CSP2103: Markup Languages
Workshop 2: Building Structure with XHTML

<h1>Learning Outcomes</h1>

At the completion of this workshop and related learning activities, you will be able to:

1. Create a strict DTD XHTML 1.0 document;
2. Evaluate a .html document for XHTM/HTML 4.01 conformity,
2. Create an advanced HTML document;

You should now have a good grasp of markup fundamentals encompassing XML and HTML. The current standard for Web development today is XHTML 1.0 (Revised), but most Internet applications development continues to be driven by HTML 4.01. Why?

The fact is that most coders working today grew up with HTML and find it easier to work with HTML 4.01 or the transitional DTD. HTML is a warm cuddly bear and who wants to forsake it for the cold demanding world of strict DTD XHTML or XML? Most clients don't demand XHTML compliance and are not even aware that a new emerging world exists beyond HTML.

We should not be surprised by this finding. Although Information Technology (IT) is popularly perceived to be a world of light speed innovation, the world of Business Applications Development (BAD) is a very different world. Business wants the job done at the lowest cost and with least complication. It is disinterested in technology excellence and will willingly and enthusiastically support legacy systems. Just think- How many desktops around town are still using Windows 98? The Macintosh has spearheaded innovation in desktop computing for years, but is ignored by business.

Very little HTML is taught directly in CSP2103. Established habits are not easily discarded in favour of better approaches. This is also a unit about markup languages, of which HTML is only one. Overwhelming adoption has made this standard important, so much so that it has become a drag on Internet applications development overall. We will demonstrate our commitment to XHTML in this workshop by building an XHTML document against the strict DTD, checking it for well formedness and validating it against the W3C validation service.

All of this will be done in a text editor. There is no place in this course for WYSIWYG editors. If your assignment code is found to include artifacts from a WYSIWYG editor, this will result in automatic failure of the assignment. You cannot learn the syntax of markup languages in graphical environment when all of these decisions are made for you by software. You cannot understand the basics of good structure and you cannot debug pages that generate run time errors, unless you know what is happening under the bonnet.

<h1>What is the basic XHTML document?</h1>

In the words of W3C:

> The Extensible HyperText Markup Language (XHTML™) is a family of current and future document types and modules that reproduce, subset, and extend HTML, reformulated in XML. XHTML Family document types are all XML-based, and ultimately are designed to work in conjunction with XML-based user agents. XHTML is the successor of HTML, and a series of specifications has been developed for XHTML.
>
> (W3C, 2003)

The basic XHTML document therefore begins with an XML declaration. The basic XML document is made up of the prolog and the document proper. The prolog contains the document type declaration, any attributes associated with it (e.g. whether it employs a PUBLIC or local SYSTEM DTD), the URL for this DTD and processing instructions for the document. The document proper begins with the root element (remember all XML documents must have a root element).

Here is the basic template for the XHTML document we will use today:-

```
<?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>
          <title>Enter the title of your XHTML document here</title>
        </head>
        <body>
          <p>Enter the body text of your XHTML document here</p>
        </body>
</html>
```

The root element <html> is provided in line 4. A namespace attribute is added to this element. A namespace organizes the elements and attributes of an XML document into separate groups. Without the namespace attribute, the reuse of common element names (e.g. 'company', 'employee', 'firstname') by different XML documents would become a fundamental source of invalidity. You can liken the role of namespace to that of a glossary or restricted vocabulary in language.

<h1>Activity1- Creating an XHTML template </h1>

1. In your text editor, create the basic XHTML template.
2. Save this document as activity_1.html and view it in IE.
3. Validate this simple document using W3C's validation service located at:- http://validator.w3.org/

4. Note any error messages produced in the validation response page. Fix any coding errors and re-validate the document.

The error message describes in detail what is currently wrong with the structure of your XML document. Now that we have a template, we can create a document head and a document body. Using XML concepts, we can think of the head as the parent element. The following table describes child elements that maybe used in conjunction with the head element:

Table 2-5 Child elements of the <head> element

Element	Description
<base>	Specifies a base URL for all of a document's relative links
<link>	Defines the relationship between linked documents
<meta>	Defines metadata about a Web page
<script>	Contains commands for scripting languages such as JavaScript and VBScript
<style>	Defines the style information for a specific element
<title>	Contains text that appears in a browser's title bar

(Gosselin, 2003,p.62)

<h1>Activity 2-Creating the Document Head</h1>

The document we are building today is a course results transcript that will hold your course results in Internet Computing in XHTML format. We will validate this XHTML document against the strict DTD for XHTML 1.0 and make any necessary corrections.

Assignment One builds the same document as an XML document with its own Document Type Definition (DTD). By undertaking the Challenge Exercise at the end of each Workshop, you can build your assignment solution for submission in Week 6 at the conclusion of each week's learning programme.

We will begin the XHTML exercise by adding some child elements to the <head> element.

Activity 2.1

Following instructions from your tutor:

1. Add a title;
2. Add meta tags for the author, description, keywords and copyright; and
3. Add a comment at the top of the page above the document declaration

When you have completed the document, save it as activity_2_1.html. Your document should look something like this:

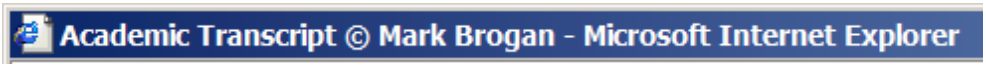
```
<?xml version="1.0" encoding="UTF-8"?>

<!--
This page was written against a strict XHTML DTD as an exercise
in the unit CSP2103-->
```

```
<!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>XHTML/XML Wizard tops Internet Computing! </title>
                  <meta name="author" content="Mark Brogan"></meta>
                  <meta name="description" content="Academic transcript of my results in B28 Internet
Computing"></meta>
          <meta name="keywords" content="Mark Brogan, Internet Computing, Edith Cowan
University"></meta>
          <meta name="copyright" content="(c) Mark Brogan, 2003"></meta>
        </head>
        <body>
          <p>Enter the body text of your XHTML document here</p>
        </body>
</html>
```

Activity 2.2

This is a very minimal document that could easily be improved upon. Using your textbook (Gosselin) or the Web find the resources required to amend your header to meet the following functional requirements:

1. To allow robot indexing of this page, but to disallow robot indexing of linked pages;
2. Add a special character to the <title> element to enable the following title bar:



<h1>Activity 3- Creating the document body</h1>

In the basic XHTML document, the <body> element is used for the substantive content of the document. Elements used in the body can comprise 1) block-level elements that give a Web page its structure and appear in their own line and 2) in-line elements that describe the text that appears on a page and are used inside block-elements. The following table from your text (Gosselin) describes the block-level elements available in the strict DTD:-

Table 2-2 Block-level elements available in the strict DTD

Element	Description
<address>	Address
<blockquote>	Block quotation
<del>	Deleted text
	Generic block-level container
<dl>	Definition list
<fieldset>	Form control group
<form>	Interactive form
<h1> - <h6>	Heading elements
<hr>	Horizontal rule
<ins>	Inserted text
<noscript>	Alternate script content
	Ordered list
<p>	Paragraph
<pre>	Preformatted text
<table>	Table
	Unordered list

Text formatting elements are in-line elements. The following is list of commonly used text formatting elements:

Table 4-1 Text-formatting elements

Element	Description
<b>	Formats text in boldface type
<big>	Formats text in a larger font
<i>	Formats text in italic type
<small>	Formats text in a smaller font
<sub>	Formats enclosed text as a subscript
<sup>	Formats enclosed text as a superscript
<tt>	Formats enclosed text as teletype or monospaced text

We will now add the basic structure in the <body> to hold our course results. At this stage we will use a combination of block-level and in-line elements that describe procedural markup, but provide no guide to meaning:-

Activity 3

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

<!--
This page was written against a strict XHTML DTD as an exercise
in the unit CSP2103-->

<!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>Academic Transcript &#169; Mark Brogan</title>
          <meta name="author" content="Mark Brogan"></meta>
          <meta name="description" content="Academic transcript of my results in B28 Internet
Computing"></meta>
          <meta name="keywords" content="Mark Brogan, Internet Computing, Edith Cowan
University"></meta>
          <meta name="copyright" content="(c) Mark Brogan, 2003"></meta>
          <meta http-equiv="content-type" content="text/html; charset=UTF-8"></meta>
          <meta name="robots" content="index, nofollow"></meta>
        </head>
<body>
<h4>Last Name: Brogan </h4>
<h5>First Name: Mark </h5>
<h4>Student No. 338865</h4>
<h1>B:28 Internet Computing</h1>
<h2>Semester 1, 2003</h2>
```

```
<blockquote>
        <h3>CSG1132 Communicating in an IT Environment</h3>
        <p><b>Result=</b>66 <b>Grade Score</b>= CR</p>
        <h3>CSP1150 Programming Principles</h3>
        <p><b>Result=</b>80 <b>Grade Score=</b> HD</p>
        <h3>ENS1161 Computer Fundamentals</h3>
        <p><b>Result=</b>72 <b>Grade Score=</b> D</p>
         <h3>CSI1241 Systems Analysis</h3>
        <p><b>Result=</b>75 <b>Grade Score=</b> D</p>
</blockquote>
<h2>Semester 1, 2003</h2>
  <blockquote>
        <h3>CSG2130 Applied Communications</h3>
         <p><b>Result=</b> 66 <b>Grade Score=</b> CR</p>
        <h3>CSP1244 Visual Programming</h3>
        <p><b>Result=</b>58 <b>Grade Score=</b> I</p>
        <h3>CSP2343 Operating Systems</h3>
        <p><b>Result=</b>75 <b>Grade Score=</b> D</p>
        <h3>CSI2341 Systems and Database design</h3>
        <p><b>Result=</b>75 <b>Grade Score=</b> D</p>
</blockquote>
</body>
</html>
```

When you have completed keying in the text, save the file as activity_3_1.html, and view it in IE. Once again check the file for strict XHTML compliance at:- http://validator.w3.org/

Make any necessary connections.

<h1>Activity 4- The same data in TABLES</h2>

No one produces XHTML (sic) that looks like activity_3_1.html these days. The following observations describe problems with this document:

- It is inherently redundant- much better to generate the document dynamically from a database using any one of a number of CGI technologies;
- White space and other layout features cannot be added with block-level and in-line elements (we require Tables and Styles to achieve this kind of appearance or layout control);
- In the absence of markup instructions, no two coders would markup the document in the same way (in other words, there is no way in which markup based on XHTML could be used for document interchange or as the basis of an ontology).

We begin with the adoption of TABLES that that allow layout control to be achieved over the data. Figure 6.1 from Gosselin, shows the basic <table> elements that can be used in XHTML:

Table 6-1 Table elements

Element	Defines
<caption>	A table caption
<col>	A table column
<colgroup>	A table column group
<tbody>	A table body
<td>	Table data
<tfoot>	A table footer
<th>	A table heading
<thead>	A table header
<tr>	A table row

CAUTION: Most of what you seen on the Web is layed out with TABLES. The main reason for this is that most Web development is still driven by the desktop PC and TABLES are fast compared with the alternative, Cascading Style Sheets (CSS). However, you should note that TABLES are problematic with other user agents (e.g. mobile phones & PDAs) and non-visual agents. For this reason, W3C prefers that we should adopt CSS. (CSS is developed in a later lecture.)

Now, in your text editor, key in the following data. When you have created the document, save it as activity_4_1.html and attempt to validate the document against the stict XHTML DTD located at: http://validator.w3.org/

Activity 4.1

Note the use of comments. You should not abandon coding good practice just because you are doing markup.

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

<!--
This page was written against a strict XHTML DTD as an exercise
in the unit CSP2103-->

<!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>Academic Transcript &#169; Mark Brogan</title>
          <meta name="author" content="Mark Brogan"></meta>
          <meta name="description" content="Academic transcript of my results in B28 Internet
Computing"></meta>
          <meta name="keywords" content="Mark Brogan, Internet Computing, Edith Cowan
University"></meta>
          <meta name="copyright" content="(c) Mark Brogan, 2003"></meta>
          <meta http-equiv="content-type" content="text/html; charset=UTF-8"></meta>
```

```
        <meta name="robots" content="index, nofollow"></meta>

    </head>
<body>
<table width="75%" border="1" cellspacing="2" cellpadding="2" summary= "Course level student record for B28.">
    <caption>Academic History no.338865</caption>
    <!-- Row 1 -->
  <tr align="left">
   <th>Last Name:</th>
   <td>Brogan</td>
   <th>First Name:</th>
   <td>Mark</td>
   <th>Student No:</th>
   <td>338865</td>
  </tr>
    <!-- An empty table row for spacing with cells filled with non-breaking spaces -->
  <tr>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
  </tr>
    <!-- Course Header -->
  <tr>
   <th>Course Code:</th>
   <td>B28</td>
   <th>Title:</th>
   <td colspan="3">Bachelor of Science (Internet Computing)</td>
  </tr>
  <tr>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
   <td> </td>
  </tr>
    <!-- Year and Semester Row -->
  <tr>
   <th width="20%">Year</th>
   <td width="15%">2003</td>
   <th>Semester:</th>
   <td>01</td>
```

Use the colspan attribute to merge data cells. You can see the effect in the IE view of the completed document below.

Oops!!

Academic History no.338865

Last Name:	Brogan	First Name:	Mark	Student No:	338865
Course Code:	B28	Title:	Bachelor of Science (Internet Computing)		
Year	2003	Semester:	01		
Unit Code	Unit Title			Result	Grade
CSG1132	Communicating in an IT Environment			66	CR
CSP1150	Programming Principles			80	HD
ENS1161	Computer Fundamentals			72	D
CSI1241	Systems Analysis and Design			75	D
Year	2003	Semester	02		
Unit Code	Unit Title			Result	Grade
CSG2130	Applied Communications			66	CR
CSP1244	Visual Programming			58	C
CSP2343	Operating Systems			75	D
CSI2341	Systems and Database Design			75	D

```
  <td> </td>
  <td> </td>
 </tr>
      <!-- Headers for unit codes and titles -->
 <tr>
  <th>Unit Code</th>
  <th colspan="3">Unit Title</th>
  <th>Result</th>
  <th>Grade</th>
 </tr>
      <!-- Begin the record proper -->
 <tr>
  <td>CSG1132</td>
  <td colspan="3">Communicating in an IT Environment</td>
  <td>66</td>
  <td>CR</td>
 </tr>
 <tr>
  <td>CSP1150</td>
  <td colspan="3">Programming Principles</td>
  <td>80</td>
  <td>HD</td>
 </tr>
 <tr>
  <td>ENS1161</td>
  <td colspan="3">Computer Fundamentals</td>
  <td>72</td>
  <td>D</td>
 </tr>
 <tr>
  <td>CSI1241</td>
  <td colspan="3">Systems Analysis and Design</td>
  <td>75 </td>
  <td>D</td>
 </tr>
 <tr>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
  <td> </td>
 </tr>
 <tr>
  <th>Year</th>
  <td>2003</td>
  <th>Semester</th>
```

```
    <td>02</td>
    <td>&amp;nbsp;</td>
    <td>&amp;nbsp;</td>
   </tr>
   <tr>
    <th>Unit Code</th>
    <th colspan="3">Unit Title</th>
    <th>Result</th>
    <th>Grade</th>
   </tr>
   <tr>
    <td>CSG2130</td>
    <td colspan="3">Applied Communications</td>
    <td>66</td>
    <td>CR</td>
   </tr>
   <tr>
    <td>CSP1244</td>
    <td colspan="3">Visual Programming</td>
    <td>58</td>
    <td>I</td>
   </tr>
   <tr>
    <td>CSP2343</td>
    <td colspan="3">Operating Systems</td>
    <td>75</td>
    <td>D</td>
   </tr>
   <tr>
    <td>CSI2341</td>
    <td colspan="3">Systems and Database Design</td>
    <td>75</td>
    <td>D</td>
   </tr>
  </table>
</body>
</html>
```

Did it validate successfully? Your tutor will assist you to de-bug your code. However you will not be able to escape the run time error generated by <th width="20%">Year</th>. In XHTML 1.0 you cannot attach a width attribute to a <th> or <td> tag. You control the width of columns by using the <colgroup> or <col> element.

Activity 4.2

2. Research the use of the <colgroup> and <col> tags in your text and amend the XHTML code for Acitivity 4.1 to make it XHTML 1.0 compliant. Save the amended

XHTML document as activity_4_2.html and validate this document against W3C's spec for XHTML;

3. XHTML specifies a number of useful table features not available in HTML 4.01. Using your text or Web research the following <table> elements and augment the TABLE created in 4.1 with the following elements:

   - <thead>
   - <tbody>
   - <tfoot>

<h1>Activity 5- Challenge Exercise (XML)</h1>

Now that we have successfully created the academic history in XHTML, we can use the understanding of document structure that we have developed to create a standalone XML document that does the same thing.

1. Review the Week One Workshop notes on XML basics;

2. Create a basic standalone XML document that includes no layout features, but uses the entities and attributes and values developed in the XHTML exercises;

3. Check this document for well-formedness in a non-validating parser and bring your solution along next week for class discussion.

M.Brogan
m.brogan@ecu.edu.au