This document is meant to serve primarily as a guide to using XHTML (Extensible Hypertext Markup Language) and CSS (Cascading Style-sheets) to create web pages, sites and interfaces.

There are three portions to this guide. The first portion details what exactly XHTML is, its purpose, its history and its future.

The second portion walks through the various parts of a correctly written XHTML file. How to markup content within an XHTML page and using hypertext and media within an XHTML webpage will be discussed as well.

The third portion of this guide begins to focus on using Cascading Style Sheets to visually style and display the markup created in an XHTML Document.

**What is XTHML?**

XHTML, or Extensible Hypertext Markup Language is one of the most modern building materials for creating web pages. As a computer programming language that is supported by web browsers and a growing number of other applications, XHTML’s purpose it to provide formatting to content using Markup.

**Note:** Visually styling a web document is not a direct responsibility of XHTML, rather CSS (Cascading Style Sheets) are used to define the visual style of the markup used in an XHTML document.

**What is Markup?**

Markup is the process of providing context for pieces of information. Markup provides semantic weight to elements within information while still keeping the association between all of the elements within information. If we think about a physical letter that would be sent to someone as an example, the letter itself is one piece of information, but there are various elements that make up a letter such as the Greeting, Address, Body, Signature and so forth. All of these elements are understood to mean different things and that understanding comes from the formatting associated with them.

**What is the difference between HTML, XTHML, and XML?**

XHTML is a transitional step from previous markup languages such as SGML (Standard Generalized Markup Language) and HTML (Hypertext Markup Language). XHTML is very similar to HTML as it shares the same expressive possibilities, but has a stricter syntax (more rules/practices to follow). Whereas HTML was an application of SGML, a very flexible markup language, XHTML is an application of XML, a more restrictive subset of SGML. Currently web browsing technologies support XHTML and HTML, but cannot however support XML in it’s original form. XML must be transformed and delivered into a document browsing technologies can understand, the general choice is XHTML.

**The benefits of XHTML - A Sound Structure**

Because XHTML is based on a stricter language (XML), there is a much stricter format that must be followed when marking up information and presenting it in an XHTML file.

A stricter format adopted from XML allows for the separation of content and the visual presentation of this content. This has many benefits as an XHTML page can be visually styled by multiple CSS documents actively or invisibly providing users...
with formatting options based on their particular situations. This comes in handy in some of the following situations:

 › Printing Documents
 › Projection/Presentation Situations
 › People with disabilities/accessibility issues

This stricter formatting of XHTML also allows the emergence of a systematic troubleshooting and validation service. This creates a standard practice of coding and marking information up.

Creating XHTML

XTHML can be written in any text editing software or in specially designed applications such as Macromedia Dreamweaver. Generally XHTML files are saved as .html files. But, dynamically produced pages such as .php files can also use XHTML.

XHTML Markup

XHTML markup is written in the form of tags. Tags are commands that tell the web agent how to process the information (content, media, and metadata) of the webpage.

Think of tags as wrapper on pieces of candy. When looking at the wrappers of candy you know what flavors and how a candy would taste. A web agent, in many cases a web browser, knows how information should be processed because of the tag that it is surrounded by. This gives context to the information being presented just as the wrapper on a piece of candy gives you understanding as to what type of candy it is or may be.

A tag command is written within angle brackets ("<" & ">") and most work like bookends with an opening tag and a closing tag. What sits between the opening and closing tags is what is effected by that tag. The difference between opening and closing tag is that the command in the closing tag is preceded by a forward slash (" / ").

Note: As we will see, there are some variations to this rule where an XHTML tag can close itself, these are called ‘self closing tags”. However, all tags in an XHTML document must be closed.

Document Structure

An XHTML document has a pre–formatted document structure. All this means is that there are certain pieces that every XHTML page must have in order to be properly formatted as XHTML (which will give all of the benefits listed above).

An XHTML Document needs to have the following elements within it:

A Document Type (DocType) definition

The first thing placed in an XHTML file, the DocType definition tells the web agent (A web browser, mobile phone, other online devices) what type of markup language a page is using. The agent then takes that information into consideration when rendering the page and its information for the user. A DocType looks like the following and is placed at the beginning of the XHTML file:
<!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">

Note: The code behind a DocType is a bit more complicated and can be examined closer by some of the links at the end of this section of the guide.

There are various DocType Definitions each with their own function. HTML has its own document type definition as does RSS (Rich Site Summary). The point of a DocType is provide the framework for what type of code can and can’t be placed below and to prepare a user’s web agent to handle the markup in the best manner possible.

Note: The <html> tag that is opened in the example above must be closed after everything else on the page. Thus everything else to follow would be wrapped in an <html></html> tag as every other tag.

Head Tag

The next piece of standard XHTML that is needed is the Head Tag portion of the webpage. The Head Tag contains information about the page such as its title, any extra information (called metadata) about the page as well as references to external files that provide functionality to the page. These files include references to Cascading Style Sheets (CSS) which contains the rules on how the page is visually styled as well as JavaScript which adds additional web behaviors such as opening new windows.

The Head Tag generally looks like the following, with the paths to external CSS files changing depending on where they are stored.

<head>
<title>The Web Page’s Title</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="author" content="Author’s Name" />
<meta name="Description" content="A Description of the Webpage"/>

<style type="text/css" media="screen">
<!--[CDATA[
@import "css.css";-->
</style>
</head>

Body Tag

The Body tag houses all of visible of an XHTML web-page. This is where the bulk of the markup will occur. For now a blank body tag looks like the following.

<body>
This is where the content of a webpage would go and it will be marked up using other XHTML tags.
</body>

Note: When writing XHTML, line breaks in the code do not affect visual display or formatting.

If all of these elements are placed together, they look something like this:

<!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>
<title>The Web Page’s Title</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="author" content="Author’s Name" />
<meta name="Description" content="A Description of the Webpage"/>

<style type="text/css" media="screen">
<!--[CDATA[
@import "css.css";-->
</style>
</head>

<body>
This is where the content of a webpage would go and it will be marked up using other XHTML tags.
</body>

Note: You can make notes and comments in XHTML by placing a "<!--" at the beginning of your comment and a "-->" at the end of your comment. A comment will not be seen by the web agent and is used commonly to keep track of things within the code.
Common XHTML Markup

The following are tags that are used within the <body> of an XHTML document.

**Paragraph**

Denotes that the contents within the tag are a paragraph of text.

<p>This is a paragraph</p>

**Strong and Emphasis (Bold and Italics)**

To place emphasis on content there are two ways to do so, to bold content, you would place strong tags around the content. To italicize content you would place emphasis tags (<em>) around it.

<strong>This is bolded text</strong>

<em>This is italicized text</em>

**Lists**

Lists are used to display short amounts of information that relate to each other in one sense or another, much as they do in real-life situations. There are various kinds of lists used in XHTML, The **ordered** and the **unordered** lists are two of the most common. The **ordered** is listed with numbers, the **unordered** is bulleted. Two create these lists you need two tags:

1.  <ul> or <ol> - Depending on which list you want. These are needed to open and close your list.

2.  <li> - Within the above tag, the list item precedes each item you are listing.

Here’s an example with the days of the week listed in an unordered list:

<ol>
  <li>Monday</li>
  <li>Tuesday</li>
  <li>Wednesday</li>
  <li>Thursday</li>
  <li>Friday</li>
</ol>

Here’s an example of a grocery list arranged by most important item to least important in an ordered list:

<ol>
  <li>Eggs</li>
  <li>Milk</li>
  <li>Bread</li>
  <li>Tomatoes</li>
  <li>Figs</li>
</ol>

**Links**

To create a hyperlink to another webpage or document you would use the anchor tag (<a>), which is made up of 2 different parts.

1.  Anchor <a> - This is the opening and closing tag, what makes the text clickable.

2.  HyperText Reference "href" – An attribute to the opening anchor tag that tells where to link to.

If you wanted to have a sentence that linked to another website, here is what it would look like in XHTML:

<a href="http://www.anotherwebsite.com/">This link will take you to another website</a>

**Images**

Images can be placed in an XHTML webpage using the following two part <img> tag can be used to reference images inside of XHTML.

<img src="flag.gif" />

Note: The <img> tag is one of a few “self-closing tags” in XHTML which do not have an end tag. In this case there is no such thing as a </img> tag, but rather the tag is closed by a “/” at the end of the initial tag, <img src="/">. Please see above in this guide for further information.
For information on more specific XHTML markup you may consult the following list of additional resources.

**Divisions**

Divisions are logical separations of content within a page. If for instance there was both a piece of content that had numerous paragraphs of information which pertained directly to the topic of your web-page, you may want to separate that from a list of links to supplemental topics that are not directly related to the main content of the page and would be placed visually on the right of the page. This type of content separation can be obtained by placing each group of content in a `<div>` tag. See the example below:

```html
<div><p>This is my main content</p><p>This is more of my main content</p></div>
<div><p>This is a set of links to supplemental information</p><ul><li><a href="link">Link 1</a></li><li><a href="link">Link 2</a></li><li><a href="link">Link 3</a></li></ul></div>
```

**IDs and Classes**

Once an XTHML tag has been written, you can then assign one id and as many classes as you want to the tag if you wish. IDs and classes are used to further distinguish content and help in the visual styling of content with CSS. The way you would add a class or ID can be seen in the example below.

```html
<p class="food, frozen" id="peas">This is a paragraph about frozen peas in the supermarket.</p>
```

**Additional XHTML Resources**

W3Schools XHTML Tutorial
http://www.w3schools.com/xhtml/

World Wide Web Consoritum (W3C) Official XHTML documentation
http://www.w3.org/MarkUp/

World Wide Web Consoritum (W3C) XHTML/HTML Validator
http://validator.w3.org

Too Easy XHTML – Lowter’s Guide
http://www.lowter.com/articles/178?PHPSESSID=f0885ba6035254fde1679764ab2968

**Styling XHTML with Cascading Style Sheets**

All of the presentational information that is contained and displayed through a webpage are delivered not by the XHTML of a page, but by the CSS (Cascading Style Sheets) that are linked in the `<head>` tag of an XHTML file.

CSS handles both physical layout, placing elements next to each other the page and also the look of things, such as color, font-size and borders.

Using CSS allows for far greater accessibility efforts and allows designers and creators of webpage to carefully and universally tailor elements of webpages according to their various audience's needs.

Generally the best practice is to keep CSS as external files that are referenced inside XHTML files but stored outside somewhere else. This makes things easier to manage and maintain.

CSS files also have a certain format that they need to be written in, however this format is more of a list of definitions and can be as extensive or simple as the author chooses.

Below is an example of a basic CSS definition which is telling a web agent to place a border around any paragraph and also setting a width of a paragraph to 80% of the entire window.
The general anatomy of a CSS definition or rule can be seen below. All parts of the definition must be in place for the rule to work and function properly.

A CSS file is merely an entire list of these definitions. There are numerous combinations of styles and definitions you can use to style XHTML. We will explore the idea of the cascade of styles along with some other CSS principles in class further.

Further Information / Questions, Comments, etc.

If you have any questions concerning the creation of this guide or anything included below, you may contact me via the following information:

Brian Talbot
Web Designer
Simmons College Web Services
300 The Fenway
Boston, MA 02115

617–521–2678
brian.talbot@simmons.edu

More Information on Cascading Style Sheets

W3Schools CSS Tutorial
http://www.w3schools.com/css/

Styleguide CSS property and value encyclopedia
http://www.stylegala.com/features/css-reference

World Wide Web Consortium (W3C) Official CSS Documentation
http://www.w3.org/Style/CSS/

World Wide Web Consortium (W3C) CSS Validator
http://jigsaw.w3.org/css-validator/

An Example of the Power of CSS – CSS Zen Garden
http://csszengarden.com/