Web Design Principles
5th Edition

Chapter One
Writing HTML for the Modern Web

Objectives

In this chapter, you will learn about:

- Creating Web pages with HTML
- The history of HTML
- Working with HTML5
- Choosing an HTML editor
- Using good coding practices

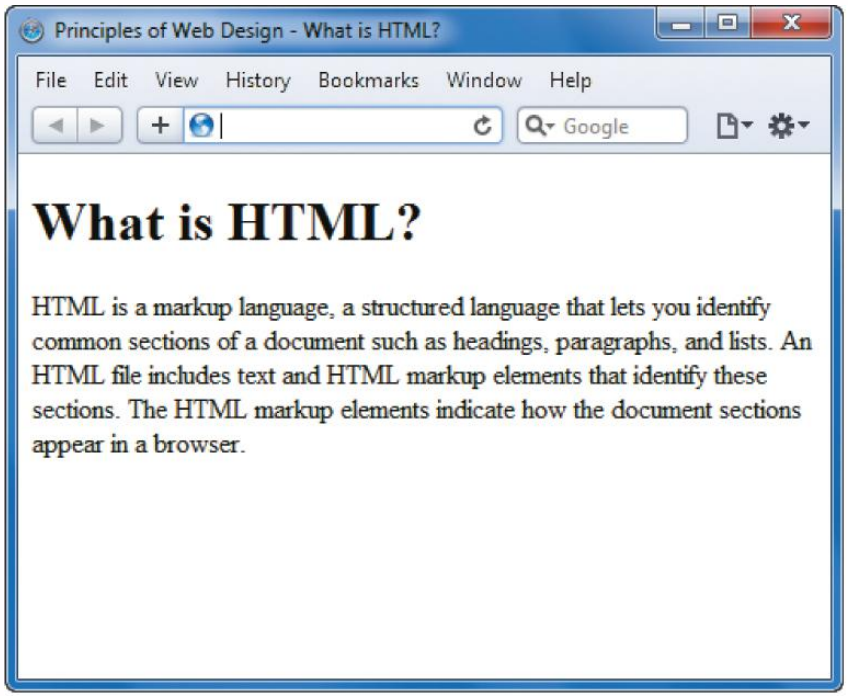
Creating Web Pages with HTML

Creating Web Pages with HTML

- HTML is a markup language that lets you identify common sections of a Web page
- Markup elements define each section
- This <h1> element defines text as a first-level heading:

```
<h1>What is HTML?</h1>
```

Creating Web Pages with HTML




```
<!DOCTYPE html>

<html>
<head>
<title>Principles of Web Design - What is HTML?</title>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
</head>

<body>
<h1>What is HTML?</h1>
<p>HTML is a markup language, a structured language
that lets you identify common sections of a
document such as headings, paragraphs, and lists.
An HTML file includes text and HTML markup elements
that identify these sections. The HTML markup
elements indicate how the document sections appear
in a browser.</p>
</body>
</html>
```

Figure 1-1 Basic Web page and its HTML code

Structure of a Basic Web Page

- The HTML file contains content text and HTML markup
- The HTML markup does not appear in the browser
- The browser interprets the code and displays the results

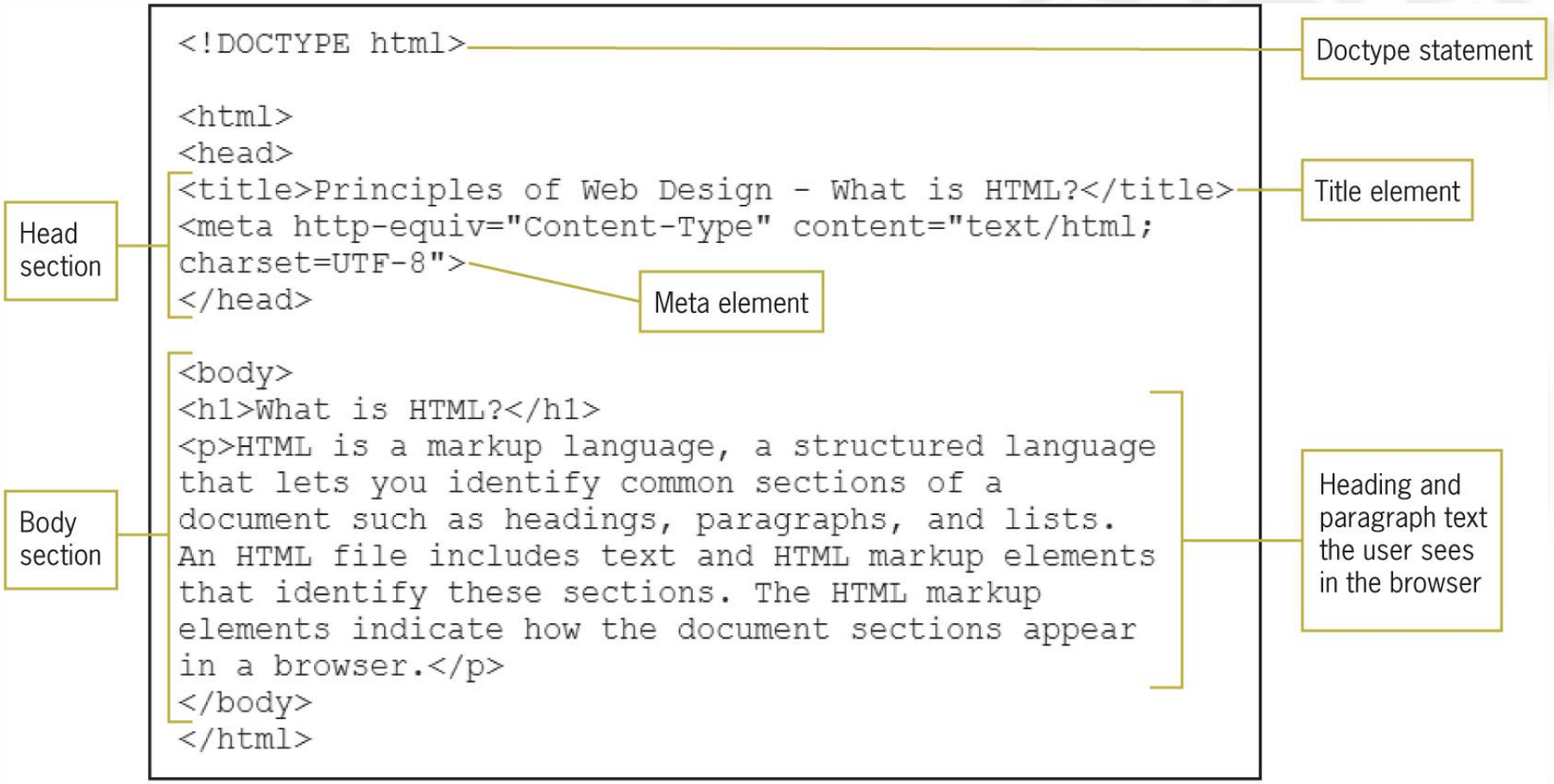


Figure 1-2 Structural elements in an HTML file

Structure of a Basic Web Page

- The document type, or doctype for short, specifies the rules for the document language
- The <html> tag is the root element
- The two main sections are the <head> and <body> elements
- The head section is the container for all of the descriptive information about the document
- The <body> section includes the content that the user sees in the browser window

Structure of a Basic Web Page

- The body of the document can contain:
  - Text
  - Images
  - Video content
  - Audio content
  - Forms for gathering information
  - Interactive content
  - Links to other Web resources

HTML in the Browser

- The browser interprets the HTML markup elements and displays the results, hiding the actual markup from the user
- Each browser interprets HTML in its own way, based on its rendering engine
- It is essential that you test your work in different Web browsers

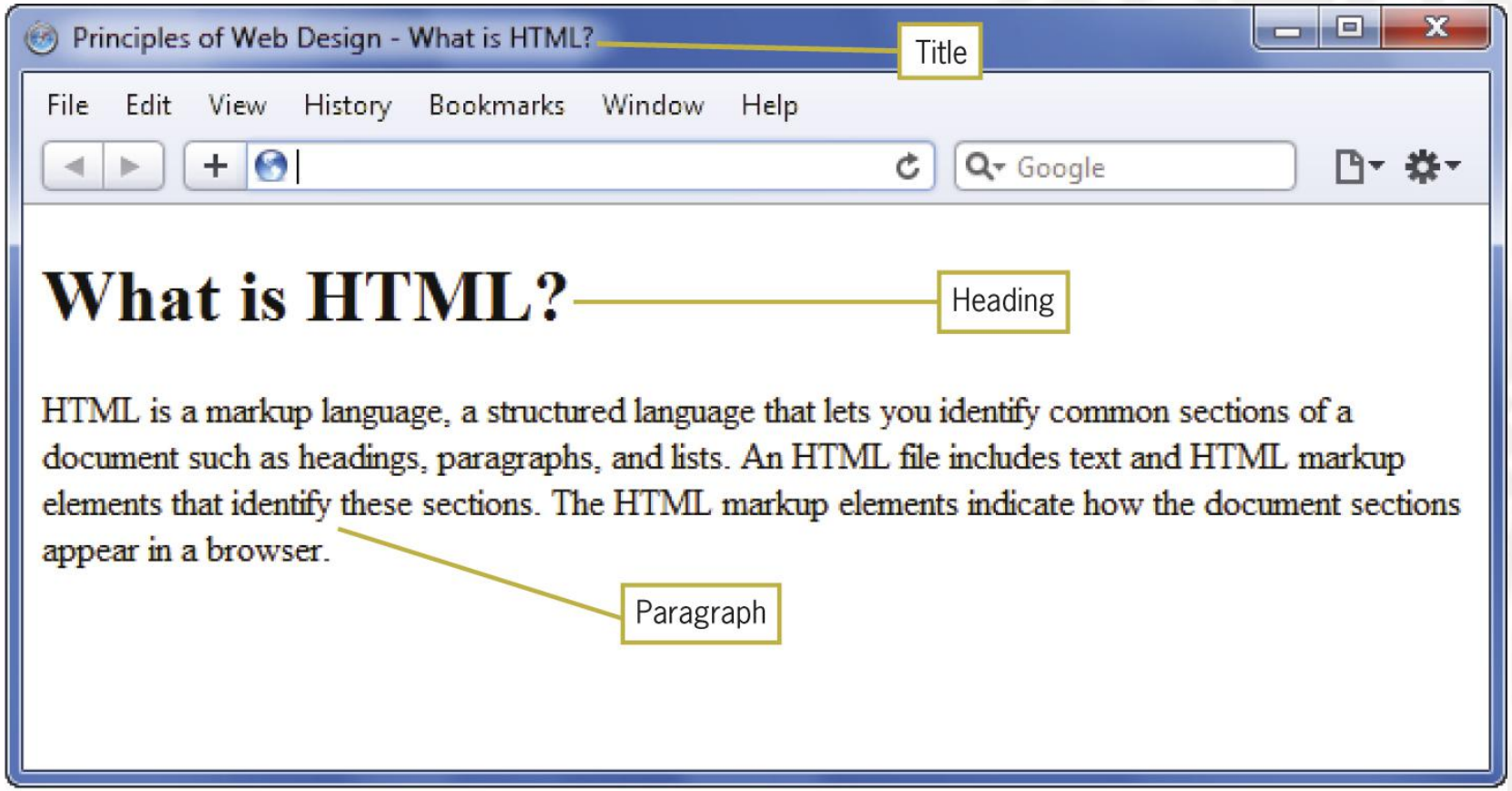


Figure 1-3 Browser interprets the HTML markup and displays the results

Adding Style with CSS

- Web designers use Cascading Style Sheets (CSS) to add presentation information to Web pages
- With CSS you can display information for different devices
- With style sheets, the presentation properties are separate from the content
- CSS lets you control the presentation characteristics of an entire Web site with a single style sheet

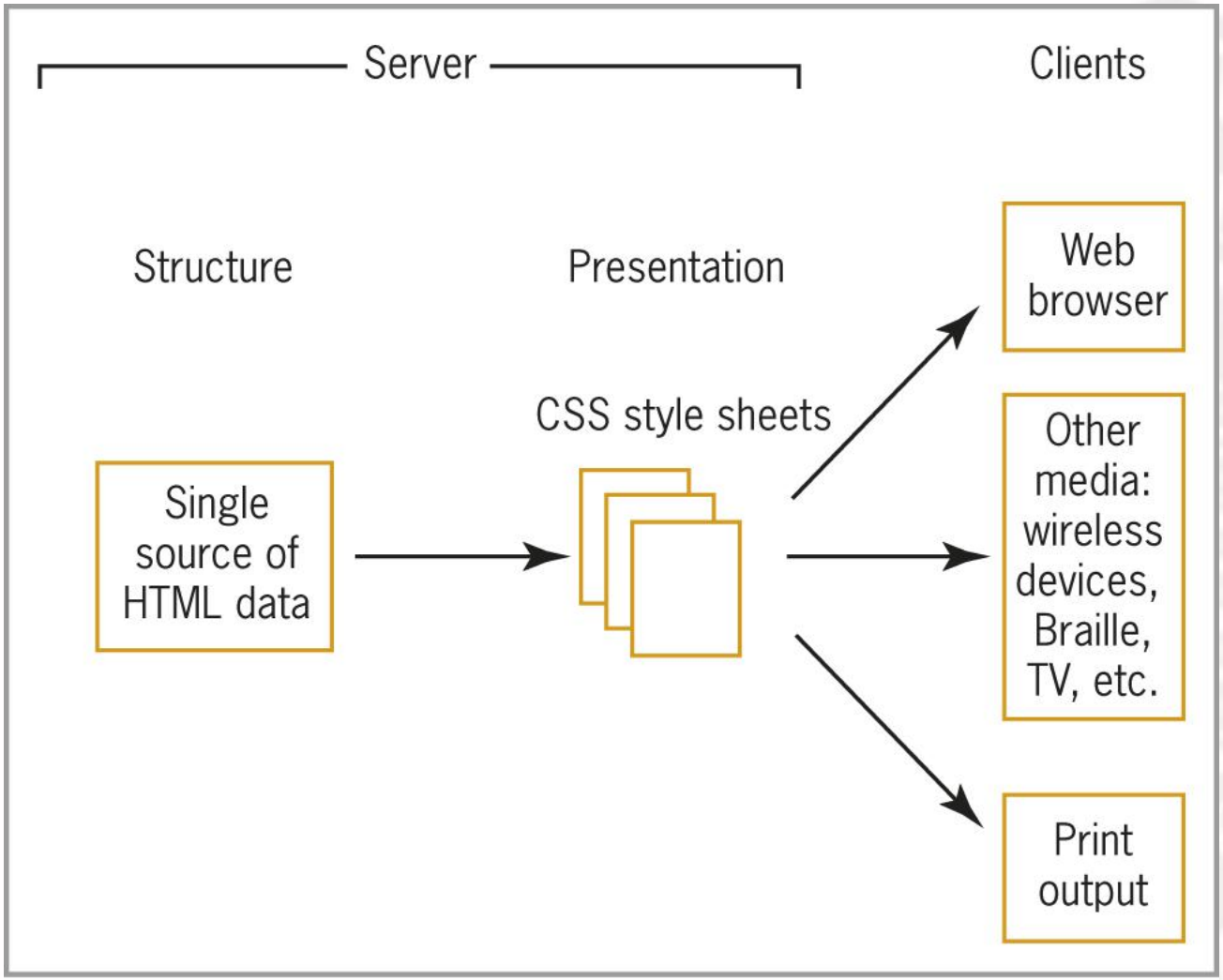


Figure 1-4 Formatting data for multiple destinations

Adding Style with CSS

- The next two figures show CSS style rules and the result in the browser
- The style rules in Figure 1-5 specify that the body text for the page will be Arial, the h1 will have a bottom border, and the paragraph will have a 30-pixel left margin.
- Figure 1-6 shows the results in Firefox

```
<!DOCTYPE html>

<html>
<head>
<title>Principles of Web Design - What is HTML?</title>

<style type="text/css">
body {font-family: arial;}
h1 {border-bottom: solid 1px;}
p {margin-left: 30px;}
</style>

</head>

<body>
<h1>What is HTML?</h1>
<p>HTML is a markup language, a structured language
that lets you identify common sections of a document
such as headings, paragraphs, and lists. An HTML
file includes text and HTML markup elements that
identify these sections. The HTML markup elements
indicate how the document sections appear in a
browser./p>
</body>
</html>
```

CSS style section

Figure 1-5 CSS style section contains presentation information

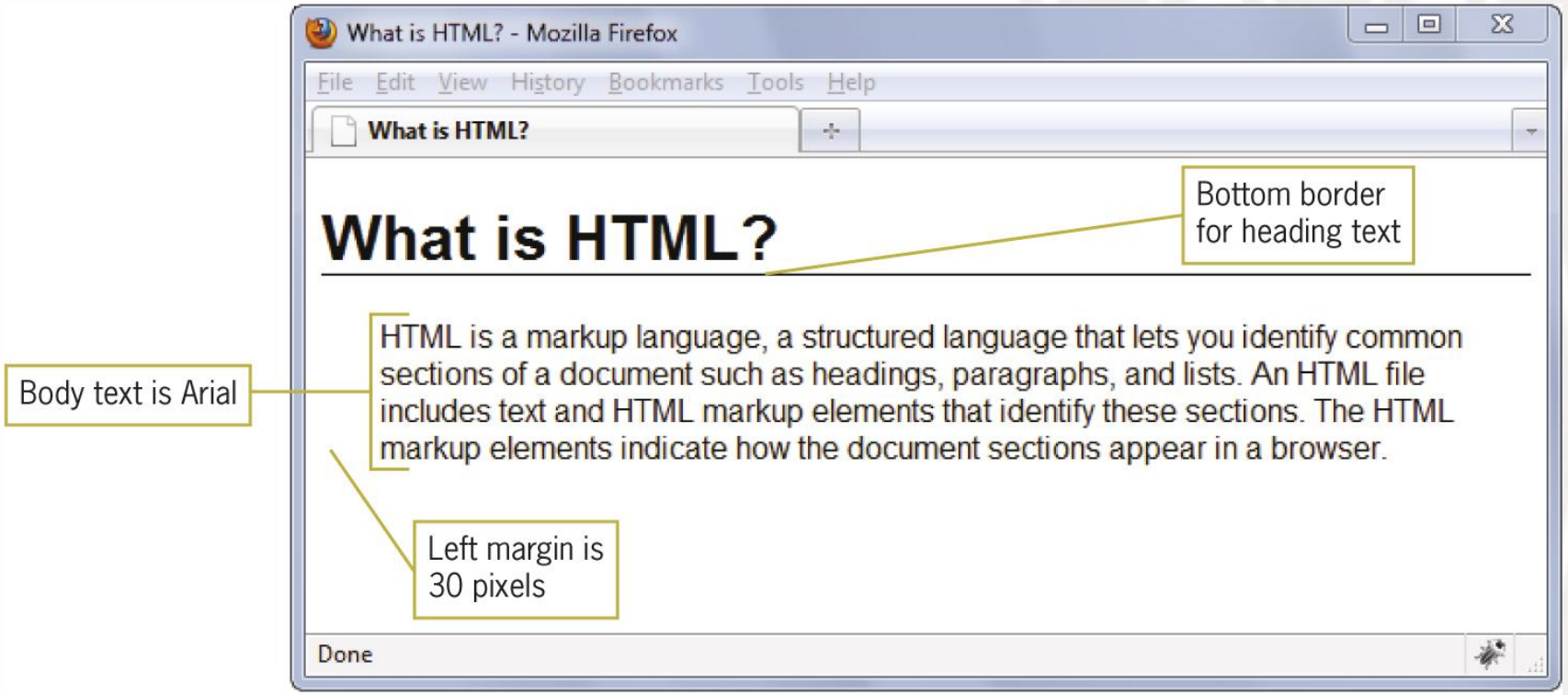


Figure 1-6 Result of adding the CSS style rules

Organizing Information with Hypertext

- The World Wide Web links information on related topics using hypertext
- You determine which terms to create as hypertext links and where users end up when they click a link
- The different types of linked content and media continually evolve

The History of HTML

The History of HTML

- As a Web designer, you will encounter all types of HTML coding practices
- Understanding the evolution of HTML will help you understand various Web design methods
- To be a successful Web designer, you need to understand the past, present, and future directions of HTML, coding standards, and common practices

The History of HTML

- Tim Berners-Lee first proposed HTML at the European Laboratory for Particle Physics (CERN) in 1989
- Berners-Lee joined the ideas of the browser, a markup language (HTML), and a communications protocol that allowed hypertext linking
- Not only could people read documents, they could easily create them using HTML

The History of HTML

- HTML is an application of the Standard Generalized Markup Language (SGML), a standard system for specifying document structure
- Berners-Lee joined the ideas of the user interface (browser), a markup language (HTML), and a communications protocol (http:) that allowed hypertext linking

A Need for Standards

- The World Wide Web Consortium (W3C) was founded in 1994 MIT
- The World Wide Web Consortium sets standards for HTML and other markup languages
- Jointly developed standards, rather than ones dictated by one vendor, benefit everyone

A Need for Standards

Version	Release Date	Highlights
HTML1.1	1992	First informal draft
HTML 2.0	1995	First release supported by graphical browsers; documents written in HTML 2.0 can still be viewed in all browsers
HTML 3.2	1997	Introduced forms and tables
HTML 4.01	1999	Added support for style sheets, and increased support for scripting and interactivity
HTML5	Future final release, in use today	Latest version adds page layout elements, audio/visual elements, enhanced animation and graphic support

Table 1-1 History of HTML

XML and XHTML – A New Direction

- In 1997, the W3C released XML, the Extensible Markup Language
- XML is essential to creating applications for the Web
- XML lets developers define their own markup language
- XML has a stricter syntax than HTML

XML Syntax Rules

- Documents must be well-formed
- All tags must nest properly and not overlap
- Use all lowercase for element names
- Always use closing tags
- Empty elements are signified by a closing slash
- Attribute values must be contained in quotation marks

Sample XML

```
<poem>
<title>An Ode to the Web</title>
<stanza>
<line>So many Web sites</line>
<line>So little time</line>
<line>And all I want to do</line>
<line>Is critique their design!</line>
</stanza>
</poem>
```

XML and XHTML – A New Direction

- XML syntax provides a solution to the problem of widely varying HTML coding standards
- The W3C combined XML and HTML to create XHTML
- XHTML follows the rules of XML
- Web developers readily adopted XHTML and CSS to standardize coding
- Many Web sites benefited from leaner standardized code

Problems with XHTML

- Relaxed syntax rules still must be applied because of legacy code
- Newer versions of XHTML moved too far away from existing Web development
- XHTML was not well received by the development community

A Proposal for HTML5

- The Web Hypertext Application Technology Working Group (WHATWG) proposed HTML5
- HTML5:
  - Supports standards-based coding
  - Compatible with HTML and XHTML
  - Compatible with CSS
  - Supports new page layout elements
  - Application and media compatible

Working with HTML5

Working with HTML5

- HTML5 attempts to address shortcomings of HTML
- Addresses needs of modern Web design
- Offers new features:
  - Logical layout elements
  - Rich media
  - Support for applications
- Removes old features:
  - All display elements have been removed in favor of CSS
  - Framesets are gone

Working with HTML5

HTML5 offers two syntaxes:

- An HTML-compatible syntax
- An XML-compatible syntax

- HTML-compatible syntax
  - More relaxed syntax
  - Code shortcuts allowed
- XML-compatible syntax
  - Consistent with XHTML
  - Uses XML syntax rules

Working with HTML5

```
<!DOCTYPE html>
<html>
<head>
<title>HTML5 Loose Syntax</title>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
</head>
<body>
<h1>Example</h1>
<p class=content>HTML5 allows a looser syntax.
</body>
</html>
```

MIME type specifies HTML

<p> element has no closing tag

Attribute value is not quoted

Figure 1-8 Looser HTML5 syntax

Working with HTML5

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>HTML5 Strict Syntax</title>
<meta http-equiv="Content-Type"
content="application/xhtml+xml; charset=ISO-8859-
1">
</head>
<body>
<h1>Example</h1>
<p class="content">HTML5 has a stricter XML-based
syntax</p>.
</body>
</html>
```

XHTML namespace

MIME type specifies XHTML

Attribute value is quoted

<p> element is properly closed

Figure 1-9 Stricter HTML5 syntax

Choosing the Correct Syntax

- HTML5 allows a mixture of these two types of syntax into one document
- A polyglot document: mixed language
- Use XML coding standards in your HTML documents
- Lets you create standardized, compatible code

Choosing the Correct doctype

- Always use a doctype statement
- Using a doctype forces the browser to display in standards mode
- The standard doctype statement for HTML5:

```
<!DOCTYPE html>
```

Choosing the Correct MIME type

- Multipurpose Internet Mail Extensions (MIME) defines content types for the Web
- Determines the type of document
- Declared in a <meta> element in the <head> section
- Also contains a character set identifier
- Your <meta> element should look like this:

```
<meta http-equiv="Content-Type"
  content="text/html; charset=utf-8">
```

Creating Syntactically Correct Code

- Documents must be well-formed
- All tags must nest properly and not overlap
- Use all lowercase for element names
- Always use closing tags
- Empty elements are marked with a closing slash
- Attribute values must be contained in quotation marks

HTML5 Element Categories

- Metadata content
- Flow content
- Sectioning root
- Sectioning content
- Heading content
- Phrasing content
- Embedded content
- Interactive content
- Transparent

New Elements in HTML5

- HTML5 has a number of new elements; see Table 1-2 in the text
- Not all elements are supported by current browsers
- Test new elements carefully

Attributes in HTML5

- Elements can contain attributes that set properties
- Earlier versions of HTML had more attributes
- HTML5 has less attributes because of CSS
- Global attributes can be applied to any element

Obsolete Elements in HTML5

- Many elements have been removed in HTML, mostly involving presentation effects
- Framesets are no longer available

Using HTML5 Elements for Page Structure

- Most Web pages contain common characteristics:
  - Header
  - Navigation
  - Articles
  - Figures
  - Footers
  - Sidebars
- These are currently marked up with elements and id or class names

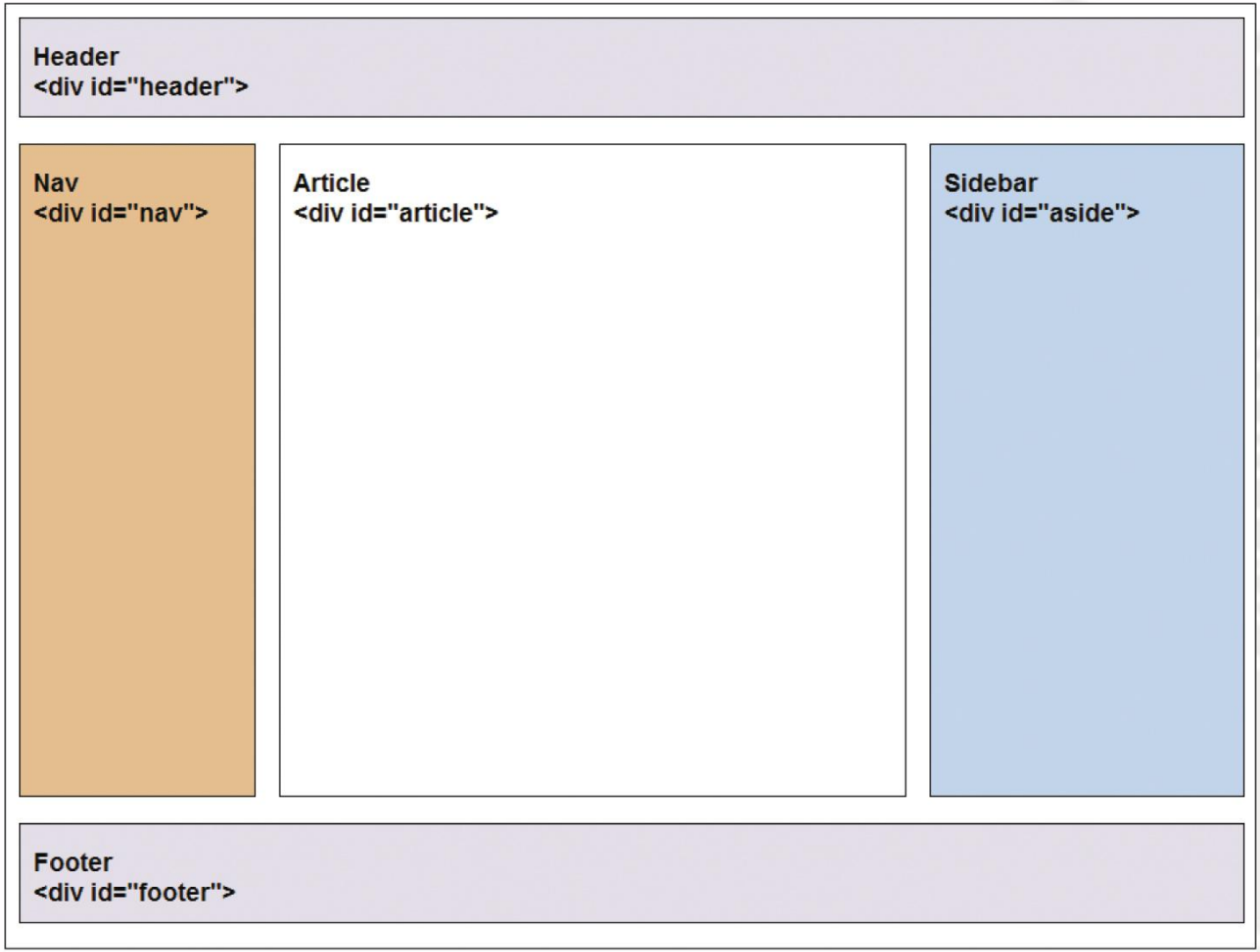


Figure 1-10 Five page sections using elements and id names

Using HTML5 Elements for Page Structure

- HTML5 offers a new set of elements for describing document structure
- HTML5 replaces the use of with named elements to structure the page
- The <article> element can be used instead of the element, for example:

```
<article>This is the main content of the Web page</article>
```

Using HTML5 Elements for Page Structure

- The HTML5 elements for page layout include:
  - <header> Contains the page header content
  - <nav> Contains the navigation elements for the page
  - <article> Contains the primary page content
  - <section> Defines sections or groupings of page content
  - <aside> Contains additional content such as a quote or sidebar
  - <figure> Contains images for the article content
  - <footer> Contains page footer content

Interactive Capabilities in HTML5

- Audio and video
- Drawing canvas
- Background application processing
- Local data storage

Choosing an HTML Editor

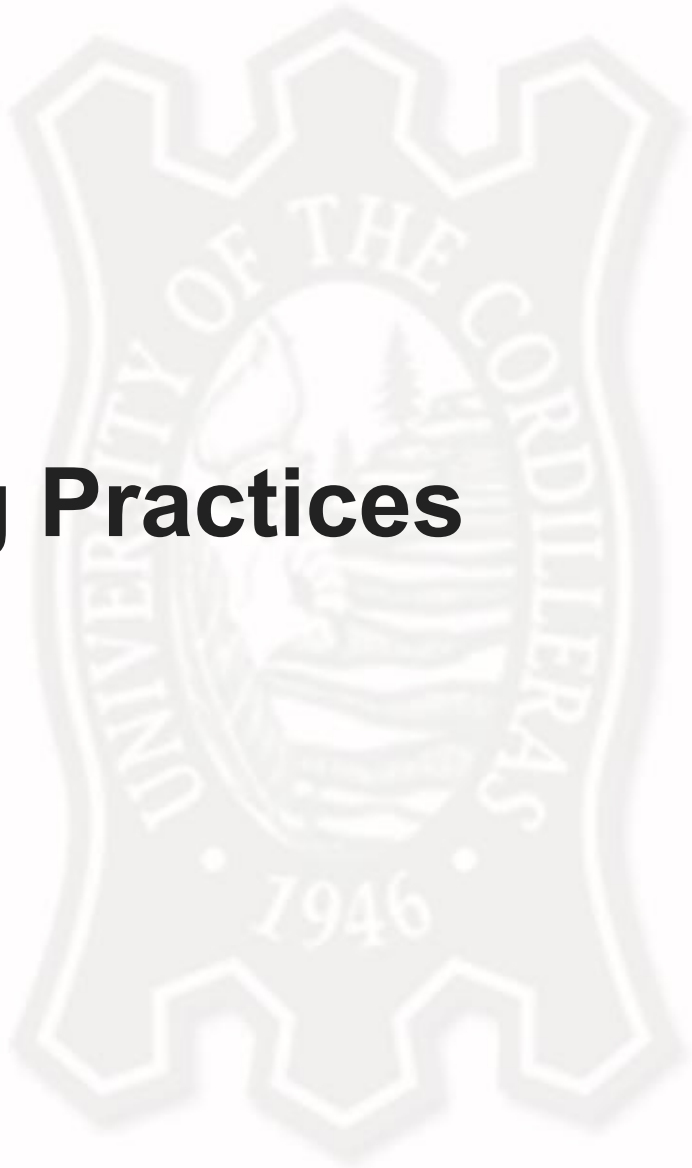
Choosing an HTML Editor

- Editors are either code-based or WYSIWYG (What You See is What You Get)
- Some editors offer both methods
- Many editors offer built-in code validators and FTP clients
- You can find low-cost freeware or shareware editors
- You can also use a text editor to create Web pages

Choosing an HTML Editor

- HTML editors contain their own logic for interpreting code
- Make sure to check your work in multiple browsers
- Some editors create overly complex code

Using Good Coding Practices



Using Good Coding Practices

- Creating code that ensures the greatest standards-compliance, presentation, and usefulness of your content
  - Stick to the standards
  - Use semantic markup
  - Validate your code

Stick to the Standards

- Stick to the W3C standards
- Separate content from presentation
- Plan to be accessible to different devices
- Standardized design is more accessible

Use Semantic Markup

- Semantic markup identifies the intended use of document sections
- Accurately describes each piece of content
- Until recently, this logical use of the HTML elements was largely ignored
- Document elements match the meaning and usage of the document sections: <p> for paragraph, <h1> for top-level heading, and so on

Validate Your Code

- Valid code conforms to the usage rules of the W3C
- The lack of valid code is a major problem
- Valid code enhances browser compatibility, accessibility, and exchange of data
- The most common mistakes include:
  - No doctype declaration
  - Missing closing tags
  - Missing alt attributes in <img> elements
  - Incorrect tag nesting
  - Unquoted attributes

Migrating from Legacy HTML to HTML5

- The transition should be a gradual process
- Clean up code on existing pages
- Plan coding conventions for new pages, removing deprecated elements
- Move display information to CSS
- Test for backwards compatibility

Summary

- Make sure to check for support of new HTML5 elements, and test carefully before adding them to your Web site
- Use the HTML5 naming conventions to name the content sections of your site even if only using them as class or id names
- Use Cascading Style Sheets
- Decide whether to code to the XML standard
- Use good coding practices