

# Chapter 8 Introduction to Web Publishing

## About This Chapter

This chapter discusses Web publishing, and in particular a publishing language called XHTML used in the Web publishing. “Language” here refers to code written in a form which computers can understand, but you do not need to worry. XHTML is a publishing language written in a form which is readable by people as well. It may take some getting used to, but even beginners will be able to create a simple Web page in about an hour once they understand the concepts.

This guide is not intended to help you become an expert Web page designer. Web publishing largely depends on individual talent of individuals in creating color scheme and design<sup>1</sup>. This chapter’s aim is to help you create a simple and effective Web page, which you will be able to create if you just follow certain rules.

What is important here is the relationship between XHTML and “structured documents.” XHTML is a computer language for Web publishing, and what XHTML directly describes is document structure. In other words, if the structure of a document is well-formed, it is very easy to rewrite it as a Web page. The main purpose of this chapter is to talk about XHTML, but this chapter also aims to help you learn about structured documents at the same time.

XHTML stands for Extensible HyperText Markup Language. It is a publishing language which conforms to a language system called XML, and is a successor to HTML, which has been traditionally used for Web publishing, and will be described later. Misinterpretations of HTML and invalid coding have been allowed because of competition in Web browser<sup>2</sup> development, bugs, deviation from HTML specifications and for other reasons. XHTML is devoid of such ambiguity and requires a certain level of strictness when writing it. You may think XHTML is cumbersome, but you can find and correct errors in it more easily.

There are different kinds of XHTML. This chapter focuses on XHTML 1.0, the most basic form of XHTML and which is compatible with HTML 4.01<sup>3</sup>.

It is important to write XHTML correctly as a computer language, but you must keep in mind here that the content in the XHTML must be also correct. The correct use of XHTML in a document which contains incorrect information is meaningless. The significance of writing a document using correct XHTML lies in communicating the information in it efficiently and effectively to as many readers as possible.

The information you have is of no value until you communicate it to someone in some form. Information you learn at university and new knowledge produced there is mostly communicated in the form of oral presentations or papers, but today Web publishing has also become an important means. We hope that you will acquire correct knowledge and be able to share it with many people.

---

<sup>1</sup> Having said that, color scheme and design theories do exist. As described later, color schemes and design are also important from the viewpoints of accessibility and usability, but this guide will only mention color schemes to an extent. For more information, refer to the bibliography.

<sup>2</sup> For example, Netscape Navigator by Netscape Communications Corporation and Internet Explorer of Microsoft Corporation.

<sup>3</sup> The latest version at present is XHTML 1.1, and a draft of XHTML 2.0 has been published. But you can work with these versions if you have a good understanding of XHTML 1.0.

## 8.1 Overview of Web Publishing

This section presents a broad overview of Web publishing before you start Web publishing. You may start reading from Section 8.3 “Starting XHTML” (p. 122) if you want to jump right in, but it is important to understand the fundamental principles of Web publishing in order to carry out actual work.

This section talks about two things: (1) the client-server model, and (2) file formats.

### 8.1.1 Client-server Model

In a nutshell, the Web involves the acquisition of information via a network. A full understanding of the communication characteristics of the Web is necessary not only for understanding Web publishing but also for understanding the overall picture of the Web, and even for just using the Web. The following subsection roughly describes what a Web browser communicates and how<sup>4</sup>.

The fundamental communication framework over the Web is the so-called “client-server model.” This means that there are two kinds of party on the Web: “servers,” which provide services, and “clients,” which receive the services<sup>5</sup>. A “client” is a Web browser installed on the PC you are using. Typical Web browsers are Microsoft Internet Explorer, Netscape Navigator and Mozilla Firefox. We call the servers we refer to in this chapter “Web servers” because they provide Web services, but actually a web server is a program running on a computer on a network<sup>6</sup>.

It may help you understand what is going on if you recall what happens when you access a Web page. As you know, no Web server sends you information without your request. A Web server waits for requests from clients and passes information in response. That is, Web browsing starts with a client request. You can send a client request to a Web server by, for example, typing in a URL (to be described later) in a Web browser, to be exact in the Internet Explorer “Address”<sup>7</sup> Bar, or by selecting a page from “Favorites” (see Figure 8.1 and Figure 8.2). Whether you are typing in a URL or selecting a page from Favorites, what you are actually doing is specifying a “URL” for your Web browser.



Figure 8.1: Internet Explorer “Address” Bar



Figure 8.2: Internet Explorer “Favorites”

URL stands for Uniform Resource Locator<sup>8</sup>. “Resource” here can be taken to refer to various kinds of information (data) which exists across the Internet. The data mostly, but not always, is in the form of “files.”

<sup>4</sup> We say “roughly” because the simple model presented here has been used since the creation of the Web and nowadays network structures that do not conform to the model are often used and new models are being developed every day. However, the basic form never changes.

<sup>5</sup> A “proxy” is also involved in the case of Waseda University, but we will skip the details.

<sup>6</sup> Apache from the Apache Software Foundation and Internet Information Services (IIS) from Microsoft are examples of popular Web servers.

<sup>7</sup> This name seems misused because what is typed into the Internet Explorer “Address” Bar is a URL to be navigated to. Note that “Address” and “Favorites” are merely names used in Microsoft’s software.

<sup>8</sup> Sometimes the term Uniform Resource Identifier (URI), which refers to a wider concept, is used instead.

Uniform Resource Locator or URL roughly means “an object which specifies the location of information on the Internet using a uniform method.”

Figure 8.3 shows an example of a URL and its meaning. The basic building blocks of URL are a “scheme,” an “authority” and a “path.”

<code>http://www.waseda.jp/mnc/index-j.html</code>
<span style="display: inline-block; width: 30%; text-align: center;">スキーム</span> <span style="display: inline-block; width: 40%; text-align: center;">オーソリティ</span> <span style="display: inline-block; width: 30%; text-align: center;">パス</span>

Figure 8.3: Uniform Resource Locator (URL)

The scheme contains “http,” “ftp,” and so forth. You may consider a scheme as being a framework for accessing a resource. Since the Web uses Hyper Text Transfer Protocol (HTTP)<sup>9</sup> to communicate, the scheme of the URL used is “http.”

You may think that the “authority” is the computer which controls the resource. Every computer or communication device on the Internet is assigned with a unique number for administrative purposes. This number is called an IP address, and consists of four numbers between 0–255 separated by periods. For example, 133.9.1.3. Any computer on the Internet can be uniquely identified by its IP address. The disadvantage of such IP addresses is that they are hard to memorize. In order to solve this problem, a database that converts names (domain names), which are easier to remember, into IP addresses may be used. For example, “ns.cfi.waseda.ac.jp” is converted into “133.9.1.3”<sup>10</sup>.

A “path” shows where the resource is located in the authority. A path is usually specified using hierarchical notation separated by slashes “/.” In other words, the concept of folders (directories) is applied here as well. A path specifies the directory in which a specific resource (a file) is and the specific file name. Note that a file name part may be omitted. For example, you may only know the scheme and the authority (or a part of the path) of a Web site (for example, `http://www.yahoo.co.jp/`) you want to visit for the first time. Since the file name can be omitted, you can access the site in such cases, too.

When a file name is omitted, the index file of the directory is treated as the specified file. An index file is a page which serves as the gateway to the directory. Usually an index file is named like “index.html” or “index.htm,” but this may change depending on the server settings. As an experiment, try accessing `http://www.yahoo.co.jp/` and `http://www.yahoo.co.jp/index.html`; you see that you will get the same result in both cases.

The key point you need to grasp here is that the Web is a system to transfer files from one computer to another. As you can see from the fact that the last two letters of HTTP stands for “transfer protocol,” the Web is essentially a file transfer system.

<sup>9</sup> In order for multiple computers to communicate with each other, it is necessary to specify in advance a communication procedure. Such a communication method is called a protocol. Communication between computers is carried out using electricity, light, radio waves, and so forth. Types of communication methods include physical method which dictates things like how to send electricity, and logical method which dictates specifically how to send and receive data, assuming that electricity is sent. The “scheme” in a URL specifies a logical communication method in most cases.

<sup>10</sup> This conversion system is called the Domain Name System (DNS).

## 8.1.2 File Formats

Web publishing is composed of the mixing of two different types of files on one page: (1) text and data, and (2) other types of files (mainly multimedia files). In simple terms, a mixture of text, pictures and other content coexists in a Web page.

The textual part which consists of characters and the document structure (to be described later) are expressed simultaneously in XHTML in a “text” format. That is, the text and the document structure are specified in the same file in a text format. Unlike other data formats, this feature where text and structure are specified together is unique to XHTML.

You have already learned about the “file” concept. But you need to pay attention to the file name here. Usually, a text file has the “txt” extension, but an XHTML file has “html”<sup>11</sup>. When you write a file in XHTML, remember to use the “html” extension.

Meanwhile, if you look around various Web pages, you rarely find Web pages written only using text. Most pages are created using “multimedia,” such as charts, pictures, photographs, videos and sound. Since it is difficult to put such information together in a text file, different types of data are kept as separate files from the main text and called up whenever needed.

That means photographs and other data are set aside as separate files, and an instruction such as “place a photograph here” is specified in XHTML where needed in the same way as formatting information is specified. The same method can be used with any file (except for text files) such as photographs, pictures, and video files.

Note that, however, the file formats which can be embedded in a Web page vary depending on the Web browser viewing it. For example, in most cases a well-known file format called Adobe Flash cannot be viewed by common Web browsers unless separate “plug-in” software is installed.

Since most Web browsers support pictures and photographs as standard, viewing such files should not be an issue. But you need to keep in mind that even if you can view certain data on your computer, it does not necessarily mean other people can do the same. In this chapter, you will learn in a broad sense about the methodology of information dissemination. When you disseminate information, you need to give careful consideration to the receivers of information.

The extension of XHTML is html and its content is text. Table 8-1 (p. 119) summarizes the file formats commonly used for Web publishing.

## 8.1.3 Section Summary

The Web is a two-way communication between a client and a server. Clients and servers are both computers, and Web browsing is carried out by the client sending a request to the server and the server responding to the request. A request is made by sending a URL from the client to the server; that is, by transmitting a request to send a resource, i.e. files. The Web server sends an HTML file, which is the main body of a Web page, and multimedia files, such as images, to be read in the HTML file, and the client displays these files.

There are various kinds of files sent and received between a server and a client. There are two prominent types of files: files which are displayed on Web pages (main XHTML files, images, etc.), and programs (JavaScript and VBS) and documents processed by other programs (MS Word documents and PDF).

---

<sup>11</sup> The “htm” extension may also work in this situation. This depends on the settings of the Web server (to be described later). Conventionally only the “html” extension has been used. But since only three-character extensions could be used with MS-DOS and Windows 3.1, the “htm” extension was created and is still in use. You should be able to use the “html” extension in almost all cases. Meanwhile, if you have a problem with using the “html” extension, you may want to ask the administrator of the server you are using about the problem.

Table 8-1: General File Formats in Web Publishing

Format	Extension	Remarks
JPEG	jpg	JPEG stands for Joint Photographic Experts Group. The name of the organization which developed the image format became the name of the file format, which features a high compression ratio and is often used for full-color images for digital cameras and other devices.
GIF	gif	GIF is a graphics format developed by CompuServe Incorporated, which was a major PC communication company in the United States. GIF can display 256 colors and also supports animation through the storage of multiple images.
PNG	png	PNG stands for Portable Network Graphics. Since PNG was developed after GIF, it is superior to GIF in every respect. PNG also supports full-color images. However, since PNG uses lossless compression, you should use JPEG in cases where a little degradation is acceptable and use PNG when degradation is not acceptable or the image is a diagram or a chart.
PDF	pdf	PDF stands for Portable Document Format, a standard for electronic documents by Adobe Systems Incorporated. Using Adobe Reader, which is also distributed by Adobe Systems free of charge, you can view PDF files. Use this format when distributing documents which you do not want to be altered.
Word	doc	DOC indicates a MS Word document by Microsoft Corporation. Since it may contain a macro coded in a language called Visual Basic for Applications (VBA), you should never open a doc file if you do not know what it contains. If you must open such a document, you should disable VBA in MS Word first.
Excel	xls	XLS indicates a MS Excel document by Microsoft Corporation. Since it may contain a macro coded in a language called Visual Basic for Applications (VBA), you should never open a doc file if you do not know what it contains. If you must open such a document, you should disable VBA in MS Word first.
CSS	css	CSS stands for Cascading Style Sheets. It provides formatting information used in Web publishing.
VBscript	vbs	VBS stands for Visual Basic Script. Visual Basic Script is a scripting language developed by Microsoft Corporation based on BASIC. You should never download or open a Visual Basic Script file if you do not know what it contains. Unknown Visual Basic Script files are often malicious programs like a computer virus. Its content is a text file.
JavaScript	js	This language was developed by Netscape Communications Corporation, and it works on Web browsers. JavaScript is supported by almost all Web browsers. However, the level of support and the language specifications employed vary depending from Web browser to Web browser. Although JavaScript can be disabled by changing the security settings of your Web browser, since Web applications which use JavaScript becoming more common lately, you may want to enable it as the need arises. Its content is a text file.

A file's type can be determined from its extension. But as mentioned earlier, the Web is a system for file transfer. It is important not to access any suspicious sites or not to download any file, the contents of which you do not know. If you think about these points from the standpoint of a Web publisher, you will naturally develop the right mindset for Web publishing.

## 8.2 Setting up Your PC for the Production of XHTML Documents

This section discusses how to set up your PC in order to write XHTML and the knowledge necessary for the production. If you know what “text files,” “text editors,” and “extensions” are and if you have been using your favorite editor and FTP software already, you may skip this section.

### 8.2.1 Editor

XHTML data is in the form of “text,” which people can reference, read and write directly. What we need to create this type of data is an application called a “text editor.” “Notepad” (**Start -> Programs -> Accessories -> Notepad**) for Windows and “TextEdit” for Mac OS X are readily available text editors. You can create the same type of data using either application.

You can use these standard text editors, or you can use other editors distributed as free software or as shareware. The first and foremost advantage of acquiring a separate editor is its high level of functionality. For example, as mentioned earlier, actual text and structural information are described in the same XHTML file. Also, as mentioned earlier, this structural information is called “tags,” and some editors come with an input support function for tags or allow you to display tags in different colors from the text.

Moreover, since a text editor is software specialized for text input, it runs faster than word processors.

It is high performing, runs fast, and is also free—it would be a waste not to use it. This guide introduces you to two text editors here, one for Windows, and another for OS X.

- Windows  
TeraPad (<http://www.5f.biglobe.ne.jp/%7Et-susumu/>) (Japanese Only)
- Mac OS  
mi(<http://www.mimikaki.net/en/index.html>)

You can use either these editors for free. However, these editors have not been created instantly out of nothing. The developers have spent their valuable time creating this software and released it for us. You must use it with gratitude. If you are truly grateful, you can send a thank you email to the developer, or in the case of mi, since the developer accepts voluntary donations from users, you may want to pay for it.

For information on how to use these editors, refer to the manuals that come with them or the respective Web pages. A text editor is a program which you use to enter characters by typing on your keyboard. The important features of a text editor are its abilities to create files, open existing files, and to end safely. In addition, the proper management of created files on your PC is important, although this has nothing to do with editors in particular.

The text editor “Hidemaru” is installed as standard on the PCs in the Computer Rooms of Waseda University.

### 8.2.2 FTP Software

FTP stands for “file transfer protocol.” As the name implies, FTP is a communication method used for the transfer files between computers. We now have another file transfer protocol in addition to HTTP.

Why do we need another piece of software for file transfer?

HTTP is used for communication between Web servers and Web clients. Clients receive documents from servers. In this section you will be creating Web pages, not just simply browsing the Web. That means you have to upload the elements of the documents (XHTML documents, images, etc.) you are going to create to a Web server. How do we upload the documents?

In most cases, authentication using a login ID and password is not required to browse the Web<sup>12</sup>. That is because, as a general rule, information is published on the Web based on the premise that everyone should have unrestricted access to it.

However, what is unrestricted is the browsing of the Web, not alteration of Web pages or the creation of new pages. The editing of a Web page someone else has created without permission is a criminal act. In other words, as far as Web pages are concerned, “creating and editing” and “browsing” are completely asymmetrical tasks. Therefore, a framework called FTP is often used with which Web pages are uploaded to a server and edited after authentication using a login ID and password<sup>13</sup>.

FTP is also based on the client-server model. That means there are FTP servers and FTP clients. In this case, an FTP server is a computer where you upload and modify files. In this sense, an FTP server is identical to a Web server. An FTP client is a computer which you operate.

Windows comes with an FTP client as standard, but to use it, you have to enter commands from the “command prompt.” It is hardly user-friendly. You should use a more user-friendly FTP client.

There are freeware FTP clients available for both Windows and Macintosh.

- Windows: FFFTP  
<http://www2.biglobe.ne.jp/~sota/index-e.html>
- Mac OS: Fetch<sup>14</sup>  
<http://fetchsoftworks.com/>

We will explain how to transfer files later. Although the general concept is the same, details differ from server to server. This guide explains how to transfer files using FFFTP, bearing in mind how PCs are set up at Waseda University. The different ways of transferring files using FTP are all quite similar because all of FTP clients are based on the same framework, FTP. The best way to learn how to use software is to use it repeatedly and so become familiar with it. You do not need to use the above-mentioned software. You can find FTP software you like and install it on your PC.

### 8.2.3 Other Software

To create XHTML documents and transfer them to a server, you only need a text editor and FTP software. However, it is helpful to have several other applications in addition to these.

First, it is useful to have software for graphics and image processing. As they say “A picture is worth a thousand words,” a photograph or an image is sometimes more eloquent than a lengthy explanation.

<sup>12</sup> In some cases authentication for Web access. Also depending on the server’s functionality, it is possible to ask users for authentication. But in this section the discussion is limited to cases where authentication is not required.

<sup>13</sup> There are many different methods besides FTP. Different organizations (universities and businesses) and Internet service providers use different systems.

<sup>14</sup> It is shareware and requires 25 dollars to use it. For educational use and tax-exempt charitable project (in the United States; handling of uses in Japan is unknown), one can apply for a free license. Under “Licensing” of the left menu on the “Fetch Softworks” page (the above-mentioned URL), you can find the link “Education/Charitable.” Apply from here. Although Fetch has a Japanese version, it is unknown whether the Education License applies to it.

Maybe you do not want to achieve a particularly dramatic effect, but you often want to include a small image in your Web page, be it a drawing you created or a digital photo you touched up.

Meanwhile, image processing software is advanced technology, and many commercial products are quite expensive. So, this guide introduces you to software which you can use free of charge. This software is called the GNU Image Manipulation Program, or GIMP.

GIMP was originally English-language software, but there is also a Japanese version as well as a translation of the manual into Japanese.

- Program  
<http://www.geocities.jp/gimproject/gimp2.0.html> (Japanese Only)
- Manual  
[http://www.geocities.jp/gimpfile/gum\\_jp/index.html](http://www.geocities.jp/gimpfile/gum_jp/index.html) (Japanese Only)

GIMP features functionality comparable to commercial software. Although some functions are a little tricky to use, free software with as many features as this is worthy of note.

Next, this guide introduces two Web applications, which do not require the installation of any software on a PC. Both applications check whether a created XHTML file is syntactically correct (valid).

- W3C Markup Validation Service  
<http://validator.w3.org/>
- Another HTML-lint gateway  
<http://openlab.ring.gr.jp/k16/htmlint/htmlint.html>

The first Web application is a validation service provided by the World Wide Web Consortium, or W3C, the organization which develops the XHTML specifications. The latter Web application points out errors and explains the reasons in Japanese, and also provides a grade. After you create your own XHTML document, check its validity using these services.

## 8.2.4 Section Summary

This section talked about useful Web publishing tools. The minimum tools required for Web publishing are a text editor and FTP software, and both come with Windows and Mac OS. However since more useful and simpler tools are available free of charge, please go ahead and use them.

In addition, although not a requirement, having an image processing program helps you extend your range of expression. The program GIMP can be used free of charge. In addition, always try to create a valid XHTML document by taking advantage of Web applications that validate your XHTML documents.

## 8.3 Starting XHTML

### 8.3.1 Markup and Tags

In this section, first we are going to create a minimal XHTML document. Actually this XHTML document is not exactly a valid XHTML document. Anyway, what we are going to do is to create a compact XHTML document which we can display on a Web browser first, and then gradually develop it into a valid XHTML document.

As mentioned earlier, XHTML is a language for describing document structure. We will first create a document which consists of headings and paragraphs. Open a text editor and type as shown below. Type all symbols and alphabetic characters using “half-width” characters<sup>15</sup>.

```
<html>
<h1>
はじめての XHTML
</h1>

<p>
これははじめての XHTML です。
</p>
</html>
```

Let us save the above file. You can save the file in any location in your file system, but choose an easy-to-find place (on the desktop, for example). Name the file “index.html” using only single-byte characters. After saving the file, exit the editor.

Next, double-click the created file to open it. If your Web browser does not open when you double-click the file, check the file name, especially its extension. Or you can start your Web browser and click “**Open**” on the “**File**” menu to open the file. Your Web browser should display a page like Figure 8.4.



Figure 8.4: Your First XHTML Document Is Displayed

Now you have created a page containing a heading and text; this means you have just learned the minimum requirements for the dissemination of information, so to speak.

If your browser does not look like Figure 8.4, start your text editor, click “**Open**” on the “**File**” menu to open the index.html file which you have just saved, and check to see if what you have entered is correct.

Let us examine the content you have entered. In XHTML, usable document elements, such as headings and paragraphs as in the above example, are predefined. In this way, we create an XHTML document by “marking up” individual elements of the actual document using predefined symbols, “tags,” which correspond to these elements.

A tag is surrounded by angle brackets “<” and “>,” for example <p>. A tag with a slash like </p> indicates the end of an element, and a tag without a slash indicates the start of the element. In the above example, you can see that the sentence “これははじめてのXHTMLです。” is enclosed with a start tag and an end tag (see Figure 8.5 on the page 124). The tag <p> indicates a “paragraph.” This means that the section enclosed with these tags is

<sup>15</sup> Single-byte alphanumeric characters are called “half-width” characters in this section. Although these expression like “half-width” and “single-byte alphanumeric characters” are not accurate, we use these expressions anyway. With a Japanese keyboard, to switch between half-width and full-width, press the “半角/全角” key which is at the upper left of the keyboard. Note that the word “XHTML” in the text does not need to be entered using half-width alphanumeric characters.

one paragraph. `<h1>` means “level 1 heading” (the highest level). Although the heading is displayed in a large font on the screen, this tag does not instruct the Web browser to display its contents in a larger font. The heading is displayed in a large font because the heading level is high. This point is explained in detail later.

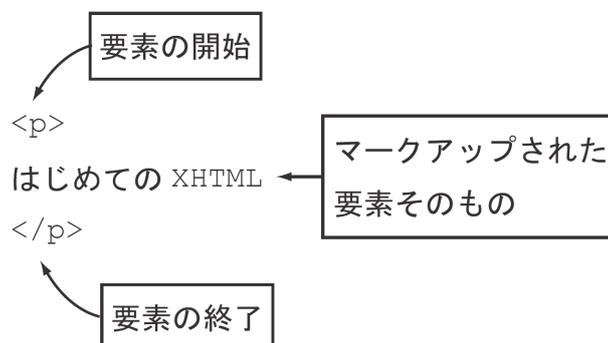


Figure 8.5: Markup

This specification of a structure or a format using tags is called “markup.” XHTML is an abbreviation of Extensible Hypertext Markup Language, and it is referred to as a “markup language” because of this feature. Do not remember XHTML as being “the language which encloses text with tags.” Tags themselves are not important, but the fact that elements are marked up with tags.

Moreover, note that the entire text is marked up by using `<html>` and `</html>` in the previous example. This means that the entire text is an html element (the formal way of describing this element is explained in detail later).

Now, usable “document elements” in XHTML are predefined as mentioned earlier. You may wonder if you can use undefined document elements. As the name of XHTML suggests, it is extensible in some ways. This point is explained later (page 144), but remember, even if XHTML is extensible, it does not mean you can define document elements and create tags randomly.

### 8.3.2 Relationship of Tags to Formatting and Document Structure

A Web browser receives a marked-up document with tags from a Web server, interprets the document, and “renders” the document. Rendering is the formatting of a document when the document is displayed on a monitor according to its structure.

Document structure and formatting are related to an extent. For example, as semantic groupings, a chapter contains sections, a section contains subsections, and a subsection contains paragraphs (see Figure 8.6 on the page 125). Chapters, sections, and subsections have respective headings, and paragraphs contain text. The character sizes used should be in the order of Chapter > Section > Subsection > Paragraph, if possible. By doing so, we can intuitively grasp the inclusive relations of texts from the character sizes used. Although document structure and formatting are essentially unrelated, an accentuated format is a “people-friendly design”<sup>16</sup>. Since this guide incorporates such formatting, please use the formatting of this guide as a reference.

<sup>16</sup> Although this is never a good practice, it is possible and not uncommon to ignore the relationship between document structure and format, and enlarge the words you want to emphasize.

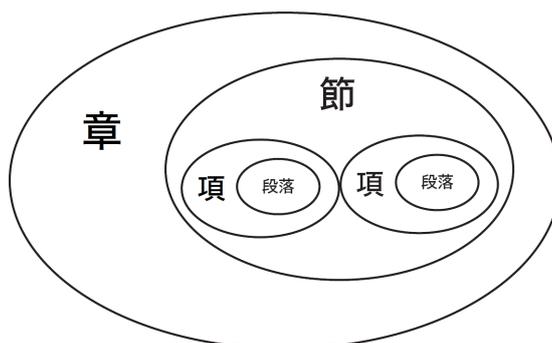


Figure 8.6: Inclusive Relationship of Chapter, Section, Subsection, and Paragraph

So, when Web browsers receive an XHTML document from a server, they often interpret and render it to display headings in larger fonts and so forth<sup>17</sup>. Furthermore, Web browsers arrange and display photographs and images called from XHTML (to be described later).

One of the reasons why XHTML relies on a client (Web browser) for interpreting and rendering by using tags is because in this way various types of clients can participate in the Web. These days Web sites can be viewed from cellular phones. Compared to PC Web browsers, many cellular phone Web browsers have various restrictions in terms of window size, number of colors, CPU, memory, and on the method of character entry. In addition, those who access the Web using a Braille display or screen reading software can receive information accurately.

For example, XHTML has tags which define headings. If we extract the elements defined as headings, we will be able to create a table of contents. This is a kind of job computers can handle well automatically, so people do not have to do it manually. A major advantage of creating information on a computer is its “reusability.” One of the objectives of the Web is to expand this reusability across the world. And this feature can be achieved because elements are expressed using tags.

An XHTML document essentially consists of this kind of document content and document structure, which are all described in the form of text. As described later, to create and edit a file in the text form, you use the above-mentioned text editor.

## 8.4 Document Elements Explained in This Chapter

In order to provide you with some perspective, XHTML 1.0 elements discussed in this guide and the tags to express these elements are summarized in Table 8-2 and 0 (p. 126), respectively. Although some of them are Web-specific or computer-specific, many of the elements have been used in typesetting on paper medium for a long time. Even if you are not familiar with the vocabulary, you will know what they are once you have a look at actual typesetting.

The document elements you can use for XHTML 1.0 are almost the same as the ones for HTML 4.01. There are many elements that have been defined for XHTML 1.0, but you do not need to learn them all (we cannot introduce all of them in this guide, either). In this section, we classify these elements based on the XHTML Basic specification<sup>18</sup> and explain only the ones we think are important.

<sup>17</sup> Rendering is completely dependent on Web browser functionality. Some Web browsers can display only text of one size. Naturally they cannot display images or other elements. Lynx and w3m are typical examples. People have different opinions regarding the utility of using such Web browsers. You should look into this yourself.

<sup>18</sup> The number of devices which can display Web pages are rapidly increasing, and include cell phones, Personal Digital Assistants, car-navigation systems, and various intelligent home appliances. XHTML Basic is the minimal set of XHTML specifications for such devices with limited processor performance, memory, and display functionality. The point of studying such XHTML is not simply so you will be able to create Web pages for cell phones. Since XHTML Basic is a basic specification for various devices, it serves as the best point of reference for XHTML study.

Table 8-2: Document Elements Discussed in This Guide

Document Structure		html, heading, title, and main body
Text	Block	heading (16 levels), paragraph, quotation, preformatted text, and contact address
	Inline	line break, citation of other documents, quotation, emphasis, stronger emphasis, and abbreviation
Hypertext		anchor
List		unordered list, ordered list, definition types, and item
Table		caption, table, row and column headers, table row, and table data
Image		image
Meta-information		meta-information

Table 8-3: Tags Corresponding to Document Elements (in Table 8-2)

Document Structure		html, head, title, body
Text	Block	h1, h2, h3, h4, h5, h6, p, blockquote, pre, address
	Inline	br, cite, q, em, strong, dfn, abbr, acronym
Hypertext		a
List		ul, ol, dl, dt, dd, li
Table		caption, table, th, tr, td
Image		img
Meta-information		meta

In addition to elements, you can specify attributes in XHTML. Some elements only have meaning when there is an attribute. The attributes you can specify for a particular element are predetermined. When we introduce an element below, we also introduce attributes as needed.

## 8.5 Text and Image Elements

### 8.5.1 Block Elements

#### Paragraph and heading

Text is the most basic document element. Text elements can be roughly divided into “block elements” and “inline elements.” These elements are described in Section 8.10 “Document Structure and Meta-information Elements” (p. 141). Just remember that “a block element is a document component” and “an inline element modifies a part of a block element.” Among the elements in Table 8-3, lists and tables are block elements and images and hypertexts are inline elements.

The XHTML document we created looks as follows now. We are going to build on this, but first of all, let us explain what this means.

```

<html>
<h1>はじめての XHTML</h1>

<p>これははじめての XHTML です.</p>
</html>

```

First, <html> indicates that this document is an HTML document. We know that the whole document is an HTML element. Actually we need more than this to have a valid HTML document, but in order to avoid making the explanation too complex, we will leave the details until later. Just put <html> and </html> at the beginning and at the end, respectively.

Next, “h” of <h1> is for “heading,” and together with “1 (one)” that follows “h” indicate this is a “level 1 heading.” Characters marked up with this tag are displayed using a Web browser’s largest font size<sup>19</sup>. You can see that on your Web browser for yourself. Large characters are used to express a large semantic grouping in document structure. The sizes of the displayed characters are in the order h1 > h2 > h3. Similar relationships hold true for level 4-6 headings.

Open the saved “index.html” file with a text editor and modify it as follows.

```

<h1>はじめての XHTML</h1>
<h2>レベル 2 の見出し</h2>
<h3>レベル 3 の見出し</h3>

<p>これははじめての XHTML です.</p>

<p>これは 2 つ目の段落です.</p>

```

There is another tag <p>. This element indicates a paragraph. Remember “p” stands for “paragraph.”

After you modify the file, open the file by double-clicking “index.html,” or start your Web browser, click “Open” on the “File” menu, and select this “index.html” to open. The file is displayed on the Web browser. If the page displayed on your Web browser does not change even after your editing of the file as above, click the “Refresh” button (see Figure 8.7).



Figure 8.7: Refresh Button

Whenever you think the Web page displayed on your Web browser may not be the latest one, try the “Refresh” button.

<sup>19</sup> There is a way to make characters larger. This method is mentioned later, but is not recommended.

If a click on the button does not do the trick, the page may be refreshed if you click the button while holding down the “Shift” or “Control” key<sup>20</sup>.

You need to pay attention to three fundamental points when marking up a document with tags.

1. Use only lower-case alphabetic characters for tags.
2. Do not overlap elements.
3. Pay attention to white space characters.

As for the first point, tags are written only using lower-case alphabetic characters. Mixing upper-case and lower-case letters was allowed before in HTML, but only lower-case letters are allowed in XHTML.

The second point means you should not do the following (since this is a “do not do this” example, you do not need to actually type it out).

```
<h1>これは見出しです  
  
<p>これは段落です。 </p>  
  
</h1>
```

Do you know which part is the problem? A heading starts, and then the second paragraph starts before the end of the heading. In this example, the start of the paragraph is followed by the end of the paragraph (</p>). This indicates the paragraph ends here. Then the heading ends. Does this mean that the paragraph resides in the heading?

This is an example of the “overlapping elements,” which are invalid in XHTML. Such overlapping elements is not only invalid with block elements here but also with inline elements to be described later.

As for the third point, characters such as spaces, tabs and line breaks in XHTML are all treated as white space characters, which basically represent demarcation between two words. Moreover, two or more consecutive white space characters are regarded as being one white space character. For example, add the following to the XHTML document you are creating.

---

<sup>20</sup> This command is called “super reload.” Super reload makes a Web browser reload the page disregarding its cache. The “cache” is described later.

```
<p>これは 4 つ目の段落です。 </p>
```

```
<p>
```

```
これは 5 つ目の段落です。
```

```
</p>
```

```
<p>
```

```
これは 6 つ目の  
段落です。
```

```
</p>
```

```
<p>
```

```
これは  
    7 つ目の  
        段落です。
```

```
</p>
```

On a browser, all of these are rendered almost identically. The difference between the fourth paragraph and the fifth paragraph is the positioning of tags. These bring about no difference at all. How about the sixth paragraph and the seventh paragraph? You may see some space at line breaks, in particular, between “の” and “段” in the sixth paragraph and between “は” and “7,” and “の” and “段” in the seventh paragraph on your Web browser. This happens because Web browsers have inherited Western language processing. This should not bother you much.

The above description is summarized as follows: tags should mark up elements properly, and two or more consecutive white space characters including a line break are regarded as being one white space character. A line break in the middle of a paragraph in the XHTML will not result in a line break when displayed on a Web browser. When a line break is required by the definition of an element, a new line starts automatically. For example, a new line starts at the end of a paragraph or a heading. A long paragraph containing multiple lines will have line breaks, too. In this case, a linefeed width (pitch) tends to be narrower than the space between paragraphs. You should try this yourself by writing a long paragraph.

Writing a computer language such as XHTML is called “coding.” When coding XHTML, you may put line breaks in places which help readability on a text editor. However, you cannot start a new line in the middle of a tag as follows.

```
<h
```

```
1>
```

```
悪い例：タグの途中で改行されている
```

```
</h
```

```
1>
```

Although it is in a sense user-friendly that you do not need to worry about white space characters, at times you want your text to display on your Web browser as you see it in your text editor. In such a case, you can use the preformatted element. This element is represented by `<pre>` and `</pre>`.

## Preformatted

Add text after the seventh paragraph as in the following example.

```
<p>  
これは  
    7つ目の  
        段落です。  
</p>  
  
<pre>  
これは  
    プリフォーマットの  
        文章です。  
</pre>
```

See the difference. Line breaks and white spaces are displayed as they are in the XHTML.

## Quotation

This element is used for displaying quoted text. Add the following example to the XHTML document you are creating.

```
<blockquote>  
<p>  
これは引用しているところ。ブロック要素なのでまとまった量の引用を表示する  
のに便利である。通常、少しインデントされて表示され、引用した文章である  
ことが分かりやすいよう表示される。  
</p>  
</blockquote>
```

You do not need to type the text exactly as it is, but its effect is easier to see if you enter more than two lines. A `blockquote` element is used for quoting block-level content. In the above example, you can see that the `blockquote` element includes another block element. In this case a paragraph is included, but it can be a heading or two or more paragraphs.

## Contact address

The last block element is a contact address. This element is used for displaying mailing addresses, email addresses, and so forth.

```

<p>
このページに関しては、次の住所まではがきか封書でご連絡ください。
</p>

<address>
〒 169-8050 新宿区戸塚町 1-104 早稲田大学
</address>

```

The above example purposely contains vague information. This element is originally used for presenting the information about the author of the Web page. Using the element, you can display the contact information concerning the document, such as the postal code, mailing address, telephone number, FAX number and email address.

Please keep in mind that you should never reveal your place of residence on the Web. You should include your email address<sup>21</sup>, but since email addresses are constantly collected from Web pages automatically and used to send unsolicited junk email, many people employ schemes such as using images to display their email address rather than text.

These are all the block elements for text introduced in this guide. You can see that most sections of nearly all documents can be expressed using these elements. You find it inconvenient if you do not use the inline elements described next, but even a simple Web page made with the elements you have learned in this section can attract many visitors if it has substantial content.

### 8.5.2 Inline Elements

Block elements are building blocks of text, and inline elements modify certain parts of block elements or give additional meaning to block elements. The tags corresponding to the inline elements discussed here are: br, cite, q, em, strong, dfn, abbr and acronym. Individual elements are described below.

#### Line break

The br element is a line break. As mentioned earlier, a line break in XHTML is not processed as a line break but as demarcation between words. Therefore, if you want to start a new line somewhere in a paragraph as it is displayed, you need to specify a line break with br.

By the way, when do you think you might want to force a line break in a paragraph? You might do so for a poem. Or you may wish to force a line break in the above-mentioned address element when you write a mailing address.

The br element is used as follows.

```

<address>
〒 169-8050<br />
新宿区戸塚町 1-104<br />
早稲田大学
</address>

```

<sup>21</sup> [Procedure for Creating Web content for Web Domains] of Waseda University states that “The Persons Responsible for Posting web documents and their contact addresses shall be indicated.” In general, the Copyright Law upholds the right of anonymous speech. However, in consideration of the issue of whether contact information should be presented to ensure responsible speech, we believe that a university, as a bastion of discourse, must present such information.

The way a slash (/) is used with the br element differs from the tags we have introduced up to now. So far the absence of a slash means the start of an element and the presence of a slash means the end of the element, and the tags enclose the relevant element in a text. However, the br element does not have such an element to enclose. That is because it has an independent meaning. When a tag itself is an element like this one, it is called an empty element. Since an empty element cannot be closed with a tag, a notation is given in such a way that a tag has both a start and an end in itself. That is why it is written as <br />.

Note that in the above example, the first two lines have the br element at the end but the last line does not. The reason is since a line break is expected at the end of the address element, a line break specified by the br element is not necessary here.

### Citation and quotation of other sources

Next, we talk about the cite element and the q element. The cite element is used for the citation of other sources.

```
<blockquote>
<p>国境の長いトンネルを抜けると雪国だった。</p>

<cite>川端 康成, 『雪国』, 新潮文庫, 1937年6月.</cite>
</blockquote>
```

One block element contains another block element in the above example, but you may start from text without any paragraphs.

Meanwhile, the q element is used when you want to quote some text within a sentence.

```
<p>
川端康成の『雪国』は、<q>国境の長いトンネルを抜けると雪国だった。</q>と
いう有名な書き出しで知られている。
</p>
```

However, usually the q element is not formatted in such a way that the quoted text can be distinguished from other text within the p element. Some Web browsers use double quotation marks (") to enclose the segment, but formatting like that in Japanese is not necessarily appropriate. As we have discussed earlier, it is important that the quoted part and the writer's work need to be clearly distinguished. Even if they are clearly distinguished as markup, you may get in trouble if they are indistinguishable in terms of formatting. It will be easier to tell if the quoted text is enclosed with 「」 (Japanese quotation marks), but these cannot be displayed well on some Web browsers.

This is one example where the functional implementation of a Web browser does not support the markup language well. There are other examples. So you should keep in mind that textual expression using XHTML may not appear the way you intended.

### Emphasis

The next elements concern emphasis. They are the em element and the strong element. You should remember em as meaning "emphasize." The question here is "why there are two elements?" They are for weaker and stronger emphasis, respectively. The strong element is stronger than the em element. These elements are usually displayed in bold. Another element which is displayed in bold but has a different meaning is the dfn element. This is an abbreviation of the word "definition," and indicates a defining instance of the enclosed term.

The following contains an example using these three elements.

```
<p>
これは<em>強調</em>ですが、こちらは<strong>より強い強調</strong>です。
定義語は<dfn>dfn 要素</dfn>で表現しますが、初出の重要な語をマークアップす
るのに利用されます。
</p>
```

These elements are all displayed in bold, but do not use them just to embolden text. These elements must be used for expressing emphasis.

### Abbreviations and acronyms

Next we cover the abbr and acronym elements. These elements are used for indicating abbreviations and acronyms, respectively. An abbreviation is a shortened word. For example, “by the way” is abbreviated to “BTW.” An acronym of “United Nations” is “UN.” An example is given below.

```
<p>
今日、授業で<abbr title="メディアネットワークセンター">メネセン</abbr>に行っ
たら、<acronym title="Waseda university InterNet Domain">WIND</acronym>
の内規を読むという宿題を出された。
</p>
```

When the above XHTML document is displayed on a Web browser which supports the abbr and acronym elements, the characters “メネセン” and “WIND” are underlined, and if you put your mouse pointer over either one, its title attributes is displayed. You should try out it yourself.

### 8.5.3 Element Attributes

“Attributes” are used with the abbr and acronym elements. These attributes have not been used with the other elements we used so far. An attribute is used for specifying the property and function of an element in detail, and is described in the start tag of the element. An element may be able to have multiple attributes. In general, an attribute is written as follows.

#### <Element Attribute 1=“Content of Attribute 1” Attribute 2=“Content of Attribute 2”>

Place a space between element and attribute 1, and between attribute 1 and attribute 2 in order to separate them. The content of an attribute is called an attribute value and is specified on the right-hand side of an equal sign. All attribute values must always be enclosed with double quotation marks (“”). In this example, in both elements an attribute called title is specified. There are many other attributes and which attributes are applicable vary by element.

For example, the above-mentioned blockquote element can have a cite attribute (which is different from a cite element).

```
<blockquote cite="http://www.waseda.jp/mnc/" title="MNCホームページ">
```

As has been just described, attributes play an important role in XML.

## 8.6 Hypertext Element

There is only one hypertext element: the `a` element. You should remember `a` as meaning “anchor.” Although there is only one hypertext element, since XHTML is a language for the describing of hypertext, this element characterizes XHTML.

This guide has not explained what hypertext is yet. In a nutshell, hypertext is “text which leads you to other related texts.” You can see it more easily if you actually play with it. Typically it is a system in which when you click an underlined character string, another page is displayed on your Web browser.

### 8.6.1 Source Anchor: Hyper-references

Let us start with an example.

```
<p>
早慶戦は<a href="http://www.waseda.jp/">早稲田大学</a>と
<a href="http://www.keio.ac.jp/">慶応大学</a>の伝統です。
</p>
```

Next, let us examine the content of the tag. The `href` attribute is specified here. This hyper-reference attribute is used for specifying a reference destination. In the above example, when you click “早稲田大学,” the top page of Waseda University appears, and when you click “慶応大学,” the top page of Keio University appears. The attribute value of the `href` attribute is a URL.

Let us add the `accesskey` attribute. For example, let us extend the above example as follows.

```
<p>
早慶戦は<a href="http://www.waseda.jp/" accesskey="1">早稲田大学</a>と
<a href="http://www.keio.ac.jp/" accesskey="2">慶応大学</a>の伝統です。
</p>
```

This attribute adds a shortcut key called an access key. In the above example, pressing the “1” key while holding down the “Alt” key while the Web page is displayed selects the link. However, to jump to the linked page, you need to press the “Enter” key once the link has been selected. Note that this action may vary depending on which Web browser is used. In the case of cell phone Web browsers, pressing the “1” key directly jumps to the linked page in many instances.

### 8.6.2 Destination Anchor: Fragments

Above, we have created a source anchor, which is a link that jumps to another document. We can also create a destination for links.

```

<h1>
<a name="midashi1">これは最初の見出しです</a>
</h1>

...

...

...

<h1>
<a name="midashi2">これは 2 番目の見出しです</a>
</h1>

```

As in the above example, you could name the headings “midashi1” and “midashi2” respectively. Once these headings are named, you can access these headings using their names. Assuming the name of the file that contains these headings is index.html, you can access the first heading using the following URL.

<http://www.example.ac.jp/index.html#midashi1>

In other words, now you can specify a section of a document directly. Such a document section is called a fragment. So far all the a elements have been applied to character strings within a paragraph. But in fact an anchor can be also set to a character string in a heading as well as to images, which will be described later. The example is shown below<sup>22</sup>.

```

<h1>
<a href="http://www.waseda.jp/">早稲田大学</a>
</h1>

<p>
<a href="http://www.waseda.jp/jp/okuma/"></a>
</p>

```

By the way, the a element is an inline element. Consequently, it is used for modifying part of the p element, a block element; that is, it is used for modifying a part of a paragraph. Therefore, the following example is incorrect (since in this example the inline element includes a block element).

```

<a href="http://www.waseda.ac.jp/">
<p>
早稲田大学
</p>
</a>

```

<sup>22</sup> This example assumes that a file called okuma.jpg exists. If no such file exists, a square with a red × will be displayed in Web browsers. But even in such cases, you can still follow the link.

## 8.7 List Elements

A list is an effective means of summarizing information in an easy-to-understand way. There are various kinds of list that can be created, such as unordered lists and ordered lists. You can also create a list using XHTML.

### 8.7.1 Unordered Lists

First, let us create an unordered list. The ul (unordered list) element is used.

```
<p>
東京六大学は次の大学から構成されています。
</p>

<ul>
  <li>早稲田大学</li>
  <li>慶応大学</li>
  <li>東京大学</li>
  <li>立教大学</li>
  <li>明治大学</li>
  <li>法政大学</li>
</ul>
```

In the above example, the li (list item) element also makes its first appearance. There are many tags in the above example, so it may look complicated. In the example, the li elements are indented to help the readability of the XHTML code, including the inclusive relationship.

### 8.7.2 Ordered Lists

The next example is an ordered list. This list is created with the ol (ordered list) element.

```
<ol>
  <li>早稲田大学</li>
  <li>慶応大学</li>
</ol>
```

All you need to do is replace `<ul>` with `<ol>`, but the result of doing so is quite different from the unordered list. If you display the rewritten XHTML document in a Web browser, you can see the difference. The ol element automatically numbers list items in sequence beginning at the top of the list. For example, if the University of Tokyo is added between Waseda University and Keio University in the above example, the displayed numbers will automatically change. Automatic numbering does not just save time; it also means you do not need to renumber the document after editing it and prevents numbering errors<sup>23</sup>.

---

<sup>23</sup> When it comes to computers, it is best to leave tasks that computers can handle to them. It is best that you try to get the maximum effect from the minimum effort.

### 8.7.3 Nested List

The `li` element can include text or block elements, such as paragraphs, as content. Moreover, it can also include another list. This is called “nesting.” The following is an example of nesting.

```
<ul>
  <li>早稲田大学
    <ul>
      <li>西早稲田</li>
      <li>戸山</li>
      <li>大久保</li>
    </ul>
  </li>
  <li>慶応大学
    <ul>
      <li>三田</li>
      <li>日吉</li>
      <li>藤沢</li>
    </ul>
  </li>
</ul>
```

In this example, you should pay attention to the correspondence of `<li>` and `</li>`. Can you tell where the closing `li` element which corresponds to the first `li` element before “早稲田大学” is? Note that a list nested in another list is always contained in one of the list items (the `li` elements) of the larger list.

You can also create a definition list. It is useful when creating glossaries. You use the `dl` (definition list) element, the `dt` (definition term) element, and the `dd` (definition definition) element to create definition lists.

```
<dl>
  <dt>早稲田大学</dt>
  <dd>東京都新宿区西早稲田 1-6-1</dd>
  <dt>慶応大学</dt>
  <dd>東京都港区三田 2-15-45</dd>
</dl>
```

## 8.8 Table Elements

A table is another convenient way for organizing and presenting information. Figure 8.8 (p. 138) shows elements that are frequently used in tables.

### 8.8.1 Two-by-two Table

Let us start from the simplest example: a table in a frame with two rows and two columns.

早稲田大学生の平均身長・体重			
	平均		備考
	身長	体重	
一年生	175	72	
二年生	178	70	
三年生	173	65	
四年生	177	68	

Figure 8.8: Table Elements

```
<table border="1">
  <tr><td>あ</td><td>い</td></tr>
  <tr><td>う</td><td>え</td></tr>
</table>
```

Let us have a look at the above example. It is a little complicated, but a long look at it reveals certain rules. First, the whole example comprises one table element. The table element has the border attribute, and 1 is specified as its attribute value. This attribute specifies the thickness of the border frame. Try specifying 2 and 3 as the attribute value of the border, and check to see how these frames differ from the frame in the case where there the value is 1. If the border attribute is not specified, no frame is displayed<sup>24</sup>.

Next, you see there are two rows. Each row consists of one tr (table row) element. These tr elements become the rows of the table. Moreover, each tr element contains two td (table data) elements. These td elements contain specific data (in this example, “あ” “い” “う” and “え”).

## 8.8.2 Row and Column Spanning

If you want to increase the number of rows, you simply increase the number of tr elements. If you want to increase the number of columns, you increase the number of td elements contained in the tr elements. Note that, however, the number of td elements in all tr elements must be the same; in other words, the number of columns must be the same. If one row has only two columns and another row has three, the table will be distorted. In order to check this point, let us take a look at spanning. This example has three rows and two columns, but since the cell of the first row spans all columns, the first row actually consists of only one column.

Now, take a look at the second line of XHTML in Example 1. The second line has only one td element, but it also has a colspan attribute, unlike the third and fourth lines which have two td elements and no attributes. As its name indicates, the colspan attribute specifies a range of columns. A single column is displayed across the width of two columns, in other words spanning two columns.

<sup>24</sup> Tables have been used for formatting in Web publishing, since they can be displayed without frames. This is not the correct way of using tables. If you use Cascading Style Sheets (CSS) (which are mentioned later), you do not need to use tables for formatting. However, CSS support differs from Web browser to Web browser, and some Web browsers display the formatting incorrectly or may even not display it at all. Therefore, some people say the use of tables for formatting is unavoidable when the design of your Web page is important to you.

表の例 1：セルの連結 (列)

```
<table border="1">
  <tr><td colspan="2">五十音</td></tr>
  <tr><td>あ</td><td>い</td></tr>
  <tr><td>う</td><td>え</td></tr>
</table>
```

五十音	
あ	い
う	え

Figure 8.9: Results of Display

表の例 2：セル連結 (行)

```
<table border="1">
  <tr><td rowspan="2">あいうえお</td><td>あ</td><td>い</td></tr>
  <tr><td>う</td><td>え</td></tr>
</table>
```

あいうえお	あ	い
	う	え

Figure 8.10: Results of Display

Next, let us consider the spanning of rows. See Example 2. The left-most column spans the first and second rows. This is an example of a cell that spans multiple rows. Note that the second line of XHTML has data for three columns but the third line has data for only two columns. In other words, both row and column spanning reduces the number of cells.

Finally, let us set up a table caption, and row and column headers. See Example 3. In the examples above, “五十音” and “あいうえお” are headers in the tables, but these headers should be distinguished from the data of other cells. Moreover, giving a proper caption to each table will help it to be easily understandable.

表の例 3：表の標題・表内見出し・セル連結

```
<table border="1">
  <caption>五十音</caption>
  <tr><td></td><th colspan="2">五十音</th></tr>
  <tr><th rowspan="2">あいうえお</th><td>あ</td><td>い</td></tr>
  <tr><td>う</td><td>え</td></tr>
</table>
```

五十音		
	五十音	
あいうえお	あ	い
	う	え

Figure 8.11: Results of Display

This example is a little complicated since it combines row and column spanning. Do not be discouraged. Let us go over one line at a time.

In the above example, the caption element has been newly added. You can place the caption element anywhere you like as long as it is placed below the table element. It is usually placed immediately below `<table>` or immediately above `</table>`. As for the heading in the table, you can see that the `th` element is used in place of a `td` element. The `th` element is displayed in boldface in many cases in order to distinguish it from the `td` element, although the way `th` elements are displayed varies from Web browser to Web browser.

## 8.9 Images

### 8.9.1 Including Images

Although images are an inline element, they are treated in almost the same way as text. Note that since images are the same as text, they must be placed in a block element, such as a paragraph, a heading or a table.

Images themselves cannot be included in XHTML files. Therefore, you store images in separate files and write XHTML that calls the image file and displays it in Web browsers during rendering. It may seem unnecessarily troublesome, but by doing this, files in various formats can be included in XHTML documents.

This section only discusses images (pictures and photographs), but you can include video clips and music into XHTML documents in the same way.

In order to display an image, you need to have an image. Whether it is a diagram, a graph, a picture or a photograph, if the image you are trying to include in an XHTML document is something you created, you will have no problem. However, in case where the image has been created by someone else, you must know about copyright. If you do not know about copyright, you therefore you cannot make the correct decision as to whether it is OK to use an image or not, so do not use it. For more information on copyright, refer to Chapter 3.

However, many works have been published on the Internet in order to allow them to be widely used by people. Many such images too are available on the Internet. If you search the Web using keywords like “free image materials,” you will be able to find various images you can use freely. This section introduces one such project involving such images, called Open Clip Art Library (<http://www.openclipart.org/>).

We have already explained that there are many types of images out there that can be included in documents (see Section 8.1.2 “File Formats” on p. 118). Image file formats that frequently used are GIF, JPEG, and PNG. GIF and PNG are often used for diagrams, graphs, and illustrations, and JPEG is often used for photographs. Here, we will use the following file.

<http://www.isc.tamu.edu/~lewing/linux/sit3-shine.7.gif>

First, let us download this file. Start a Web browser and type the above URL in the “Address” Bar (or “Location” Bar) A picture of a penguin should be displayed by the Web browser. Select “**File**” -> “**Save As...**” The “**Save As**” dialog box is displayed. Type the file name “penguin.gif” and save the file. You should save it in the same place you have saved the XHTML file you are creating at the moment. You may close the Web browser after saving the file.

As mentioned earlier, images are treated in almost the same way as text. When you want to have an image, you write the following XHTML in the place where you want to insert it.

```
<p>

</p>
```

Here, an image has been inserted in a paragraph element as an example. But you can also place an image in an existing paragraph or heading element, in a table. Moreover, note that since the img element is an empty element like the br element, a slash is added at the end of the tag.

The img element has two required attributes: the src attribute and the alt attribute. Src is an abbreviation of source and specifies the URL (URI) of the image file to be displayed. When specifying an image file which is in the same location as the XHTML file, you simply need to write the name of the image file. Alt is an abbreviation of alternative and specifies text which will be displayed as an alternative for the image for Web browsers which cannot display images, such as text browsers and screen reading browsers. The alt attribute is required from the viewpoint of accessibility.

You can specify the display size of images.

```
<p>

</p>
```

In this example, the image is displayed with a width and height of 100 pixels (100 points on a screen), each. Be careful not to specify a width and height which would change the width-height ratio of the original image. Doing so will distort the image. To view the size of the original image file, display the image on your Web browser, right-click the image, and select “**Properties.**”

Note that when you publish an XHTML document you create in this way, you will have to publish the image files with the document as well.

### 8.9.2 Copyright of Images in Web Publishing

In the case of images, you need to pay special attention to copyright. You can easily save on your PC any image which can be displayed on a Web browser, and also you can easily include such images in your Web pages. However, the unauthorized reproduction of images is a clear violation of copyright. You must not use any image unless there is an explicit statement that you are allowed to use it. Moreover, even when you use an image with permission or in a way permissible under the Copyright Law, you need to include an appropriate copyright notice and the source of the image.

In practice, some copyright issues about images are difficult to decide in light of the Copyright Law. As an example, let us consider the following case.

The src attribute of the img element contains an URL. The previous example assumed that the image file is located in the same location as the XHTML document. However, as long as the correct URL is used, there is no particular restriction as to where the image file can be placed. Therefore, for example, you can easily include an image published on the Web on your own Web page by doing the following.

```
<p>

</p>
```

In this case, it is not necessary to copy the image file and transfer it to where the XHTML file is. Technically, a Web page creator did not copy the image file in this case. The copying is actually performed by the Web browsers used by the viewers of the Web page, not by the Web page creator.

Does this mean we can use any image freely in this way? Of course not.

Then, what if you include an image in a manner which satisfies the definition of “citation” under the Copyright Law? As a matter of fact, there are only a few legal precedents regarding the citation of images in comparison to the citation of texts. As the situation now stands, no definite criteria exists. However, unless you have a strong social mission to establish a precedent yourself about whether the above example violates the copyright of others at the expense of your own precious time and money, you should avoid such activity that exists in a gray area.

In summary, you should consider the citation of an image to be practically impossible unless the image is used for criticism of the image itself. Since using a link does not violate copyright, you should include a link instead of a citation.

## 8.10 Document Structure and Meta-information Elements

Up to now, we have created only individual parts of a document, but we cannot create valid XHTML documents in this way. XHTML becomes a valid XHTML document only when information about the document (including the information that it is an XHTML document) is correctly structured as well as the document as a whole being correctly structured.

First, let us consider the document structure. The normative definition of XHTML states that an XHTML document must comprise the following four parts.

1. XML declaration
2. A line which describes the XHTML version information (DOCTYPE declaration)
3. A line which describes information about the document (Header)
4. Text (Body)

The following pages will explain each of these and what is needed for a document to be “valid XHTML.”

### 8.10.1 XML Declaration

The XML declaration declares the XML version and character encoding used in the document. Probably only a few of you can easily understand the above sentence. Let us take a look at an actual example.

```
<?xml version="1.0" encoding="Shift_JIS"?>
```

In the above example, an xml element is defined using the attributes version with a value of 1.0 and encoding with a value of “Shift\_JIS.”

An XML document is text, and people can read it by opening it with a text editor. Also the creation of an XML document always takes into account that it will be automatically processed by a computer. This XML declaration is included for such automatic processing by computer, and lets the computer know the XML version and character encoding that is to be processed.

Think of character encoding as a system to express characters inside computers. Basically, computers can handle only numbers. Characters are converted into numbers to be expressed. In other words, there are tables for the conversion between characters and numbers inside computers.

In the case of the Japanese language, there are at least four kinds of “conversion tables,” i.e. character encodings<sup>25</sup>. There are various language systems (and character systems corresponding to them) besides Japanese and character encodings corresponding to such language systems. The default XML character encodings are UTF-8 and UTF-16. So, the specification of character encoding can be omitted when a document uses the default character encoding, UTF-8 or UTF-16. Conversely, the character encoding needs to be specified when the character encoding of the document is something other than the default, UTF-8 or UTF-16.

You must be careful when you use a text editor which supports multiple character encodings, such as TeraPad. But basically if you specify nothing, either on Windows or Mac OS, your documents will be created in Shift JIS. So you just need to write in the same way as in the above example. However, character encoding should not to be left to chance. Be sure to check the character encoding when creating a document.

### 8.10.2 DOCTYPE Declaration

The DOCTYPE declaration is used for declaring the XHTML version (or HTML version) to which the XHTML conforms. See the following example.

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

<sup>25</sup> Normally, in addition to ISO-2022-JP, which is called JIS code, there are EUC, Shift JIS, UTF-8 (Unicode) and more. There is a convention of using ISO-2022-JP for email and Net News, but other than in these cases, you can use any character encoding you like. However, you should use the same character encoding within a file. In Windows or Mac OS, Shift JIS is mainly used. The fact that there are many different character encodings has led to various kinds of confusion, but the situation is gradually settling down.

The above example is divided into three lines due to space limitations, but this declaration can be written in one line separated by single-byte spaces.

This declaration can be divided into seven parts.

The first part is “<!”: this is called a markup declaration. These two characters make up one symbol. A character string following the markup declaration indicates the declaration type. “DOCTYPE” here indicates that it is a document type declaration. Since putting any character between a markup declaration and a document type declaration is prohibited, you need to write these two without a space, like this: “<!DOCTYPE.”

The next “html” indicates the XML root element. XML is always written using a tree structure, and the root element is described here. This document structure is discussed later.

“PUBLIC” is an external identifier. This specifies a reference to a document called a public identifier, which is explained below. Instead of specifying an external identifier and a public identifier, you can describe at length document rules called Document Type Declarations (DTD, containing information about which tags are used in XHTML), but usually they are described separately.

The information following “PUBLIC” that is enclosed with double quotation marks is a public identifier. This public identifier consists of an owner identifier (–//W3C), a delimiter (//) and a text identifier (DTD XHTML 1.0 Transitional//EN).

The owner identifier indicates that the owner of this external document is W3C (World Wide Web Consortium). The text identifier further consists of a public text class (DTD), a public text description (XHTML 1.0 Transitional), a delimiter (//), and a public text language (EN). The public text description indicates the version and type of XHTML.

The public text language at the end indicates the language the public text (to be explained next) is written in, not the language of the file this DOCTYPE declaration is written in.

Next, the system identifier is described. The system identifier describes a URL which points to the location of a DTD. Finally, the DOCTYPE declaration is closed by “>” which is a delimiter terminating the markup declaration.

Note that you need to use different DOCTYPE declarations depending on the version of XHTML you are going to write in. You do not need to memorize the above DOCTYPE example, but always keep in mind that you need to have a good understanding of the specification of the document you are about to create and to write the correct DOCTYPE.

The DOCTYPE is information used by software. You need to write the DOCTYPE absolutely correctly, word for word, including the correct usage of upper- and lower-case letters.

### 8.10.3 Head Element

The head element comes after the XML and DOCTYPE declarations and provides information about the document itself, including its title, author, and so forth. In other words, the head element is information about information; therefore, it can also be called “meta-information”<sup>26</sup>.

The head element can contain as many pieces of meta-information as you would like. This section introduces you to required and general meta-information.

---

<sup>26</sup> The meta-information in the head element is mainly used by Web browsers and search engines. That is, it can be assumed that meta-information is used for parsing and analysis by a computer. Some types of meta-information are provided in a document text in the same way as the address element is.

```

<head>
  <title>文書の題名</title>
  <meta http-equiv="content-type" content="text/html; charset=Shift_JIS"/>
  <meta name="author" content="著者の氏名" />
  <meta name="keywords" content="キーワード" />
  <meta name="description" content="ページの概略" />
</head>

```

In the above example, five elements are described between <head> and </head>.

As the title element specifies the title of the document, it must be included in the head element and cannot be omitted. The title element is displayed on the title bar of a Web browser (see Figure 8.12). Moreover, it is usually the title element that is displayed when this page is added to the Favorites list. There are a number of reasons why the title element is so important, but we can safely say that the title of a document is the most important piece of meta-information.



Figure 8.12: Title Bar of Browser with Title Element Displayed

The other four elements are all meta elements. The first of these meta elements indicates that the document type is text, it is written in HTML, and its character encoding is Shift JIS. You must include this meta element in the head element together with the title element.

All other elements indicate the name and content attributes. The name attribute specifies the type of meta-information, and the content attribute specifies the content. Other frequently used attributes are the author, keywords, and a description.

All of the elements in the above example are meta elements except for the title element. The head element may also contain the link element and the script element.

#### 8.10.4 Html Element and Body Element

Finally, this section introduces the html element and the body element and then describes the overall structure of an XHTML file.

The html element was introduced as being the root element in the section entitled DOCTYPE Declaration (see p. 142). Before explaining about the root element itself, let us explain the way it is written.

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">

```

The html element has three attributes. Let us begin our explanation by looking at the second and third of these. Both of these attributes are used to specify the language information and so define the language used for the XHTML document. In the above example, they specify Japanese (ja), a natural language, as the language being used. The languages that are available for specification are defined in ISO 639-1, which is a list of language codes that represent the languages specified by the International Organization for Standardization (ISO). Japanese, for example, is represented by “ja.”

It may seem a little strange that the above example has two attributes that specify the language. This is a product of the dual nature of XHTML 1.0; that is, XHTML 1.0 is HTML expressed by XML. Since the expression `xml:lang="ja"` is used in XML and the expression `lang="ja"` is used in HTML, XHTML 1.0 recommends the use of both.

Let us look now at the first attribute. This attribute, “xmlns,” specifies the namespace. As we have already learned, XHTML uses predefined elements and vocabulary (tags) that correspond to these elements to mark up a document. In other words, XHTML is restricted by its use of predefined elements and so lacks extensibility<sup>27</sup>.

To compensate for this, XML allows the use of multiple XML languages (XHTML is one of them). Let us look at an example of this. Web publishing is performed by XHTML, but the document is used for analyzing the business performance of a company and its financial statements are written in Extensible Business Reporting Language (XBRL). Not only that, but the MathML is used to describe the formulas used in the analysis, SVG (Scalable Vector Graphics) is used to draw a graph of analysis, and the newspaper articles cited as reference data are described using NewsML. In other words, one document contains many documents written in many different languages (though all of them are XML).

The problem is that the vocabulary (tags) becomes confusing. For example, when we looked at the title element, we learned that XHTML uses this to define the title of a document. In another XML, however, a title may mean the managerial position of a person or the championship title of a sporting event. To resolve this issue, we use namespaces. Namespaces are distinguished by their URLs in order to guarantee uniqueness. These URLs generally only have a symbolic meaning. In practice, however, we can expect to be able to acquire some information about the namespace when we access a URL.

Now let us return to the `html` element and look at what we mean when we say that the `html` element is the root element. What this means is that every XHTML document contains one `html` element<sup>28</sup>, which contains all the other elements. When we draw a hierarchical relationship such as a tree diagram, the `html` element is drawn at the bottom of the diagram (or at the top of the pyramid if the tree diagram is rotated 180 degrees). Therefore, an object at the apex of such a hierarchical structure is called the “root.” For example, because “C:Y” is typically the uppermost layer of the directory structure in the Windows file system, this is called the “root directory.”

The `html` element is structured to contain the head element, which we have already discussed, and the body element. The body element contains the actual content. All the elements to be contained in the paragraph and heading elements, which were explained earlier, must be described in the body element. To give you an idea of what needs to be described and what order it needs to be presented in, let us now look at a simple example of valid XHTML that includes other XML declarations and DOCTYPE declarations.

---

<sup>27</sup> Plug-ins are often used to acquire this extensibility. Adobe Flash and Apple’s QuickTime are well-known examples of such plug-ins. However, both Flash and QuickTime are essentially proprietary standards, and being unable to view certain content unless you have Flash Player or another such browser plug-in can be a problem in itself.

<sup>28</sup> Multiple `html` elements never exist side by side. No `html` element is contained in another `html` element. There can be only one `html` element per XHTML document, and the `html` element must include all other elements.

```

<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<meta http-equiv="content-type" content="text/html; charset=Shift_JIS"/>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">

<head>
  <title>文書の題名</title>
</head>

<body>

<h1>見出し</h1>
<p>段落</p>

</body>

</html>

```

Figure 8.13 illustrates the structure of the above XHTML document. Check the positional and inclusion relationship that exists among the various individual declarations and elements. You may want to try drawing a tree diagram with the html element at the top for yourself.

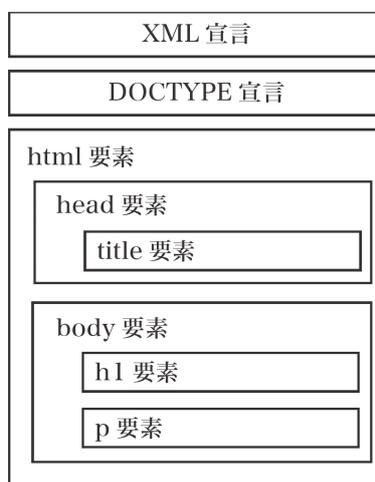


Figure 8.13: XHTML Structure

## 8.11 Fine Tuning Format

### 8.11.1 How to Write Formatting Information

Making a structured document that is a valid XHTML document is not the end of our work, however. Appearance is also important in this world, and it is a fact that when given a choice between two documents with identical content, most people will choose the more attractive looking one.

However, XHTML documents such as the one you have created should only describe the document structure, and you should not include formatting information in them. The usual way to incorporate formatting information in an XHTML document is to provide links to external files from which information is read in. An external file which contains such formatting information is called a “style sheet.” “Style” quite literally means format. For example, a style sheet may contain such information as “h1 element is in 18 point Gothic font and center justified.”

However, since creating a style sheet from scratch is not easy, you usually “overlap” (cascade) your style sheet with the default style sheet of your browser. For example, you may partially overwrite the style sheet with a rule such as “h1 is 24 point.”

Since these style sheets can cascade, they are called Cascading Style Sheets (CSS).

There are several reasons why formatting information is described using CSS rather than by directly describing it in XHTML. Such reasons include the following.

- XHTML documents can be kept simple and concise.
- The created design can be reused.
- Design changes can be applied to all documents at once.
- Document creation and design works can be kept separate.

### 8.11.2 Creating and Specifying CSS in XHTML

Two steps must be performed in preparing CSS: the first step is to create a CSS file and the second is to specify it in XHTML and read it in. First, let us create a CSS file.

The following describes a style.

```
p.strong { color: #ff0000; }
```

Here, p.strong is called a selector. In this example, it specifies that of all the p elements in this XHTML, those with a class defined as “strong” should be colored #ff0000. Code such as “{ color : #ff0000; }” is called a rule set, the color is a property, and #ff0000 is its value. A variety of objects can be specified as the properties and values. Also multiple properties can be specified at the same time.

Each color is broken down into three colors—Red, Green, and Blue (RGB)—and these are individually expressed in terms of 256 levels of color depth in hexadecimal notation from 00 to FF. The color specified in the above example is the deepest shade of red.

The following is an XHTML segment that corresponds to the above example.

```
<p class="strong">
  ここは強調されます。
</p>
```

If a selector is specified only with an element (i.e. a class such as “strong” is not specified) as in the following example, the rule set will be applied to all the p elements.

```
p { color: #ff0000; }
```

On the other hand, if a selector is specified only with a class (i.e. an element such as “p” is not specified), the rule set will be applied to all elements to which the class is specified.

```
.strong { color: #ff0000; }
```

```
<h1 class="strong">ここは強調表示されます</h1>
<p class="strong">ここも強調表示されます</p>
```

A class selector can also be collectively applied to a block element or an inline element.

```
<div class="strong">
<h1>
これはブロック要素です
</h1>
</div>
<p>段落中に<span class="strong">インライン</span>でクラス指定</p>
```

A property can specify a wide range of things, including the font, the color and background color of the font, the height and width, the border, and the text layout. There are also many different values that correspond to these properties. A detailed list of these values can be found only on the W3C Web page (in English), but since many Web sites provide an explanation of CSS and some Web sites even provide free CSS templates, you may want to search for these elsewhere.

Finally, you create a CSS file with the name “style.css” and then specify the file in an XHTML file. Specify this in the head element in the XHTML as follows.

```
<head>
  <title>文書の題名</title>
  <link rel="stylesheet" href="style.css" type="text/css" />
</head>
```

Keep in mind that when you place the XHTML file on your Web site, you need to place this style.css file with it (in the above example, the file should be in the same directory).

It has even become common practice to use different CSS for different Web browsers. When you use CSS, be aware that the display may vary depending on the Web browsers.

## 8.12 Discussion

Finally, let us discuss why a system like the Web was developed.

The axiom that knowledge does not emerge from nothing is not limited to just the written word. Without exception, knowledge has to have emerged from somewhere. To avoid repeating unnecessary arguments, we must study the accumulated knowledge of our predecessors. It is also necessary to clarify the knowledge on which your study is based because this clarifies the positioning of your study within a series of discussions.

There is little point in publishing your thoughts if all you are doing is simply self-righteously describing your personal thoughts on a matter. If you are writing only for your own benefit, that is a different matter, but if other people will be reading your work, you are expected to indicate what you have referenced, what other arguments you may have referred to in your developing your own argument, as well as how much and what kind of materials are available to your readers should they wish to know more about the issue.

Given this, it should be clear that a bibliography is an essential part of any document. Consequently, academic papers place great emphasis on the bibliography. In the academic world, the importance of a paper is judged according to “the number of times it is cited in other papers.” Simply put, this indicates how much practical help the paper has provided in other people’s studies.

This is the philosophy behind the use of hyperlinks, which is a fundamental concept of the Web. This use may not sound particularly interesting, but hyperlinks do have another important use as well: they are “what makes it easy for Web pages to be found.”

The Internet is like a road without traffic signs. A Web browser can be operated intuitively, but it is not so easy for novice users to get to the information that they need. This is why we use search engines.

A search engine called Google has established a favorable reputation. This search engine is characterized by its simple screen layout, its speedy search function, and the large number of search results produced when a key word or phrase is searched for; in other words, the search engine checks a wide range of Web pages.

Unfortunately, however, even using a search engine can often produce so many search results that you cannot find the information you are looking for. The main reason why Google enjoys its good reputation is its on-target way of displaying search results; that is, the search results are displayed in order of relevance so users often get to the information they are looking for on the first attempt (or so they think). One of the technologies used for this type of search is a ranking mechanism called PageRank, which is an index that indicates the significance of a certain site.

Put simply, this computation method is based on the idea that “a Web page to which many good Web pages are linked must also be a good Web page”([4])<sup>29</sup>. The two founders of Google received master’s degrees from Stanford University, and the concept behind Google is basically the same as that behind the way papers are evaluated in the academic world, as described above. It is safe to say that, be it on paper or on the Web, a document’s value can be increased to beyond the intrinsic value of the information it contains if appropriate cross-references are provided.

Meanwhile, a tremendous amount of effort is required to provide references in a paper document. It could take as much as a whole day to search a library for the books referenced in a substantial paper from its bibliography. Hyperlinks ease the burden involved in referencing a bibliography.

It is important to remember that providing effective links in a document is the responsibility of the author. You should not hesitate to provide a link if one is required because it would direct the reader to material that is of relevance to your work. Some Web sites require you to ask for permission before creating a link to their Web pages<sup>30</sup>. If this is the case, you should ask for permission to provide a link to the relevant Web page as a matter of courtesy, but you are not obliged to ask for and receive permission to use a link in light of any laws, guidelines, manners, ethics or any other value judgments. This cuts both ways, however, as you cannot prohibit someone from creating a link to your document once it is published on the Web or force them to ask for your permission.

---

<sup>29</sup> This document is written in such a way that liberal arts students should be able to understand its content. Increasing the chances of a Web site being found by search engines (meaning that when you search for a certain key word or phrase, the Web site appears early in the list of search results) is called Search Engine Optimization (SEO). Understanding the logic behind a search engine is the first step toward achieving SEO. For those who are interested in engaging in Web publishing in the future, this document will serve as a valuable resource. It is highly recommended.



## Related Materials

- [1] 前野譲二. Linux 演習. オーム社, 東京, 2005.
- [2] S. Hambridge. Netiquette guidelines. <http://www.ietf.org/rfc/rfc1855.txt>, 1995. 日本語訳 <http://www.cgh.ed.jp/netiquette/rfc1855j.html>.
- [3] Bruce et al Alberts. Molecular biology of the cell. Garland Science, New York, 2002.
- [4] 馬場肇. Google の秘密- pagerank 徹底解説. <http://www.kusastro.kyotou.ac.jp/%7Ebaba/wais/pagerank.html>, 2001.
- [5] American Psychological Association. APA 論文作成マニュアル. 医学書院, 東京, 2004.
- [6] Umberto Eco. 論文作法: 調査・研究・執筆の技術と手順. 而立書房, 東京, 1991.
- [7] Ron Fry. アメリカ式論文の書き方. 東京図書, 東京, 1994.
- [8] Darrell Huff. 統計でウソをつく法: 数式を使わない統計学入門. 講談社, 東京, 1968.
- [9] 中尾浩. Windows95 版人文系論文作法. 夏目書房, 東京, 1998.
- [10] 中尾浩. マッキントッシュによる人文系論文作法. 夏目書房, 東京, 1995.
- [11] 中島利勝. 知的な科学・技術文章の書き方: 実験レポート作成から学術論文構築まで. コロナ社, 東京, 1996.
- [12] 井出翁. レポート作成法: インターネット時代の情報の探し方. 日外アソシエーツ, 東京, 2003.
- [13] 伊丹敬之. 創造的論文の書き方. 有斐閣, 東京, 2001.
- [14] 伊藤義之. はじめてのレポート: レポート作成のための 55 のステップ. 嵯峨野書院, 京都, 2003.
- [15] 佐藤望. アカデミック・スキルズ: 大学生のための知的技法入門. 慶應義塾大学出版会, 東京, 2006.
- [16] 保坂弘司. レポート・小論文・卒論の書き方. 講談社, 東京, 1978.
- [17] 吉田健正. 大学生と大学院生のためのレポート・論文の書き方. ナカニシヤ出版, 京都, 第 2 版, 2004.
- [18] 君島浩. 日本語発表作法: 系統的トレーニング. 日科技連出版社, 東京, 2001.
- [19] 君島浩. 日本語作文作法: 系統的トレーニング. 日科技連出版社, 東京, 2000.
- [20] 奥村晴彦. LaTeX2 $\epsilon$  美文書作成入門. 技術評論社, 東京, 改訂第 4 版, 2006.
- [21] 小林康夫. 新・知の技法. 東京大学出版会, 東京, 1998.
- [22] 小林康夫. 知の論理. 東京大学出版会, 東京, 1995.

- [23] 小林康夫. 知の技法: 東京大学教養学部「基礎演習」テキスト. 東京大学出版会, 東京, 1994.
- [24] 小笠原喜康. 議論のウソ. 講談社, 東京, 2005.
- [25] 小笠原喜康. インターネット完全活用編: 大学生のためのレポート・論文術. 講談社, 東京, 2003.
- [26] 小笠原喜康. 大学生のためのレポート・論文術. 講談社, 東京, 2002.
- [27] 慶應義塾大学. 卒業論文の手引. 慶應義塾大学出版会, 東京, 新版新装版, 2003.
- [28] 戸田山和久. 論文の教室: レポートから卒論まで. 日本放送出版協会, 東京, 2002.
- [29] 斉藤孝. 学術論文の技法. 日本エディタースクール出版部, 東京, 新訂版, 2005.
- [30] 斉藤孝. 学術論文の技法. 日本エディタースクール出版部, 東京, 第2版, 1998.
- [31] 新堀聡. 評価される博士・修士・卒業論文の書き方・考え方. 同文館出版, 東京, 2002.
- [32] 日本エディタースクール. 日本語表記ルールブック. 日本エディタースクール出版部, 東京, 2005.
- [33] 日本エディタースクール. 原稿編集ルールブック: 原稿整理のポイント. 日本エディタースクール出版部, 東京, 2005.
- [34] 日本エディタースクール. 文字の組方ルールブック. 日本エディタースクール出版部, 東京, 2001.
- [35] 日本エディタースクール. 校正記号の使い方: タテ組・ヨコ組・欧文組. 日本エディタースクール出版部, 東京, 1999.
- [36] 日本科学技術情報センター. SIST ハンドブック: 科学技術情報流通技術基準, 1989-.
- [37] 早稲田大学出版部. 卒論・ゼミ論の書き方. 早稲田大学出版部, 東京, 第2版, 2002.
- [38] 木下是雄. レポートの組み立て方. 筑摩書房, 東京, 1994.
- [39] 木下是雄. 理科系の作文技術. 中央公論社, 東京, 1981.
- [40] 木下長宏. 「名文」に学ぶ表現作法続・大学生のためのレポート・小論文の書きかた. 明石書店, 東京, 2005.
- [41] 木下長宏. 大学生のためのレポート・小論文の書きかた. 明石書店, 東京, 2000.
- [42] 末武国弘. 科学論文をどう書くか: 口頭発表の仕方まで. 講談社, 東京, 1981.
- [43] 杉原厚吉. どう書くか: 理科系のための論文作法. 共立出版, 東京, 2001.
- [44] 東京大学教養学部. 人文・社会科学の統計学. 東京大学出版会, 東京, 1994.
- [45] 東京大学教養学部. 自然科学の統計学. 東京大学出版会, 東京, 1992.
- [46] 東京大学教養学部. 統計学入門. 東京大学出版会, 東京, 1991.
- [47] 東郷雄二. 東郷式文科系必修研究生活術. 夏目書房, 東京, 2000.
- [48] 桜井雅夫. レポート・論文の書き方: 上級. 慶應義塾大学出版会, 東京, 改訂版, 2003.
- [49] 水島洋. 科学論文がスラスラ書ける! パソコンのやさしい使い方: 図表・グラフ作成から統計解析, 画像処理, 電子投稿までこの1冊ですぐできる! 羊土社, 東京, 2005.
- [50] 沢田昭夫. 論文の書き方. 講談社, 東京, 1977.

- [51] 河野哲也. レポート・論文の書き方入門. 慶應義塾大学出版会, 東京, 第3版, 2002.
- [52] 渡辺徹. 学生・研究者・技術者のための LATEX を用いた論文作成術. プレアデス出版, 大阪, 2006.
- [53] 為田英一郎. 文章作法入門. ナカニシヤ出版, 京都, 2004.
- [54] 石原千秋. 大学生の論文執筆法. 筑摩書房, 東京, 2006.
- [55] 祝部大輔. 科学論文作成テクニック. ビー・エヌ・エヌ, 東京, 1992.
- [56] 花井等. 論文の書き方マニュアル: ステップ式リサーチ戦略のすすめ. 有斐閣, 東京, 1997.
- [57] 藤田哲也. 大学基礎講座: 充実した大学生活をおくるために. 北大路書房, 京都, 改増版, 2006.
- [58] 藤田哲也. 大学基礎講座: これから大学で学ぶ人におくる「大学では教えてくれないこと」. 北大路書房, 京都, 2002.
- [59] 讃岐美智義. デジタル文献整理術: 最新 EndNote 活用ガイド. 克誠堂出版, 東京, 改訂第2版, 2005.
- [60] 讃岐美智義. 研究者のための文献管理 PC ソリューション: PubMed. 秀潤社, 東京, 2005.
- [61] 阪田せい子. だれも教えなかった論文・レポートの書き方. 綜合法令出版, 東京, 1998.
- [62] 芝野耕司. JIS 漢字字典. 日本規格協会, 2002.
- [63] Stanton McCandlish. Eff's top 12 ways to protect your online privacy.  
[http://www.eff.org/Privacy/eff\\_privacy\\_top\\_12.php](http://www.eff.org/Privacy/eff_privacy_top_12.php), April 2002.
- [64] W3C. Web content accessibility guidelines 1.0. <http://www.w3.org/TR/1999/WAI-WebCONTENT-19990505>, 1999. 日本語訳 <http://www.zspc.com/documents/wcag10/>.
- [65] W3C. Web content accessibility guidelines 2.0. <http://www.w3.org/TR/2004/WD-WCAG20-20040311/>, 2004. 日本語訳 <http://alfasado.net/WD-WCAG20-20040311-JA/>.
- [66] 富士通株式会社. 富士通ウェブ・アクセシビリティ指針 2.0 版.  
<http://jp.fujitsu.com/webaccessibility/v2/>, 2004.
- [67] Microsoft Corporation. Microsoft アクセシビリティホーム.  
<http://www.microsoft.com/japan/enable/default.msp>, 2005.
- [68] ウェブアクセシビリティの情報提供コーナー: みんなのウェブ.  
<http://www2.nict.go.jp/ts/barrierfree/accessibility/>, 2003.
- [69] W3C. Xhtml basic. <http://www.w3.org/TR/xhtml-basic/>, 2000.
- [70] W3C. Xhtml 1.0: The extensible hypertext markup language. <http://www.w3.org/TR/2000/RECxhtml1-20000126>, 2000.
- [71] W3C. Xhtml 1.1 - module-based xhtml. <http://www.w3.org/TR/xhtml11/>, 2001.
- [72] W3C. Xhtml 2.0 w3c working draft 22 july 2004. <http://www.w3.org/TR/2004/WD-xhtml2-20040722/>, 2004.
- [73] W3C. Html 4.01 specification. <http://www.w3.org/TR/REC-html40/>, 1999.
- [74] 早稲田大学メディアネットワークセンタ.

- [75] Rajeev Motwani Terry Winograd Lawrence Page, Sergey Brin. The pagerank citation ranking: Bringing order to the web. <http://www-db.stanford.edu/%7Ebackrub/pageranksub.ps>, 1998.
- [76] 山名早人, 近藤秀和. 解説 : サーチエンジン google. 情報処理, 42 巻 8 号, pp. 775-780.
- [77] ヤコブニールセン. ユーザビリティエンジニアリング原論- ユーザーのためのインタフェースデザイン. 東京電機大学出版局, 2002.
- [78] ユーザ工学入門 - 使い勝手を考える・ISO13407 への具体的アプローチ. 共立出版, 1999.
- [79] 警察庁生活安全局少年課. 少年非行等の概要 (平成 18 年 1 月~12 月). <http://www.npa.go.jp/safetylife/syonen34/20070215.pdf>, February 2007.

Academic Literacy (2009 edition)

---

Published on April 1, 2009

Editors/Authors: Joji Maeno and Noriaki Kusumoto  
Planning: Waseda University Media Network Center  
Publisher: Kenji Horiguchi  
Publishing Office: Waseda University Press  
<http://www.waseda-up.co.jp>  
ISBN 978-4-657-09308-0

---

The copyright of this guide belongs to Joji Maeno, Noriaki Kusumoto, and the Waseda University Media Network Center. Please refer to the section entitled License (the next page of the cover page) for more information.

©2005-2009 by Joji Maeno, Noriaki Kusumoto and Media Network Center, Waseda University.

**Printed in Japan**