

Introduction to HTML Training

Introduction to HTML

Lesson 1, Activity 3: A Simple HTML Document

We're starting this lesson off with a quick exercise so that you can get your hands on HTML right away. Then we'll explain what you have created and start introducing you to basic HTML concepts.

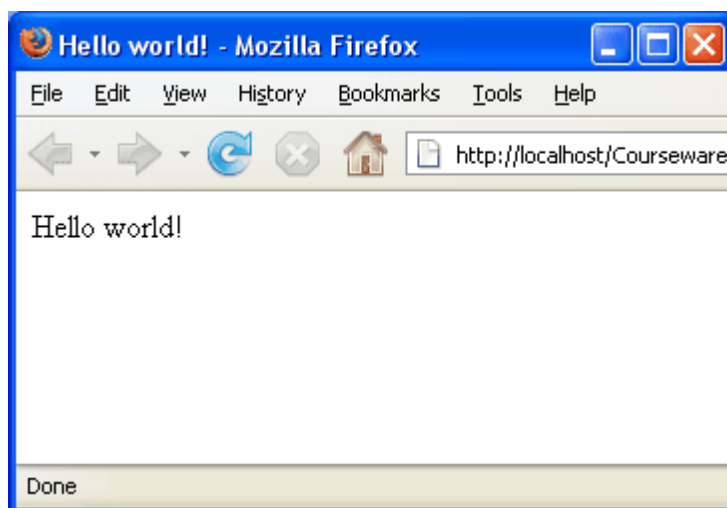
Duration: 5 to 15 minutes.

In this exercise, you will create your first HTML document by simply copying the text shown below. The purpose is to give you some sense of the structure of an HTML document.

1. Open a simple text editor such as Notepad and create a new file. *Do not use an HTML editor for this exercise.*
2. Save the file as HelloWorld.html in the HTMLBasics/Exercises folder.
3. Type the following exactly as shown:

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Hello world!</title>
</head>
<body>
  Hello world!
</body>
</html>
```

4. Save the file again and then open it in your browser by navigating to the file in your folder system and double-clicking on it. The page should appear as follows:



Solution:

[HTMLBasics/Solutions/HelloWorld.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Hello world!</title>
</head>
<body>
  Hello world!
</body>
</html>
```

Lesson 1, Activity 5: The HTML Skeleton

At its simplest, an HTML page contains what can be thought of as a skeleton - the main structure of the page. It looks like this:

Code Sample:

[HTMLBasics/Demos/Skeleton.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title></title>
</head>
<body>
  <!--Content that appears on the page-->
</body>
</html>
```

The <head> Element

The <head> element contains content that is not displayed on the page itself. Some of the elements commonly found in the <head> are:

- Title of the page (<title>). Browsers typically show the title in the "title bar" at the top of the browser window.
- Meta tags, which contain descriptive information about the page (<meta>)
- Script blocks, which contain javascript or vbscript code for adding functionality and interactivity to a page (<script>)
- Style blocks, which contain Cascading Style Sheet rules (<style>).
- References (or links) to external style sheets (<link>).

The <body> Element

The <body> element contains all of the content that appears on the page itself. Body tags will be covered thoroughly throughout this manual.

Whitespace

Extra whitespace is ignored in HTML. This means that all hard returns, tabs and multiple spaces are condensed into a single space for display purposes.

Code Sample:

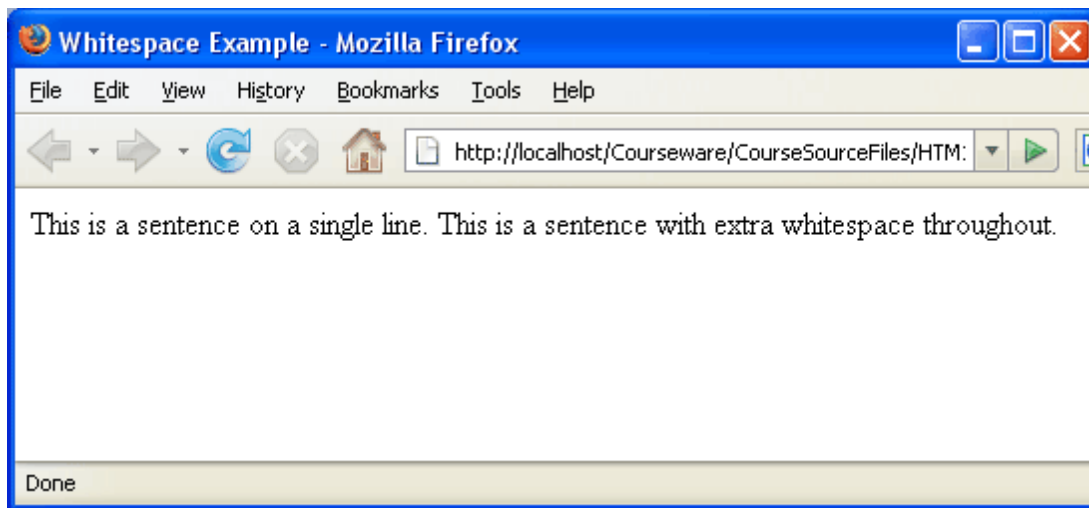
[HTMLBasics/Demos/Whitespace.html](#)

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
<title>Whitespace Example</title>
</head>
<body>
This is a sentence on a single line.

This
is
a
    sentence with
    extra whitespace
throughout.

</body>
</html>
```

The two sentences in the code above will be rendered in exactly the same way.



Lesson 1, Activity 6: HTML elements describe the structure and content of a Web page. Tags are used to indicate the beginning and end of elements. The syntax is as follows:

```
<tagname>Element content</tagname>
```

Attributes

Tags often have attributes for further defining the element. Attributes come in name-value pairs.

In HTML (without the X) some tags have attributes that don't take a value, like so:

```
<tagname att1>Element content</tagname>
```

Note that attributes only appear in the open tag, like so:

```
<tagname att1="value" att2="value">Element content</tagname>
```

The order of attributes is not important.

Empty vs. Container Tags

The tags shown above are called container tags because they have both an open and close tag with content *contained* between them. Tags that do not contain content are called empty tags. The syntax is as follows:

```
<tagname>
```

or

```
<tagname att1="value" att2="value">
```

Blocks and Inline Elements

Block-level Elements

Block elements are elements that separate a block of content. For example, a paragraph (<p>) element is a block element. Other block elements include:

- Lists (and)
- Tables (<table>)
- Forms (<form>)
- Divs (<div>)

Inline Elements

Inline elements are elements that affect only snippets of content and do not block off a section of a page. Examples of inline elements include:

- Links (<a>)
- Images ()
- Formatting tags (, <i>, <tt>, etc.)
- Phrase elements (, , <code>, etc.)
- Spans ()

Important: many formatting tags have been deprecated in HTML 4.0 and XHTML 1.0. Although the ones listed here are not deprecated, their use is discouraged as the same effect can be accomplished with CSS.

Comments

Comments are generally used for one of three purposes.

1. To write helpful notes about the code; for example, why something is written in a specific way.
2. To comment out some code that is not currently needed, but may be used sometime in the future.
3. To debug a page.

HTML comments are enclosed in <!-- and -->. For example:

```
<!-- This is an HTML comment -->
```

Lesson 1, Activity 7: Special Characters

Special characters (i.e, characters that do not show up on your keyboard) can be added to HTML pages using entity names and numbers. For example, a copyright symbol (©) can be added using `©` or `©`.

The following table shows some of the more common character references.

Description	Entity Number	Entity Name	Symbol
Quotation mark	"	"	"
Ampersand	&	&	&
Less than	<	<	<
Greater than	>	>	>
Non-breaking Space	 	 	
Inverted exclamation	¡	¡	¡
Cent sign	¢	¢	¢
Pound sterling	£	£	£
Yen sign	¥	¥	¥
Copyright	©	©	©
Registered trademark	®	®	®
Fraction one-fourth	¼	¼	¼
Fraction one-half	½	½	½
Fraction three-fourths	¾	¾	¾
Inverted question mark	¿	¿	¿
en dash	–	–	—
em dash	—	—	—
dagger	†	†	†
horizontal ellipsis	…	…	...
euro	€	€	€
trademark	™	™	™

Lesson 1, Activity 9: The HTML/XHTML Version

XHTML 1.0 and HTML 4.0 consist of the same sets of elements. (HTML5 introduces some new elements, but those are not covered in this course.) The only difference between HTML and XHTML is that HTML is fairly flexible; whereas, XHTML has strict rules.

HTML is a SGML-based; whereas XHTML is XML-based. XML and SGML are both meta-languages (languages for defining other languages). XML applies stricter rules than SGML does.

DOCTYPE Declarations

The DOCTYPE declaration goes at the beginning of the document and is used to indicate which version of (X)HTML the page uses. There are three versions of (X)HTML documents: strict, frameset and transitional (loose). In HTML, the DOCTYPE declaration is optional. In XHTML, it is required.

All this DOCTYPE stuff used to be really important and is explained below, but with the introduction of HTML5, there an easy way out, which we highly recommend as it is both simpler and forward-compatible. Instead of worrying about the complicated DOCTYPEs explained below, simply use the new HTML5 DOCTYPE:

```
<!DOCTYPE HTML>
```

This DOCTYPE is, believe it or not, completely backward compatible and will make all browsers work in "standards mode."

As many many web pages will still use the old DOCTYPEs, they are worth understanding and are explained below.

Strict

The strict versions of HTML and XHTML do not allow use of tags and attributes that have been deprecated.

Deprecated tags are tags that the W3C has indicated may eventually be removed from the specification because of new and better ways of accomplishing the same thing. Most of these tags are for formatting; the W3C recommends using Cascading Style Sheets instead. See

<http://www.w3.org/TR/html401/appendix/changes.html#h-A.3.1.2> for a list of deprecated tags.

The strict versions do not support framesets.

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

HTML Strict
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
```

Transitional (Loose)

The transitional (or loose) versions of HTML and XHTML allow for the use of deprecated tags and attributes. The transitional versions also do not support framesets.

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

HTML Transitional
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
```

Frameset

The frameset versions of HTML and XHTML are the same as the transitional versions, except that they also support frames.

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

HTML Frameset
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
    "http://www.w3.org/TR/html4/frameset.dtd">
```

Closing Tags

HTML 4.0 allows some closing tags to be omitted. For example, in HTML, list item (``) tags do not require a matching close tag (``).

In XHTML, all tags must be closed. Empty tags are closed by adding a forward slash before the final angle bracket of the tag:

```
<tagname att1="value" att2="value" />
```

Note the space before the forward slash. Though this is not required by XHTML, it may help older browsers from getting confused.

In HTML 4.0, the forward slash is not required:

```
<tagname att1="value" att2="value">
```

Case Sensitivity

In HTML, case is not important. In XHTML, all tags and attributes must be in lowercase letters.

Quotes

In HTML, attribute values do not always have to be in quotes; whereas, in XHTML quotes are required. Either single quotes or double quotes may be used.

Nesting

In both HTML and XHTML, tags should be nested properly. Proper nesting requires nested tags to be closed in reverse order from which they were opened. Another way to say this is that each element must be completely contained by its parent element. For example, the following line of code uses improper nesting:

```
<parent><child>sometext</parent></child>
```

The corrected line looks like this:

```
<parent><child>sometext</child></parent>
```

Some XML Stuff

The XML Declaration

XHTML documents are, by definition, XML documents. This means that they follow the rules of XML. Although not required, it is good practice to include an XML declaration in your XHTML documents. If included, the XML declaration must be at the very beginning of the document. The XML declaration looks like this:

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

For best results, it is best to define the encoding in a `meta` tag as well. Note that you should include the following tag within the `head` tag:

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

The XHTML Namespace

In XHTML documents, the `html` tag must contain an `xmlns` declaration for the XHTML namespace, which indicates that the document must conform to the rules defined in the XHTML namespace. The syntax is shown below:

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

Lesson 1, Activity 10: **HTML5**

Although HTML5 is not fully supported by modern browsers, there are a couple of things you can start doing right now:

1. Use the new simpler DOCTYPE: `<!DOCTYPE HTML>`
2. Use this simpler meta tag to specify the character set: `<meta charset="UTF-8">`

You can also safely avoid all the XHTML stuff, including the XML declaration and the XHTML namespace.

This makes the opening of your HTML page simpler:

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="UTF-8">
```

We will start our HTML files in this way throughout the course.

Lesson 1, Activity 12: **lang and xml:lang**

The `lang` and `xml:lang` attributes are used to tell the browser (or other user agent) the language contained within an element. The W3C recommends that both `lang` and `xml:lang` be included in the `html` tag of all XHTML documents, like so:

```
<html xml:lang="en" lang="en">
```

[According to the W3C](#), these attributes may be helpful in:

- Assisting search engines
- Assisting speech synthesizers
- Helping a user agent select glyph variants for high quality typography
- Helping a user agent choose a set of quotation marks
- Helping a user agent make decisions about hyphenation, ligatures, and spacing
- Assisting spell checkers and grammar checkers