XHTML Basics

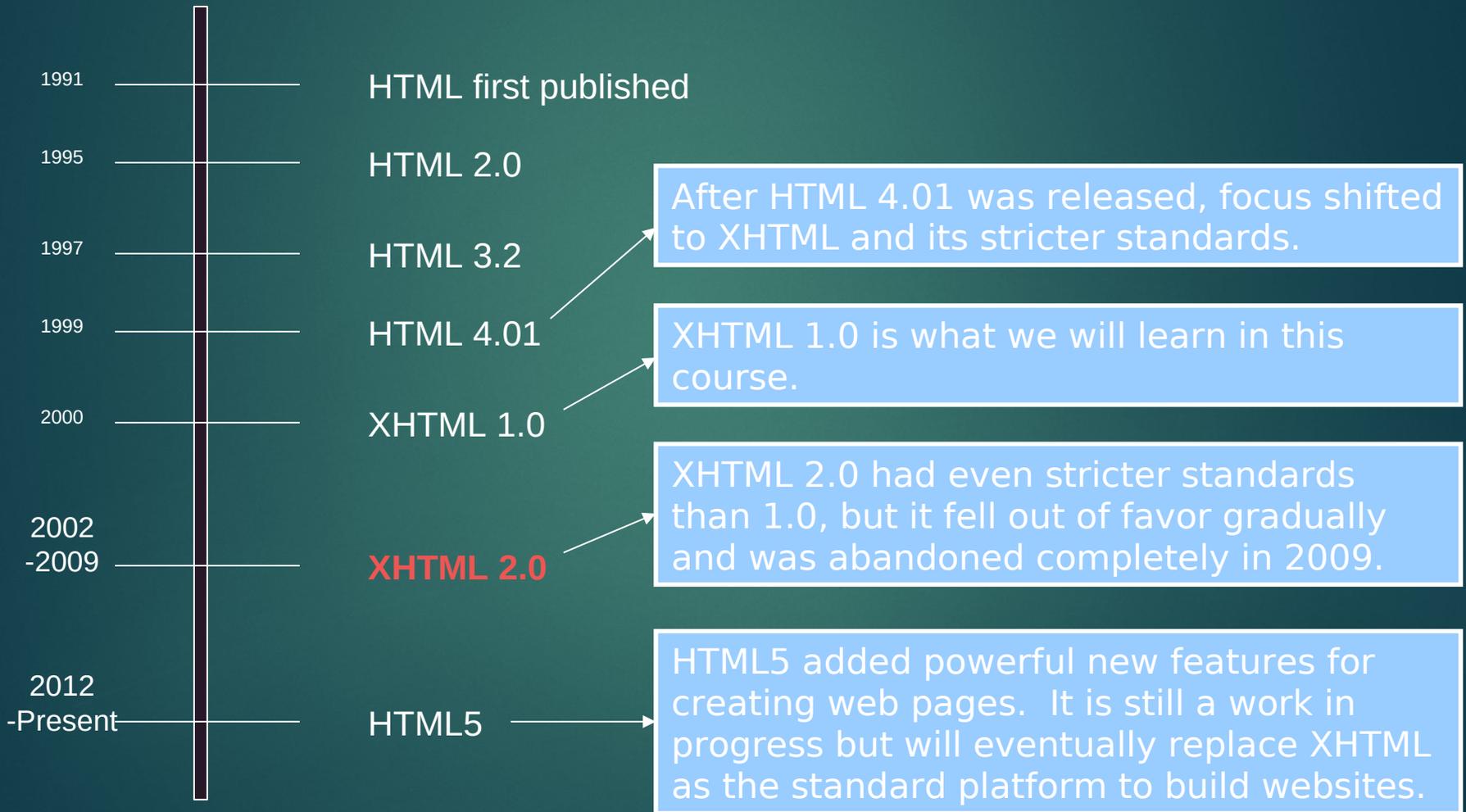# What is XHTML?

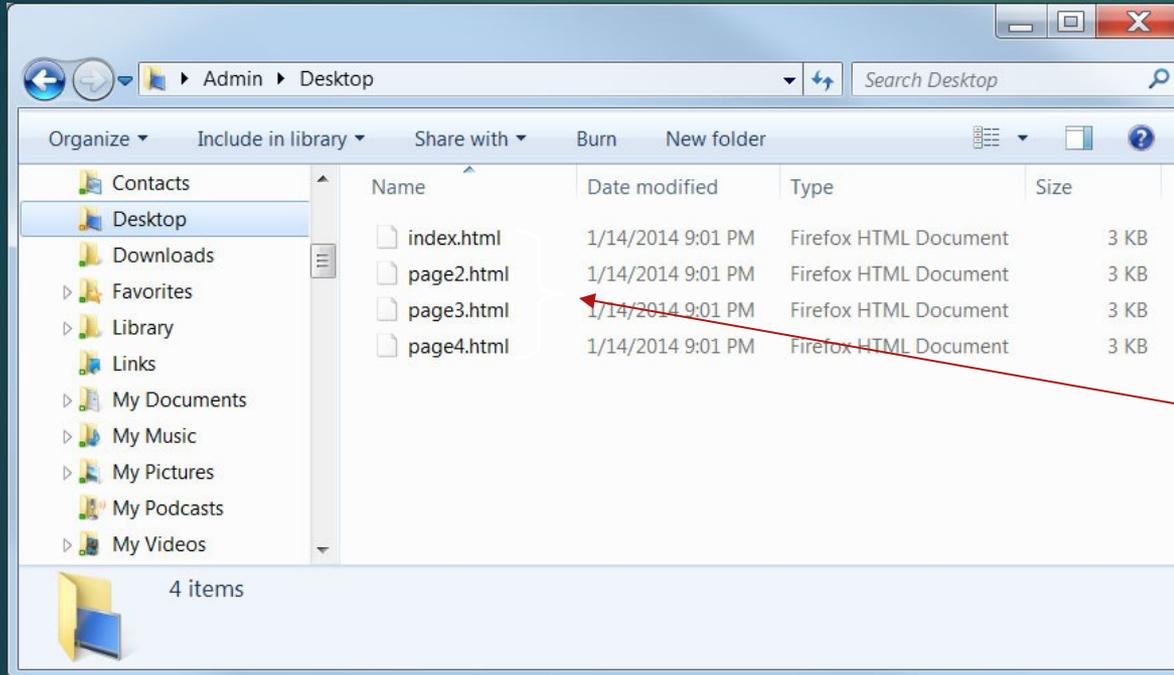XHTML is the language used to create and organize a web page:

- XHTML is newer than, but built upon, the original HTML (**HyperText Markup Language**) platform.

- XHTML has stricter rules and does not allow some elements formerly used in HTML.

- One benefit of using XHTML instead of HTML is that it helps make web pages look identical in different browsers, such as Internet Explorer, Firefox, Chrome, Safari, Opera, etc.

- XHTML defines and organizes the page content but does not format or style it. That is the job of CSS (**Cascading Style Sheets**), which we will learn later in this course.

History of HTML

Year	Version
1991	HTML first published
1995	HTML 2.0
1997	HTML 3.2
1999	HTML 4.01
2000	XHTML 1.0
2002 -2009	**XHTML 2.0**
2012 -Present	HTML5

After HTML 4.01 was released, focus shifted to XHTML and its stricter standards.

XHTML 1.0 is what we will learn in this course.

XHTML 2.0 had even stricter standards than 1.0, but it fell out of favor gradually and was abandoned completely in 2009.

HTML5 added powerful new features for creating web pages. It is still a work in progress but will eventually replace XHTML as the standard platform to build websites.

What is a Web Page?

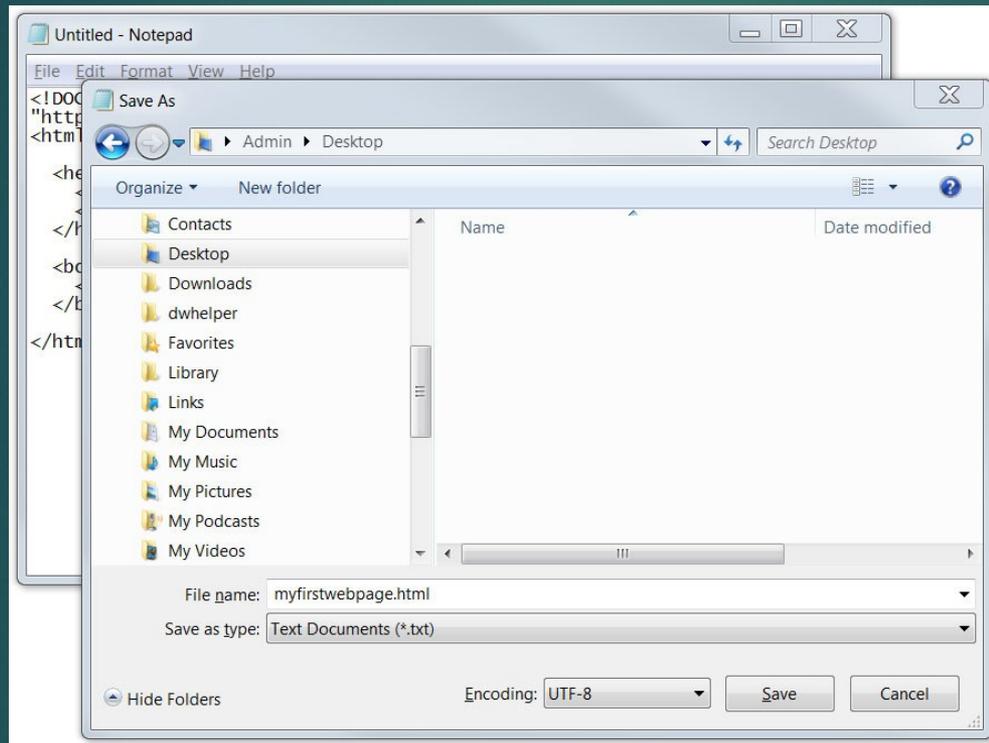A web page is a file on a computer, just like any other file:

Web pages are plain text documents saved with the ".html" file extension.

These HTML files can be opened in two different ways. If we open them using a text editor, such as Notepad, we see the HTML code. If we open them in a web browser (such as Internet Explorer or Firefox), the browser processes the code and displays the content in the browser window as a web page.

Creating a Web Document

To create a new web document, we can type it directly into a text editor, such as Notepad, and then save it with a ".html" extension:

Never use Microsoft Word or other word processing applications to create or edit web documents. These programs add extra characters to the file which can interfere with the web page displaying correctly.

Basic XHTML Document

In this course, we will demonstrate XHTML documents with yellow code
windows. Some code might be **highlighted** for discussion purposes:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Page Title Goes Here</title>
</head>
<body>
<!--  Page content goes here -->
</body>
</html>
```

This document contains all the essential components of a functional web
page. It can be copied and then modified to begin building a new web page.
From now on, we'll refer to this template as the skeleton file.

We'll start having fun designing our own web pages very soon, but first we have to go
over a few basic components of this skeleton page, beginning with the DOCTYPE.

The DOCTYPE

A DOCTYPE declaration must be specified on the first line of each web document:

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

The DOCTYPE tells the web browser which version of HTML the page is written in. In this class, we will be using 'XHTML Transitional', which allows us a little flexibility.

Another possibility is to use an 'XHTML Strict' DOCTYPE. It is less flexible and does not permit the use of some older HTML components.

The <html> Element

Immediately following the DOCTYPE declaration is the <html> element:

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

The <html> element tells the browser that the page will be formatted in HTML and, optionally, which world language the page content is in.

If our page were in a language other than English ("en"), we would change the two-letter code to the appropriate language code.

Don't worry about how complicated the DOCTYPE and <html> lines appear. We won't ever have to type them in. When we build a new web page using the skeleton file, these lines will already be there for us.

The <head> and <body> Elements

```
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Page Title Goes Here</title>
</head>
<body>
<!--  Page content goes here -->
</body>
```

The <head> element surrounds all the special "behind the scenes" elements of a web document. Most of these elements do not get displayed directly on the web page.

The <body> element surrounds all the actual content (text, images, videos, links, etc.) that will be displayed on our web page.

Just as with a person, the head always belongs on top of the body.

The <meta> Element

Immediately after the <head> line, we place this <meta> element:

```
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Page Title Goes Here</title>
</head>
```

The <meta> element provides information about the document to web browsers and search engines. This line declares that the document is encoded in the UTF-8 (Unicode) character set.
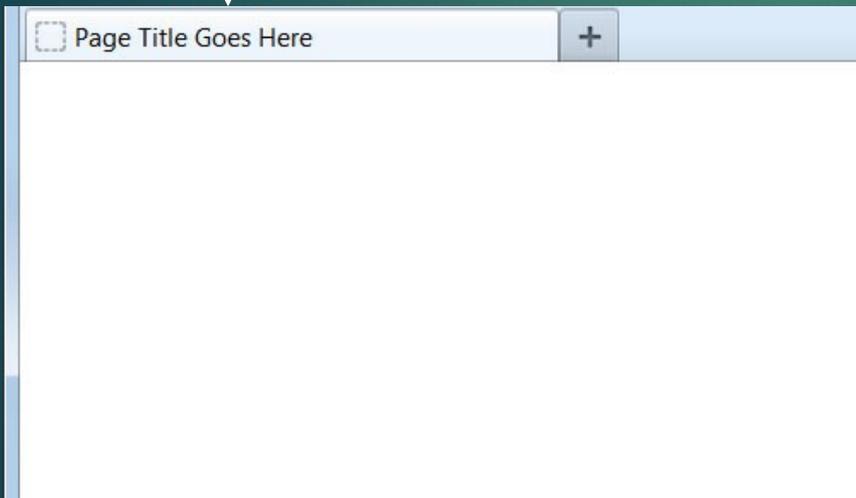
The UTF-8 character set is the dominant choice for documents on the internet. We will use it for all our web pages.

There can be multiple <meta> lines in the same web page. The <meta> element is often used to provide additional information such as page keywords, a page description, and the author(s) of a web document.

The <title> Element

The <title> element defines what text will show in the web browser's title bar:

```
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Page Title Goes Here</title>
</head>
```

Page Title Goes Here   +

The <title> element is always placed within the <head> section of the document.

Each page of your website should have a unique title.

Every web document must include one and only one instance of DOCTYPE, <html>, <head>, <body>, and <title>.

Lowercase Letters

In XHTML, all elements must be in lowercase letters:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Page Title Goes Here</title>
</head>
<body>
<!--  Page content goes here -->
</body>
</html>
```

The DOCTYPE declaration is <u>not</u> an element. Also, the lowercase rule applies only to the XHTML code within the < and > characters, not the actual content to be displayed on the web page.

Opening and Closing Tags

A tag is an XHTML "command" that tells the browser to perform some action:

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

<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Page Title Goes Here</title>


</head>
<body>
<!--  Page content goes here -->
</body>
</html>
```

In XHTML, all elements must have both an opening tag (shown in green) and a closing tag (shown in red). The closing tags begin with a forward slash.

The <meta> element is unusual in that it is a **self-closing element**. We'll see more examples of this element type in the next lesson.

The XHTML tags that surround the actual content of a page are collectively known as **markup**. Markup is not visible when the page is viewed in a browser.

Nesting Elements

When we write XHTML code, we must take care that elements are properly nested:

```
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Page Title Goes Here</title>
</head>
<body>
<!-- Page content goes here -->
</body>
```
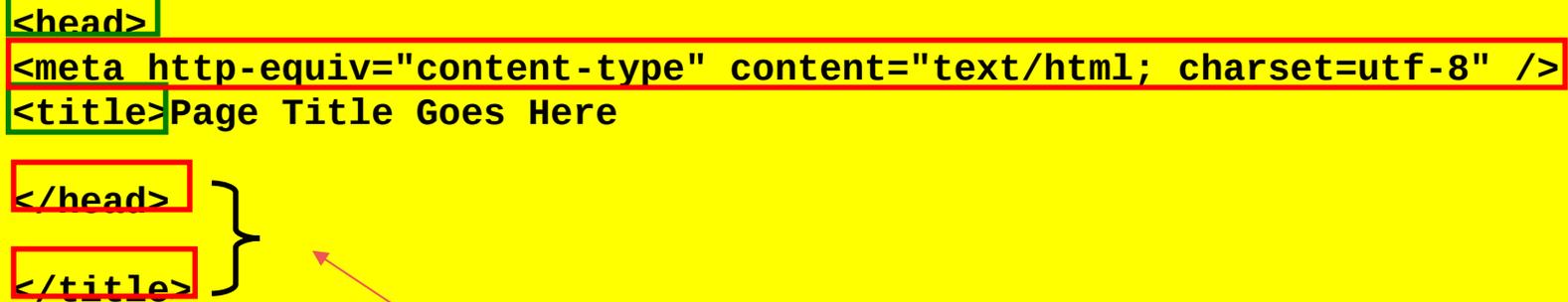
Nesting means that an element is completely contained within another element.

In these lines of code, the <meta> and <title> elements are nested within the <head> element.

Nesting Error Example

One of the most common errors committed by beginning web designers is to overlap HTML elements, resulting in a nesting error:

```
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Page Title Goes Here

</head>

</title>
```

In the above example, the <head> element was closed before the <title> element was closed. A nesting error like this can cause serious problems when the web page is displayed in a browser.

Later in the course, we will use a "validation" tool that will hunt down and locate these types of mistakes for us, prior to our publishing a web page.

Line Breaks and Spaces

Browsers ignore line breaks and spaces. We can take advantage of this fact by placing blank lines and indents in our code to make it easier to organize and read:

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

  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title>Page Title Goes Here</title>
  </head>

  <body>
    <!--  Page content goes here -->
  </body>

</html>
```

We will standardize on using two spaces for each level of indenting.