Tutorial 9
Working with XHTML

HTML, CSS, and Dynamic HTML

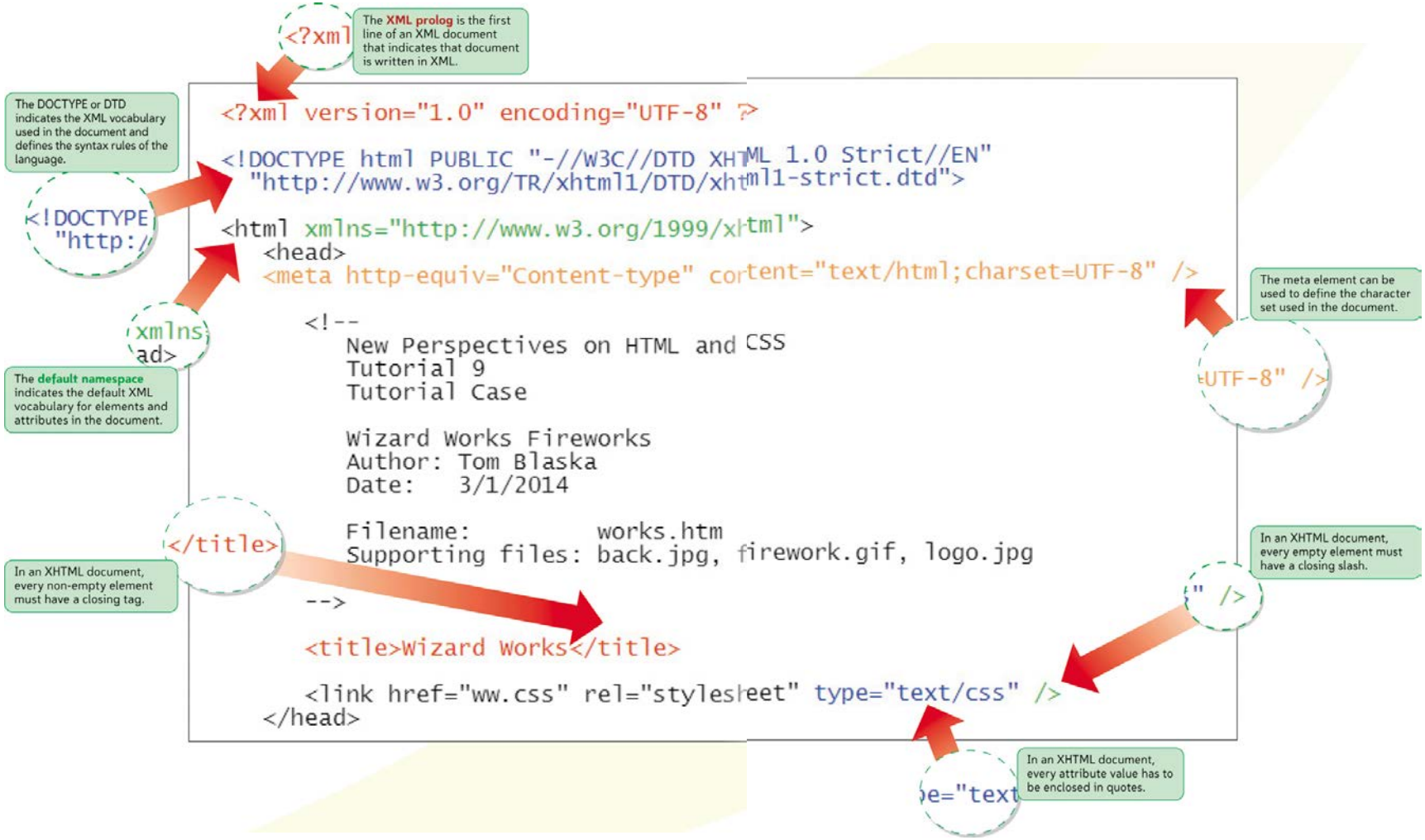
5TH EDITION

Carey, P. (2013). New perspectives on HTML, CSS, and Dynamic HTML: Comprehensive (5th ed.). Australia ; United Kingdom: Course Technology Cengage Learning.

Objectives

- Describe the history and theory of XHTML
- Understand the rules for creating valid XHTML documents
- Apply a DTD to an XHTML document
- Understand how to apply the XHTML namespace
- Explore the relationship between HTML5 and XHTML
- Test an XHTML document under the transitional DTD
- Test an XHTML document under the strict DTD
- Explore the use of character and parsed character data

Structure of an XHTML Document



Introducing XHTML

- SGML (Standard Generalized Markup Language)
  - Device-independent and system-independent
  - Introduced in the 1980s
  - Not intended for the World Wide Web
- HTML
  - Standards get confusing among browsers
  - Can be applied inconsistently

Introducing XHTML

Figure 9-1 Versions of XHTML

Version	Date Released	Description
XHTML 1.0	2001	This version is a reformulation of HTML 4.01 as an XML vocabulary, bringing the rigor of XML to Web document code.
XHTML 1.1	2002	A minor update to XHTML 1.0 that allows for modularity and simplifies writing extensions to the language.
XHTML 2.0	Discontinued in 2009	A follow-up version to XHTML 1.1; XHTML 2.0 was not backward compatible with earlier XHTML versions and was discontinued due to lack of support.
XHTML5	In development	A version of HTML5 written as an XML vocabulary; unlike XHTML 2.0, XHTML5 will be backward compatible with earlier XHTML versions.

Creating an XHTML Document

- The first line of an XHTML file contains a statement called a **prolog** that indicates the document adheres to the syntax rules of XML. The form of the XML prolog is

```
<?xml version="value"
encoding="type" ?>
```

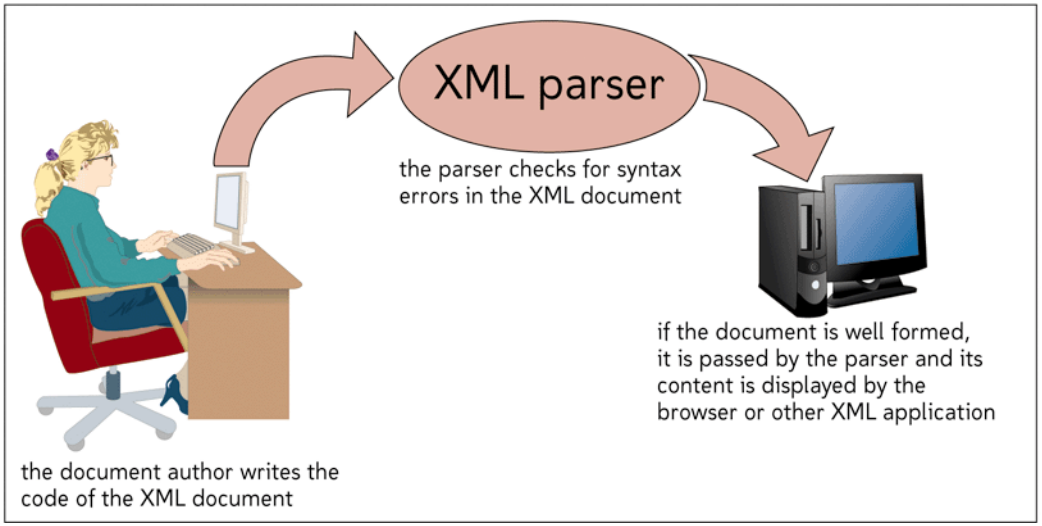
Figure 9-2 Inserting the XML prolog and meta element

```
<?xml version="1.0" encoding="UTF-8" ?>
<html>
   <head>
      <meta http-equiv="Content-type" content="text/html;charset=UTF-8" />
```

Creating Well-Formed Documents

- XML documents must be evaluated with an **XML parser**
- An XML document with correct syntax is a **well-formed document**

Figure 9-4 Testing for well-formedness



Creating Well-Formed Documents

Figure 9-5 Rules for well-formed XML

Rule	Incorrect	Correct
Element names must be lowercase.	`<P>This is a paragraph.</P>`	`<p>This is a paragraph.</p>`
Elements must be properly nested.	`<p>This text is <b>bold.</p></b>`	`<p>This text is <b>bold.</b></p>`
All elements must be closed.	`<p>This is the first paragraph.` `<p>This is the second paragraph.`	`<p>This is the first paragraph.</p>` `<p>This is the second paragraph.</p>`
Empty elements must be terminated.	`This is a line break. `	`This is a line break. `
Attribute names must be lowercase.	`<td ALIGN="right">`	`<td align="right">`
Attribute values must be quoted.	`<table width=620>`	`<table width="620">`
Attributes must have values.	`<option selected>`	`<option selected="selected">`

Creating a Well-Formed Document

- XHTML documents must also include a single root element that contains all other elements
  - For XHTML, that root element is the html element
- **Attribute minimization** is when some attributes lack attribute values
  - XHTML doesn't allow attribute minimization

Attribute Minimization in HTML and XHTML

Figure 9-6 Attribute minimization in HTML and XHTML

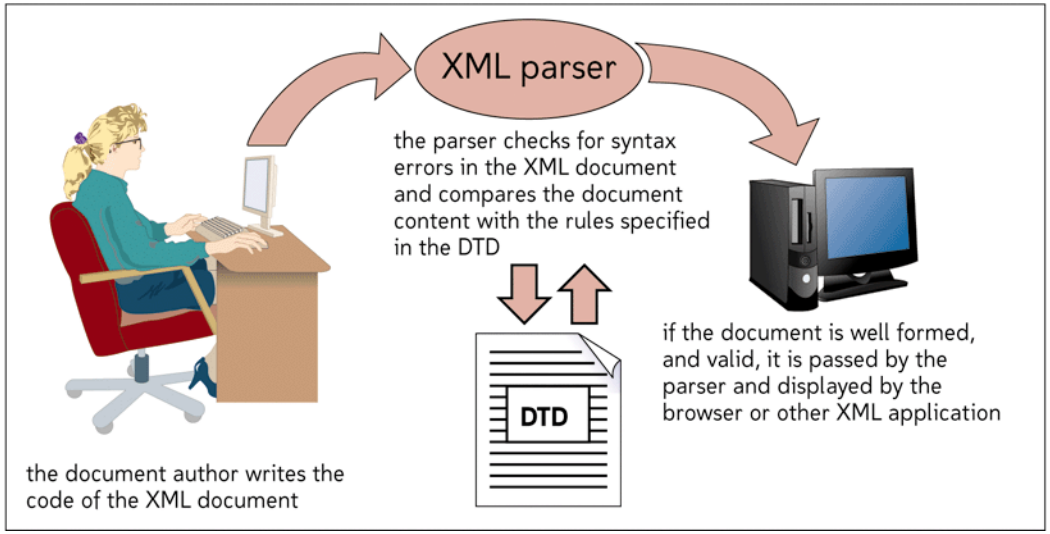
HTML	XHTML
compact	compact="compact"
checked	checked="checked"
declare	declare="declare"
readonly	readonly="readonly"
disabled	disabled="disabled"
selected	selected="selected"
defer	defer="defer"
ismap	ismap="ismap"
nohref	nohref="nohref"
noshade	noshade="noshade"
nowrap	nowrap="nowrap"
multiple	multiple="multiple"
noresize	noresize="noresize"

Creating Valid XHTML Documents

- A valid document is a well-formed document that also contains only those elements, attributes, and other features that have been defined for the XML vocabulary that it uses
- To specify the correct content and structure for a document, the developers of an XML-based language can create a collection of rules called the **document type definition** or **DTD**

Creating Valid XHTML Documents

Figure 9-7 Testing for validity



DTDs

- **Transitional**: supports many of the presentational features of HTML, including the deprecated elements and attributes. Best used for older documents that contain deprecated features.
- **Frameset**: used for documents containing frames, and also supports deprecated elements and attributes
- **Strict**: does not allow any presentational features or deprecated HTML elements and attributes. Does not support frames or inline frames. It is best used for documents that need to strictly conform to the latest standards

Creating Valid XHTML Documents

- The DTD used depends on the content of the document and the needs of your users
- To support old browsers, use the **transitional** DTD
- To support old browsers in a framed Web site, use the **frameset** DTD
- To support more current browsers and to weed out any use of deprecated features, use the **strict** DTD

Creating a Valid Document

- Elements **not** allowed under the strict DTD:
  - applet
  - basefont
  - center
  - dir
  - font
  - isindex
  - menu
  - noframes
  - s
  - strike
  - u

Creating a Valid Document

- Some attributes are restricted, while others are required in XHTML

Attributes Prohibited in the Strict DTD

Figure 9-9 Prohibited attributes under the XHTML strict DTD

Element	Prohibited Attributes
a	target
area	target
base	target
body	alink, bgcolor, link, text, vlink
br	clear
caption	align
div	align
dl	compact
form	name, target
hn	align
hr	align, noshade, size, width
img	align, border, hspace, name, vspace
input	align
li	type, value
link	target
map	name
object	align, border, hspace, vspace
ol	compact, start
p	align
pre	width
script	language
table	align, bgcolor
td	bgcolor, height, nowrap, width
th	bgcolor, height, nowrap, width
tr	bgcolor
ul	type, compact

Required XHTML Attributes

Figure 9-10 Required XHTML attributes

Element	Required Attributes
applet	height, width
area	alt
base	href
basefont	size
bdo	dir
form	action
img	alt, src
map	id
meta	content
optgroup	label
param	name
script	type
style	type
textarea	cols, rows

Inserting the DOCTYPE Declaration

- To specify which DTD is used by an XML document, you add a DOCTYPE declaration directly after the XML prolog

```
<!DOCTYPE root type "id" "url">
```

Inserting the DOCTYPE Declaration

Figure 9-11 DTDs for different versions of HTML and XHTML

DTD	DOCTYPE
HTML 2.0	`<!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">`
HTML 3.2	`<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">`
HTML 4.01 strict	`<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN "http://www.w3.org/TR/html4/strict.dtd">`
HTML 4.01 transitional	`<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
HTML 4.01 frameset	`<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">`
HTML5	`<!DOCTYPE html>`
XHTML 1.0 strict	`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`
XHTML 1.0 transitional	`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
XHTML 1.0 frameset	`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`
XHTML 1.1	`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">`
XHTML5	`<!DOCTYPE html>`

The XHTML Namespace

- A namespace is a unique identifier for elements and attributes originating from a particular document type (like XHTML or MathML)
- Two types of namespaces:
  - Default: applied to any element or attribute in the document

```
<root xmlns="namespace">
```

The XHTML Namespace

- Local: applies to only select elements
  - Each element in the local namespace is marked by a prefix attached to the element name

    xmlns: ***prefix*****="*****namespace*****"**
  - Identify any element belonging to that namespace by modifying the element name in the tag

    `prefix:`*`element`*

Setting the XHTML Namespace

- To set XHTML as the default namespace for a document, add the xmlns attribute to the html element with the following value:

```
<html
xmlns=http://www.w3.org/1999/xhtml
>
```

HTML5 and XHTML

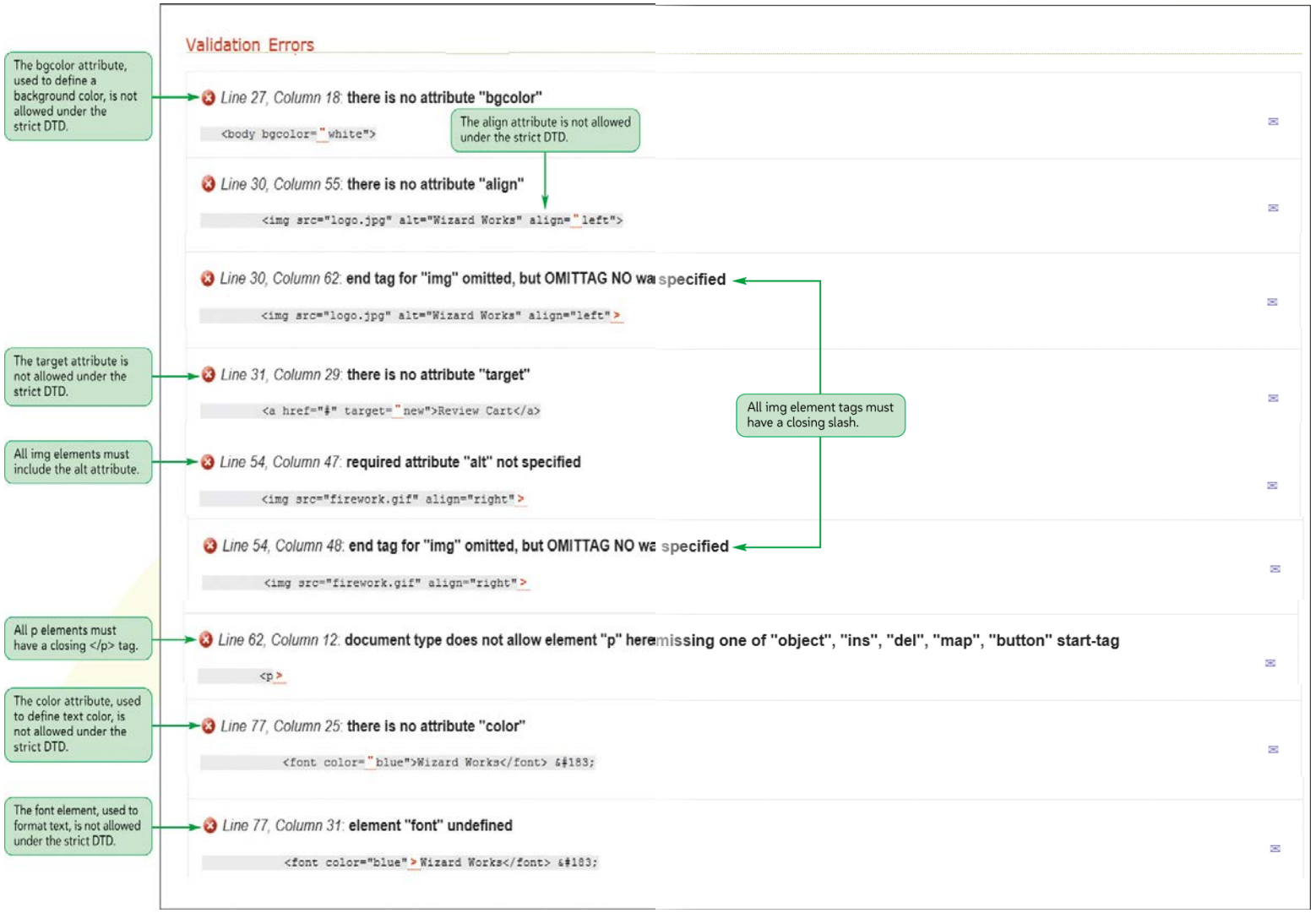
- HTML5 was developed to be backward compatible with earlier versions of HTML, and also to support the common application of HTML syntax
- The rules for HTML5 are much more open than for XHTML

HTML5 and XHTML

Figure 9-14 HTML5 vs. XHTML

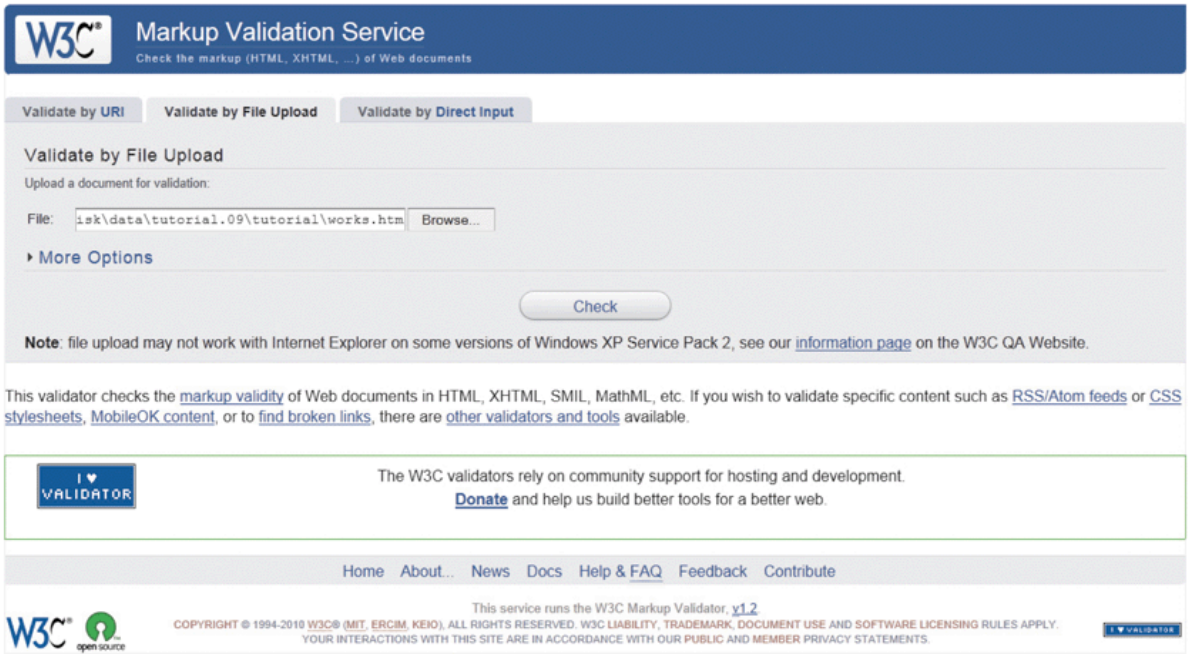
Syntax Issue	HTML5	XHTML
Attribute minimization	Attributes need not have attribute values.	All attributes must have attribute values.
Attribute names	Attribute names are not case sensitive.	Attribute names must be lowercase.
Attribute values	Unquoted attribute values are allowed.	All attribute values must be enclosed in quotes.
DOCTYPE	A DOCTYPE is required in the form <!DOCTYPE html >	The DOCTYPE is optional.
Element names	Element names are not case sensitive.	Element names must be lowercase.
Empty element tags	A closing slash may or may not be used with an empty element tag.	Empty element tags must include a closing slash.
Error validation	In HTML5 there are no well-formedness constraints; no errors are fatal.	Well-formedness errors are fatal.
Namespace prefixes	Namespace prefixes are not supported in HTML5.	Namespace prefixes are supported in XHTML.
Namespaces	Elements and attributes for known vocabularies (HTML, SVG, and MathML) are implicitly assigned.	Namespaces must be explicitly declared using the `xmlns` attribute.
Processing instructions	HTML5 does not support processing instructions and instead treats the enclosed text as a comment.	Allows the use of processing instructions closed with `?>`

Validating an XHTML Document



Validating Under XHTML Transitional

Figure 9-15 Selecting a file for validation



Testing Under XHTML Strict

- To test under another DTD, you'll need to change the DOCTYPE declaration

Figure 9-21 Pasting the XHTML 1.0 strict DTD

```
<?xml version="1.0" encoding="UTF-8" ?>

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

<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      <meta http-equiv="Content-type" content="text/html;charset=UTF-8" />
```

Figure 9-26 Inserting code for the W3C validation icon

```
      <p>
         <a href="http://validator.w3.org/check?uri=referer">
         <img src="http://www.w3.org/Icons/valid-xhtml10" alt="Valid XHTML 1.0 Strict"
              height="31" width="88" /></a>
      </p>

   </div>

</body>
```

Using Embedded Style Sheets in XHTML

- **Parsed character data** (**PCDATA**) is text processed by a browser or parser
- **Unparsed character data** (**CDATA**) is text not processed by the browser or parser
  - A CDATA section marks a block of text as CDATA so that parsers ignore any text within it