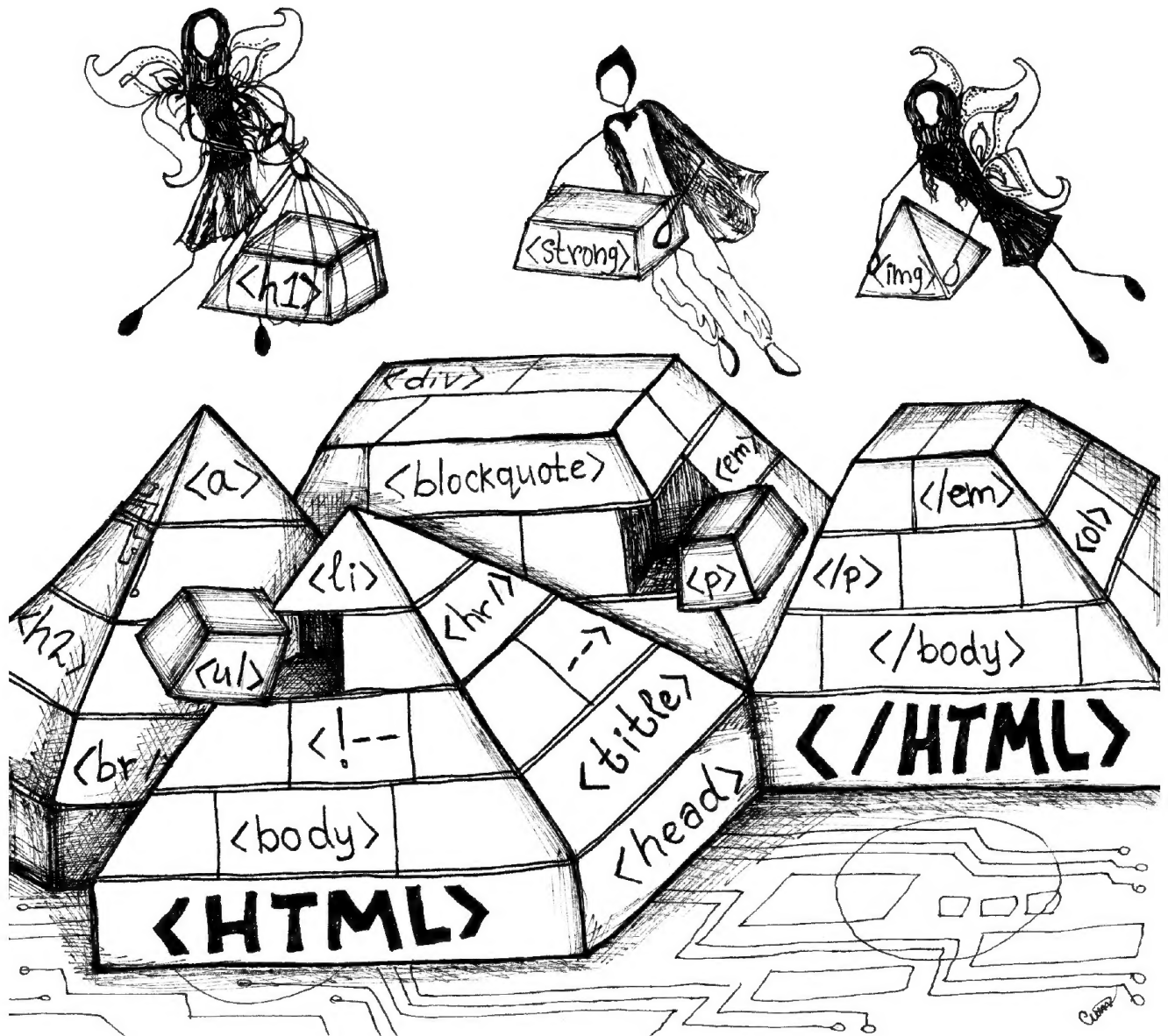# Chapter 2 HTML Basics

Introduction

We'll begin our study of web programming by learning the core language for writing web pages: Hypertext Markup Language (HTML). HTML describes the contents of your page, such as headings, paragraphs, images, and lists.

The version of HTML we'll learn is the latest and most standard, called XHTML. The pages you'll write will work in any modern browser.

This chapter doesn't attempt to provide a complete list of HTML tags and attributes. Certain aspects of the language, such as forms and tables, are left for later chapters that focus on those elements.

Each of our chapters will include a larger example called a "case study" that we will develop throughout or at the end of the chapter. This chapter's case study will be a recurring example of a blog page about travels to various far-away lands. As we learn new concepts in this chapter, we'll apply many of them to improving this page.

2.1 Basic HTML

In this section we'll discuss the basics of HTML and web pages, as well as a brief history of the language and how it came to be the way it is today. In following sections we'll dive into the details of HTML syntax for creating complex web pages.

2.1.1 History

Since its creation in 1991 by Tim Berners-Lee, one of the founding fathers of the Internet, HTML has been the dominant language for creating web pages. HTML is a language consisting of text content surrounded by markings that specify the meaning of the content. As with many languages, HTML has gone through different versions and standardization processes over the years:

- 1993: Initial official proposed description of HTML submitted to the IETF standards group.
- 1995: HTML 2 becomes an official standard language by a publication called RFC 1866.
- 1996-97: HTML 3.2 standardizes various features including forms, tables, image maps, and internationalization.
- 1997: HTML 4 is proposed by W3C standards body, adding style sheets, scripting, frames, embedding objects, internationalization, and accessibility for disabilities.
- HTML 4.01, the last major version of the language, is published in 1999 by W3C. A majority of the pages on the web today still use HTML 4.01 as their stated language.
- 2000-01: XHTML, a more standardized offshoot of HTML based on a language named XML, is proposed by W3C.

The language has several major goals that are reflected in the evolution made to it over the years. Each new version has endeavored to: add new features; improve interoperability, make HTML com-

patible with all major browsers, platforms, and devices; and be accessible to all kinds of users, including those with disabilities affecting sight and hearing.

In these chapters we will focus on *XHTML*, a more recently created dialect of HTML. XHTML is a stricter language than HTML, and its syntax is more regular and standardized. XHTML has a more clearly refined set of goals than past versions of HTML. While older HTML was a mixed language used to describe a document's content, appearance, and behavior, XHTML chooses to focus solely on describing the document's content and structure. The task of describing exactly how the document should look and how it should behave are handled by other languages that interact with XHTML.

A well-written web page has the following division of responsibilities:

- HTML describes the content of the document.
- Style Sheets (written in a language named CSS) describe the appearance of the document.
- Scripts (written in languages such as JavaScript) describe the behavior of the document.

We will learn the HTML syntax in detail throughout this chapter and later chapters. Example 2.1 shows an initial example of a complete HTML document. Figure 2.1 shows the output when you save the above code into a file called wonderful_world.html and open it in the Firefox browser.

In general it does not matter what name you give to your HTML file, but one particular file name is worth mentioning. Most web servers consider the filename index.html (and similar names such as index.php, index.jsp, etc.) to represent the main page for a particular web site. So if you omit the file name from a URL, index.html is assumed to be the filename. This is useful because, for example, you can tell people your web site's URL is http://www.example.com/, and when they visit that URL, they'll actually be shown the contents of http://www.example.com/index.html.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>What a Wonderful World</title>
  </head>

  <body>
    <h1>My first web page</h1>

    <p>
      A friend told me once that he thought the www
      in URLs stood for what a wonderful world
      (I think it does!).
    </p>
  </body>
</html>
```

Example 2.1 Basic web page wonderful_world.html

**Figure 2.1 Basic web page appearance in Firefox browser**

HTML is a rich and complex language that can describe not only standard text documents but also tables of data, forms for sending information to a server (such as for purchasing items from an online store), complex images and multimedia, interactive games, and more. Because the language is so rich, we will study it in layers, learning more of its features in several subsequent chapters. In this chapter, we'll learn the basics of structuring text content and images.

> **element**
>
> A piece of HTML markup that surrounds text content and describes its meaning in the page.

An HTML document is a text file named with an `.html` extension. It contains text content and HTML markup to tell the browser how that content should be structured. The *content* is information that you would like the user to see. The *markup* consists of *tags* that describe the content and tell the browser how to structure and display it.

Tags consist of a lowercase tag name surrounded by angle brackets. For example, a paragraph is represented by the `<p>` tag. Most tags come in pairs, with a opening tag (for example, `<p>`) followed by some text, followed by a closing tag (for example, `</p>`). A pair of HTML tags and their enclosed content are collectively called an *element*. Figure 2.2 summarizes the basic syntax of an HTML element.
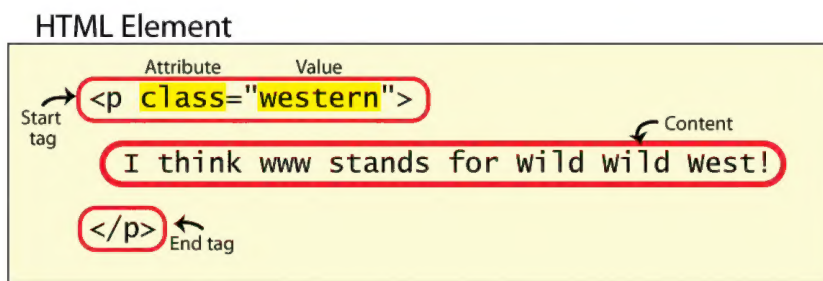
HTML Element

Figure 2.2 A paragraph element

2.1.2 Page Structure

The basic syntax of an XHTML page is shown in Example 2.2. Every page we write in this textbook will have this same basic structure. When we show code examples later in the chapter and elsewhere in the book, we will show just a subset of the page, only the relevant contents of the body.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>page title</title>
    other resources or information about the page
  </head>

  <body>
    page content
  </body>
</html>
```

Example 2.2 Syntax template for a basic XHTML page

The first three lines are the document type definition and opening `html` tag. The document type definition is a declaration that our page is written using XHTML v1.1 syntax (along with the URL to the XHTML specification); the opening `html` tag specifies the beginning of the page. We aren't going to examine the contents of these lines in detail. The main thing you should know is that these lines are required if we want our page to comply with web standards. Therefore these exact 3 lines will appear at the top of every web page we write in this textbook, so you can copy them and paste them into the pages you write.

The rest of the page consists of two major sections, the header and body. The header, represented by the `head` tag, contains general information about the page. The body, represented by the `body` tag, holds the content to display.

Header information is used by the browser but not displayed on the page. The most common element in the header is `title`, which specifies the title text to be shown in the browser's top bar. The header also includes any CSS style sheets or JavaScript code to attach to the page. (We'll learn about those later.)

2.1.3 Block Elements

There are two types of elements: block elements and inline elements. A *block element*, such as a paragraph or bulleted list, generally represents a significant element of the page and can contain a large amount of content spanning many lines. A block element can contain other block and inline elements inside it, called nested elements; for example, a paragraph can contain a link. The browser displays each block element with a line

> **block element**
>
> A significant item on a web page (paragraph, list, etc.) that occupies a rectangular block of space on the page.

break and vertical margins above and below it. In the following sections, we'll show several block elements you can place inside the page's `body` section.

Paragraphs (p)

Element	p
Description	Paragraph of text (block)
Syntax	`<p>content</p>`

Much of the content of web pages is organized into paragraphs. Each paragraph begins with the `<p>` tag and ends with the `</p>` tag. Each `p` (like all block elements) is displayed on its own line with a vertical margin above and below it.