Intro to XHTML

Intro to XHTML

Tags and formatting them:

Intro to XHTML

Tags and formatting them:

```
<html>
    your page content goes here
</html>
```

Intro to XHTML

Tags and formatting them:

<html>
    your page content goes here
</html>

This is a tag, and an <html> tag contains (almost) all the content on your page.

Intro to XHTML

Tags and formatting them:

<html>
    your page content goes here
</html>

This is a tag, and an <html> tag contains (almost) all the content on your page.

This type of tag is called a container tag, because it contains content. There are also simple tags, such as
, <hr> and <img>, that do not contain content. Most of the tags you'll use to code your site are container tags.

Elements

HTML and XHTML refer to tags and what they contain as elements. An element includes the start and end tags, and everything in between.

Elements

HTML and XHTML refer to tags and what they contain as elements. An element includes the start and end tags, and everything in between. For example:

This is a paragraph.
This is a link.

Elements

HTML and XHTML refer to tags and what they contain as elements. An element includes the start and end tags, and everything in between. For example:

This is a paragraph.
This is a link.

Start tag*	Element content	End tag**
	This is a paragraph	
	This is a link	

* or opening tag
** or closing tag

Hey wait. What's with that weird
 thing?

Hey wait. What's with that weird
 thing?

The
 tag and some others like it can function as empty elements.
 inserts a line break into your pages. <hr> inserts a horizontal rule.

Hey wait. What's with that weird
 thing?

The
 tag and some others like it can function as empty elements.
 inserts a line break into your pages. <hr> inserts a horizontal rule.

XHTML demands that empty elements be closed in the start tag, like so:

<hr />

Hey wait. What's with that weird
 thing?

The
 tag and some others like it can function as empty elements.
 inserts a line break into your pages. <hr> inserts a horizontal rule.

XHTML demands that empty elements be closed in the start tag, like so:

<hr />

The <img> tag is another tag that functions like this. Images work oddly in html, but what you need to know is that in xhtml, img tags have attributes (more on this later) but don't themselves contain content.

Hey wait. What's with that weird
 thing?

The
 tag and some others like it can function as empty elements.
 inserts a line break into your pages. <hr> inserts a horizontal rule.

XHTML demands that empty elements be closed in the start tag, like so:

<hr />

The <img> tag is another tag that functions like this. Images work oddly in html, but what you need to know is that in xhtml, img tags have attributes (more on this later) but don't themselves contain content.

HTML element syntax:

HTML element syntax:

An HTML element starts with a **start (or opening) tag** and ends with an **end (or closing) tag**.

HTML element syntax:

An HTML element starts with a **start (or opening) tag** and ends with an **end (or closing) tag**.

The **element content** is everything in between the start and the end tag.

HTML element syntax:

An HTML element starts with a **start (or opening) tag** and ends with an **end (or closing) tag**.

The **element content** is everything in between the start and the end tag.

Some HTML elements have **empty content**. Those are closed in the start tag.

HTML element syntax:

An HTML element starts with a **start (or opening) tag** and ends with an **end (or closing) tag**.

The **element content** is everything in between the start and the end tag.

Some HTML elements have **empty content**. Those are closed in the start tag.

Most HTML elements can have **attributes**, about which more later.

HTML element syntax:

An HTML element starts with a **start (or opening) tag** and ends with an **end (or closing) tag**.

The **element content** is everything in between the start and the end tag.

Some HTML elements have **empty content**. Those are closed in the start tag.

Most HTML elements can have **attributes**, about which more later.

Start tag	Element content	End tag
	This is a paragraph	
	This is a link	

Block and inline elements

Most HTML elements can be defined as either **block level** or **inline** elements.

Block and inline elements

Most HTML elements can be defined as either **block level** or **inline** elements.

Generally speaking, block level elements end and start a new line in the browser, while inline elements do not.

Block and inline elements

Most HTML elements can be defined as either **block level** or **inline** elements.

Generally speaking, block level elements end and start a new line in the browser, while inline elements do not.

Common block level elements:
<h1>, <p>, <ol>, <ul>,

Block and inline elements

Most HTML elements can be defined as either **block level** or **inline** elements.

Generally speaking, block level elements end and start a new line in the browser, while inline elements do not.

Common block level elements:
<h1>, <p>, <ol>, <ul>,

Common inline elements:
<a>, <img>,
,

Block and inline elements

Most HTML elements can be defined as either **block level** or **inline** elements.

Generally speaking, block level elements end and start a new line in the browser, while inline elements do not.

Common block level elements:
<h1>, <p>, <ol>, <ul>,

Common inline elements:
<a>, <img>,
,

Note: many inline elements are/were used for presentation, which is not ideal. We'll come back to block and inline elements when we learn CSS.

Nested HTML elements:

Nested HTML elements:

```
<html>
    <body>
        <p>This is a paragraph right here.</p>
    </body>
</html>
```

Nested HTML elements:

```
<html>
    <body>
        <p>This is a paragraph right here.</p>
    </body>
</html>
```

HTML elements can be nested within one another, and they generally are. The example above contains three HTML elements. Anyone want to guess what they are?

Nested HTML elements:

```
<html>
    <body>
        <p>This is a paragraph right here.</p>
    </body>
</html>
```

HTML elements can be nested within one another, and they generally are. The example above contains three HTML elements. Anyone want to guess what they are?

The elements are: html (to define the page); body (to define the body content) and p (to define a paragraph).

Nested HTML elements:

```
<html>
    <body>
        <p>This is a paragraph right here.</p>
    </body>
</html>
```

HTML elements can be nested within one another, and they generally are. The example above contains three HTML elements. Anyone want to guess what they are?

The elements are: html (to define the page); body (to define the body content) and p (to define a paragraph).

The indents are standard practice to define different levels of the document. They don't affect how a browser reads the content, but how you'll read it.

HTML attributes:

HTML attributes:

HTML elements can have **attributes**, and often do.

HTML attributes:

HTML elements can have **attributes**, and often do.

Attributes provide **extra information** about the element.

HTML attributes:

HTML elements can have **attributes**, and often do.

Attributes provide **extra information** about the element.

Attributes are always specified in the **start tag**.

HTML attributes:

HTML elements can have **attributes**, and often do.

Attributes provide **extra information** about the element.

Attributes are always specified in the **start tag**.

Attributes come in **name/value pairs**. The proper syntax is:

name=”value”

Attribute example:

Attribute example:

A link to Google.

Attribute example:

A link to Google.

Tag: a

Attribute example:

A link to Google.

Tag: a
Attribute: href=”http://www.google.com”

Attribute example:

A link to Google.

Tag: a
Attribute: href=”http://www.google.com”
Name: href

Attribute example:

A link to Google.

Tag: a
Attribute: href=”http://www.google.com”
Name: href
Value: http://www.google.com

Attribute example:

A link to Google.

Tag: a
Attribute: href=”http://www.google.com”
Name: href
Value: http://www.google.com

Tips:
* The name is separated from the value by an equals sign (=)
* The value should always have quotes around it (“ “).
* Double or single quotes work equally well, but if you have double quotes inside an attribute, they must be nested inside single quotes.
    example: name=’Eldrick “Tiger” Woods’

Comments:

Comments:

Comments are parts of an HTML or XHTML document that are read by the browser, but don't show up in a page. They serve two main functions:

Comments:

Comments are parts of an HTML or XHTML document that are read by the browser, but don't show up in a page. They serve two main functions:

1) to describe the document content and structure to other people (or serve as reminders for yourself);

2) to tell a browser to execute a script, stylesheet or something else based on the version of the browser.

Comments:

Comments are parts of an HTML or XHTML document that are read by the browser, but don't show up in a page. They serve two main functions:

1) to describe the document content and structure to other people (or serve as reminders for yourself);

2) to tell a browser to execute a script, stylesheet or something else based on the version of the browser.

They also are used by programmers to quickly turn on or off pieces of code without actually deleting them from the program.

Comments:

The comment syntax for HTML is as follows:

<!-- this is a comment -->

Comments:

The comment syntax for HTML is as follows:

<!-- this is a comment -->

This is not a valid comment:

<!-- this -- this is a comment -->

Differences between HTML and XHTML:

Differences between HTML and XHTML:

XHTML is a **stricter version** of HTML. There's a lot of "bad" HTML on the Internet. XHTML is designed as a markup language where the code must be marked up properly and well formed.

Differences between HTML and XHTML:

XHTML is a **stricter version** of HTML. There's a lot of "bad" HTML on the Internet. XHTML is designed as a markup language where the code must be marked up properly and well formed.

Major differences:

Differences between HTML and XHTML:

XHTML is a **stricter version** of HTML. There's a lot of "bad" HTML on the Internet. XHTML is designed as a markup language where the code must be marked up properly and well formed.

Major differences:

* XHTML elements must be **properly nested**

Differences between HTML and XHTML:

XHTML is a **stricter version** of HTML. There's a lot of "bad" HTML on the Internet. XHTML is designed as a markup language where the code must be marked up properly and well formed.

Major differences:

* XHTML elements must be **properly nested**
* XHTML elements must **always be closed**

Differences between HTML and XHTML:

XHTML is a **stricter version** of HTML. There's a lot of "bad" HTML on the Internet. XHTML is designed as a markup language where the code must be marked up properly and well formed.

Major differences:

* XHTML elements must be **properly nested**
* XHTML elements must **always be closed**
* XHTML elements must be in **lowercase**

Differences between HTML and XHTML:

XHTML is a **stricter version** of HTML. There's a lot of "bad" HTML on the Internet. XHTML is designed as a markup language where the code must be marked up properly and well formed.

Major differences:

* XHTML elements must be **properly nested**
* XHTML elements must **always be closed**
* XHTML elements must be in **lowercase**
* XHTML documents must have **one root element**

Properly nesting elements:

Properly nesting elements:

In HTML, some elements can be improperly nested within one another:

This text is bold and italic.

Properly nesting elements:

In HTML, some elements can be improperly nested within one another:

This text is bold and italic.

In XHTML, all elements must be properly nested within one another:

This text is bold and italic.

Properly nesting elements:

In HTML, some elements can be improperly nested within one another:

This text is bold and italic.

In XHTML, all elements must be properly nested within one another:

This text is bold and italic.

* note: we won't use HTML tags to actually style content; this is just an example

Properly closing elements:

Properly closing elements:

In HTML, non-empty elements can lack closing tags:

This is a paragraph.
This is another paragraph.

In XHTML, all elements must be closed:
This is a paragraph.
This is another paragraph.

Properly closing elements:

Properly closing elements:

In HTML, empty elements can also lack closing tags:

A break:
A horizontal rule: <hr>
An image: <img src=”lolcat.gif” alt=”Cat with melon on its head”>

Properly closing elements:

In HTML, empty elements can also lack closing tags:

A break:
A horizontal rule: <hr>
An image: <img src="lolcat.gif" alt="Cat with melon on its head">

In XHTML, empty elements must be closed:

A break:
A horizontal rule: <hr />
An image:

XHTML elements must be in lower case

XHTML elements must be in lower case

In HTML, mixed case works just fine:

```
<BODY>
    <p>This is a paragraph
</BODY>
```

XHTML elements must be in lower case

In HTML, mixed case works just fine:

```
<BODY>
    <p>This is a paragraph
</BODY>
```

In XHTML, you must use lower case:

```
<body>
    <p>This is a paragraph</p>
</body>
```

XHTML documents must have one root element

XHTML documents must have one root element

All XHTML elements must be nested within the <html> root element. Child elements must be in pairs and correctly nested within their parent element.

XHTML documents must have one root element

All XHTML elements must be nested within the <html> root element. Child elements must be in pairs and correctly nested within their parent element.

The basic structure of a document is:

```
<html>
    <head>
        <title>
        </title>
    </head>
    <body> ... </body>
</html>
```

Mandatory XHTML elements

An XHTML page must have a DOCTYPE declaration.

Mandatory XHTML elements

An XHTML page must have a DOCTYPE declaration.

What's a doctype declaration? It's a way for browsers to know what type of (x)html you're using, so they can properly process your page's content.

Mandatory XHTML elements

An XHTML page must have a DOCTYPE declaration.

What's a doctype declaration? It's a way for browsers to know what type of (x)html you're using, so they can properly process your page's content. Here's the one we'll be using for your initial XHTML pages:

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

Mandatory XHTML elements

An XHTML page must have a DOCTYPE declaration.

What's a doctype declaration? It's a way for browsers to know what type of (x)html you're using, so they can properly process your page's content. Here's the one we'll be using for your initial XHTML pages:

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

Don't worry about remembering that right now. You can find doctype declarations on the W3C site, or bookmarked on my Delicious account.

Mandatory XHTML elements

An XHTML page must have a DOCTYPE declaration.

What's a doctype declaration? It's a way for browsers to know what type of (x)html you're using, so they can properly process your page's content. Here's the one we'll be using for your initial XHTML pages:

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

Don't worry about remembering that right now. You can find doctype declarations on the W3C site, or bookmarked on my Delicious account.

OK. That's enough leturing; let's get out of here and start building some web pages.