The Odin web server runs a version of the Unix operating system. Unlike Windows and Macintosh, Unix is case sensitive. `Image.gif` and `image.gif` on Windows refer to the same file, but on Unix these are different files. You should
consistently use lowercase file names and path names when you develop your web site on your PC. Otherwise references to other web pages and images might work on your PC but not work on a Unix web server.

Another distinction between Unix and Windows is that Unix does not recognize certain characters in a file name that are valid under Windows. For example, the & can be part of a Windows file name, but not Unix.

When manually entering all the HTML markup, an important consideration that eases the transition to a working website is to use only simple filenames. Avoid long file names and avoid punctuation and spaces in your file names.

The problem that typically requires the most time in achieving a working web site might first appear as a simple objective: match the reference to a file name in a mark-up tag to the actual path name of the file – not almost the path name, but literally character for character the same name.

The file name What is-this thing called.jpg is more difficult to reproduce than is the file name w.jpg. If the name of an image file contains a long string of characters, possibly upper and lower case and including punctuation, simply rename the image file before linking to a web page. Of course, if the img tags and a tags are entered by browsing for the file with a WYSWYG visual editor such as Dreamweaver, then the correct spelling and case of each letter will be automatically entered. Even so, the simplicity of short filenames with no spaces and all in lowercase is appreciated.

Checking for Missing Links

A web site can contain hundreds of individual web pages with thousands of links to other pages on the site as well as to pages on other sites around the world. One annoying problem that all web users have encountered is clicking on a link that goes nowhere, a missing link, and then observing a message such as presented here.

Web design applications such as Dreamweaver provide a check for all links in the site. The W3C provides a free service that performs the same task: http://validator.w3.org/checklink

Provide the URL (web address) of the home page of a site and the provided program will check for missing links on all pages at that site.
Optional: Web Standards

Standards define HTML, CSS and other technologies with the goal that different web browsers all render the same HTML page the same. Each mark-up tag should have the same effect regardless of the browser that renders the page. This goal is not perfectly realized in practice, but the implementation of common standards allows most modern browsers to provide approximately the same rendering of the same web page from the underlying HTML mark-up and CSS style rules.

The international standards organization that certifies web standards is the World Wide Web Consortium or W3C, though that task is now being shared with an industry based organization called the Web Hypertext Application Technology Working Group (WHATWG). The W3C director is Tim Berners-Lee, the inventor of HTML and other crucial web standards. The web has evolved quickly and the standards for writing web pages are also evolving. Technology and the associated standards available today either did not exist, or

The W3C provides a free validation service to certify that a specific web page conforms to the standard specified by the DOCTYPE, such as HTML5. The address for this service is http://validator.w3.org/. To use the validation program, go to the web page and specify the file to validate. The file may be located either already on a web site, or on a local disk. Clicking the relevant Check button on the web page yields the validation output, illustrated below for a successfully validated page.

What types of problems cause an HTML file to not successfully validate? To review:
1. Elements such as anchor links and images must be enclosed within containing blocks such as a paragraph <p> or a division <div>.
2. *Every* mark-up tag must have a beginning and an end, *always* indicated by a backslash. For example, if mark-up defines the beginning of the paragraph, `<p>`, there must also be mark-up that defines the end of the paragraph, `</p>`, in this case a separate tag. Tags such as the image mark-up and the line-break mark-up, which are implemented with only a single tag, require the backslash at the end of the single tag. For example, `<br />` for a line break.

3. All styling is done with CSS. No legacy HTML styling is allowed.

**Optional: Character Set Standards**

The Internet is an international phenomenon. Different alphabets around the world use different characters in a text file. The specific characteristics of a text file, such as an HTML file, depend on the underlying alphabet used to build the words on the page. Each web page displays with one of these character sets, which are standardized for each of the world’s major alphabets.

The most modern coding scheme is based on Unicode, such as the character coding called utf-8.

The author of a web page should specify the specific character set the browser uses when displaying the web page. Specify the standard character set with the following meta tag.

```
<meta charset="utf-8" />
```

This tag appears at the beginning of the Head section, following the `<head>` tag.
SECTION V

ADDENDUMS

This section provides ancillary information regarding the information in this paper as well as additional information on the use and construction of web sites.
More Information

On-line HTML & CSS tutorials

Ultimate reference: www.w3schools.com
Beginning: www.stuffucanuse.com/CSS
Lists and some other topics: CSS.maxdesign.com.au

Aid for Building Web Pages

Color displays for choosing rgb values: on class web site

Examples

Examples of different CSS rules on the same content: www.csszengarden.com
Randomly designed CSS rules to the same content: www.strangebanana.com

Books


Placement of CSS Style Rules within a Document

The placement of style sheets generally is in external files, an external style sheet discussed on p. Error! Bookmark not defined.. Two other possibilities also exist.

Internal style sheets

An internal style sheet is located in the Head of a single HTML file. Style rules in an internal style sheet pertain only to that specific web page. Enclose the rules within the <style> tags, as shown below.

```
<head>
<title>A Real Web Page</title>
<style type="text/css">
    ---- CSS style rules here ----
</style>
</head>
```

Braces enclose style rules placed in either an external style sheet or an internal style sheet.

In-line styles

An in-line style rule is embedded within a specific HTML tag. This type of style rule formats only the one corresponding element, such as the single paragraph in this example. The enclosure of the style rules defined in the third location employs a slightly different syntax. The in-line style places the style rule right in the element’s mark-up tag with the style attribute,

```
<tag style="CSS style rules here ">
```
as in

```
<p style="color: blue; font-size: 2em;">
```

In general the use of in-line styles should be minimized, but are occasionally useful if the style is unique to a single element and is likely not to be changed or used anywhere else in the site. Placing all the style rules in one place, preferably an external file, makes maintenance of a web page’s and web site’s design more straightforward.
INDEX

A
Amaya, 35

C
character set
ISO-8859-1, 18
CSS, 20
CSS placement
external style sheet, 23
in-line style, 4
internal style sheet, 4
CSS rules
appearance, 21
positioning, 21

D
directory structure, 38
DTD, 15

E
element, 3
block, 7
inline, 8

F
file
binary, 2
ftp, 41

H
HTML, 2
HTML editor, 34

M
markup
DOCTYPE, 15
markup tag, 3
markup-tag
body, 3

div, 7
head, 3
heading, 5
html, 3
hypertext anchor, 8
image, 8
lists, 5
paragraph, 5
span, 9
tables, 6

P
pathname, 38
absolute, 38
relative, 39

S
selector, 22, 26
class, 27
descendant, 32
element type, 22
pseudo-class, 23

text editor, 4

U
URL, 8

W
W3C, 15
web server, 41
white space, 5
working directory, 39

X
XHTML, 14
XML, 14