

## Lecture 2: XHTML

### 1. XHTML-1.0-strict

<http://www.w3.org/TR/xhtml1/>

HTML is by far the most important mark up language. It predates XML and was in use long before XML was thought of.

XHTML is a recasting of HTML as an XML application.

The W3C recommendation XHTML 1.0 consists of a DTD that describes HTML.

#### 1.1 A sample page

```
<?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" xml:lang="en" lang="en">
  <head>
    <title>Sample Page</title>
  </head>
  <body>
    <p>A sample <b>web</b> page.</p>
  </body>
</html>
```

The <?xml ... ?> header is not mandatory – it tells us that we are using xml and the character set being used.

The !DOCTYPE tag permits validation. The PUBLIC value is used for a famous DTD – in this case it is followed by the actual URL of the DTD.

The namespace xmlns is applied to the html and all of its descendants.

## 1.2 XHTML1.0 DTD extract

```

<!ELEMENT html (head, body)>
<!ATTLIST html
  %i18n;
  id      ID      #IMPLIED
  xmlns   %URI;   #FIXED 'http://www.w3.org/1999/xhtml'
>
<!ENTITY % i18n
"lang     %LanguageCode; #IMPLIED
xml:lang  %LanguageCode; #IMPLIED
dir       (ltr|rtl)      #IMPLIED"
>
<!ENTITY % head.misc "(script|style|meta|link|object)*">
<!ELEMENT head (%head.misc;
  ((title, %head.misc;, (base, %head.misc;?) |
  (base, %head.misc;, (title, %head.misc;))))>
<!ELEMENT title (#PCDATA)>
<!ENTITY % block
  "p | %heading; | div | %lists; | %blocktext; | fieldset | table">
<!ENTITY % Block "(%block; | form | %misc;)*">
<!ELEMENT body %Block;>
<!ENTITY % fontstyle "tt | i | b | big | small ">
<!ENTITY % misc "noscript | %misc.inline;">
<!ENTITY % inline "a | %special; | %fontstyle; | %phrase; | %inline.forms;">
<!ENTITY % Inline "(#PCDATA | %inline; | %misc.inline;)*">
<!ELEMENT p %Inline;>
<!ATTLIST p
  %attrs;
>
<!ELEMENT b %Inline;>  <!-- bold font -->

```

```

html(head,body) →      html(head(%head.misc;,title),body)
                     →      html(head(title(#PCDATA)),body(%Block))
                     →      html(head(title(#PCDATA)),body(%block))
                     →      html(head(title(#PCDATA)),body(p))
                     →      html(head(title(#PCDATA)),body(p(%Inline)))

```

We can further resolve p(%Inline)

```

p(%Inline) →      p(#PCDATA|%inline)*
             →      p(#PCDATA,%inline,#PCDATA)
             →      p(#PCDATA,%fontstyle,#PCDATA)
             →      p(#PCDATA,b,#PCDATA)
             →      p(#PCDATA,b(%Inline),#PCDATA)
             →      p(#PCDATA,b(#PCDATA),#PCDATA)

```

## 2. HTML components

An html document includes:

- head section – here we typically have (in no particular order)
  - meta data (data about the page)
  - scripts (client-side javascript)
  - style (css – considered in later lectures)
- body section – here we typically have a sequence of block level items such as
  - div – pointless block level tag
  - p – classic paragraph tag
  - pre – fixed width character, line ends are respected
  - form – contains components for obtaining user responses
  - table – layout grids
  - dl – definition list
  - ul – un-ordered list
  - ol – ordered list
  - h1 – heading

The block level elements contain PCDATA and in-line elements such as

- a – anchors, hyper-links
- b – bold
- i – italics
- span – pointless inline
- tt – fixed width characters
- sub – subscript
- sup – superscript
- button

### 3. Modular XHTML 1.1

Version 1.1 of XHTML removes some legacy tags and introduces mechanisms to make it easy for designers to incorporate parts of xhtml into their own applications or to make their own variations.

#### 3.1 Using bits of xhtml

We might have an application that could benefit from a part of xhtml.

Xhtml includes a number of entities that may be included as part of another DTD. Examples are %Inline.mix, %Block.mix, %Flow.mix (both inline and block level). In addition there are entities such as %InlNoAnchor.mix

##### 3.1.1 Quiz without xhtml

L02Q1.xml
<pre>&lt;!DOCTYPE quiz SYSTEM "L02Q1.dtd"&gt; &lt;quiz&gt;   &lt;qu&gt;What is scott's password?&lt;/qu&gt;   &lt;d&gt;gnu&lt;/d&gt;   &lt;d&gt;lion&lt;/d&gt;   &lt;d&gt;tiger&lt;/d&gt; &lt;/quiz&gt;</pre>

L02Q1.dtd
<pre>&lt;!ELEMENT quiz (qu,d+)*&gt; &lt;!ELEMENT qu (#PCDATA) &gt; &lt;!ELEMENT d (#PCDATA)&gt;</pre>

The xml application works fine – but only allows questions in plain text. We might want to allow the qu tag to contain any in-line html elements – this would permit use of many formatting features from html.

##### 3.1.2 Quiz with bits of html

L02Q2.xml
<pre>&lt;!DOCTYPE quiz SYSTEM "L02Q2.dtd"&gt; &lt;quiz&gt;   &lt;qu&gt;&lt;i&gt;Alice&lt;/i&gt; sends secrets   to...&lt;/qu&gt;   &lt;d&gt;Ben&lt;/d&gt;   &lt;d&gt;Bill&lt;/d&gt;   &lt;d&gt;Bob&lt;/d&gt; &lt;/quiz&gt;</pre>

L02Q2.dtd
<pre>&lt;!ENTITY % xhtml SYSTEM   "xhtml/DTD/xhtml11-flat.dtd"&gt; %xhtml; &lt;!ELEMENT quiz (qu,d+)*&gt; &lt;!ELEMENT qu (#PCDATA %Inline.mix;)*&gt; &lt;!ELEMENT d (#PCDATA)&gt;</pre>

In this second example we allow inline tags (such as <i>) to be included in the qu section. The dtd pulls in all of the entity %xhtml; this entity is realised in the dtd – it creates many other entities – including the entity %Inline.mix

We everything that may be included in html inline tags is available in the qu node.

### 3.1.3 Quiz as a variation of html

In this example we choose to represent the quiz as an enhancement of html. We define a dtd that gives us html with a couple of extra new tags qu and d.

L02Q3.xml	L02Q3.dtd
<pre>&lt;!DOCTYPE html SYSTEM "L02Q3.dtd"&gt; &lt;html&gt;&lt;head&gt;&lt;title&gt;Quiz&lt;/title&gt;&lt;/head&gt; &lt;body&gt; &lt;p&gt;Complete the phrase:&lt;/p&gt; &lt;quiz&gt;   &lt;qu&gt;foo&lt;/qu&gt;   &lt;d&gt;ba&lt;/d&gt;   &lt;d&gt;fa&lt;/d&gt;   &lt;d&gt;na&lt;/d&gt; &lt;/quiz&gt; &lt;/body&gt;&lt;/html&gt;</pre>	<pre>&lt;!ENTITY % Block.extra " quiz"&gt; &lt;!ENTITY % xhtml SYSTEM   "xhtml/DTD/xhtml11-flat.dtd"&gt; %xhtml; &lt;!ELEMENT quiz (qu,d+)*&gt; &lt;!ELEMENT qu (#PCDATA)&gt; &lt;!ELEMENT d (#PCDATA)&gt;</pre>

Notice that we specify the entity %Block.extra; before realising the %xhtml; entity.

The %Block.extra; entity is also defined within the xhtml file – however the rule is that only the first definition of an entity is effective – the second definition is ignored.

The %Block.extra; entity shows up in the ELEMENT declarations...

```
<!ENTITY % Block.extra "" >
<!ENTITY % Block.class
  "%BlkStruct.class;
  %BlkPhras.class;
  %BlkPres.class;
  %BlkSpecial.class;
  %Block.extra;"
>
<!ENTITY % Block.mix
  "%Heading.class;
  | %List.class;
  | %Block.class;
  %Misc.class;"
>
<!ENTITY % body.qname "body">
<!ENTITY % body.content
  "( %Block.mix; )+"
>
<!ELEMENT %body.qname; %body.content; >
```

## 4. Some html

### 4.1 div and span

The div tag is used to mark a block of text. A block of text can be singled out for a number of purposes:

- it may be named - the id attribute should be used for this
- it may have a particular format applied
- it may be used in scripting – the content or format might be changed dynamically

The span tag is similar – but it operates in-line rather than as a block.

### 4.2 table

The table tag can be very useful in laying out pages in a flexible way. A common layout is to have a bar at the top of the page and a panel to the left and a panel to the right.

This is the bar at the top – it might have a mast head or logo in it – possibly global navigation elements		
The left side bar. Often used for context sensitive navigation features	Main body – this is where the actual content goes.  If we give 15% of the table width each to the side-bars then we have only 70% left for content.  The borders are shown in this example. These are often not required on the final page.	The right side bar. Possibly a complete waste of screen space.

The above could be achieved with the following html...

Additional work: review the excellent web monkey tutorial

<http://www.webmonkey.com/teachingtool/>

Student **must** complete the bold items.

Students must *avoid* the items in italics – they are deprecated.

Students may tackle the plain text examples.

<b>Paragraphs</b>	Teletype	Adding/Aligning Images
<b>Headlines</b>	<i>Blink</i>	<b>Ordered lists</b>
<b>Links</b>	Preformatted text	<b>Unordered lists</b>
<i>Mailtos</i>	<i>Background image</i>	<b>Definition lists</b>
<i>Comment tags</i>	<i>Background color</i>	Image borders
Bold/Italics	Blockquotes	Wrapping
<i>Font color</i>	Line breaks	Adding/Aligning Images
<i>Font size</i>	Teletype	

[http://www.webmonkey.com/reference/html\\_cheatsheet/](http://www.webmonkey.com/reference/html_cheatsheet/)

### Directed Reading

Find out what the IGNORE tag does in a DTD.

How is this used in modular xhtml to allow re-users of xhtml to disallow modules – such as the applet tags.

HTML design issues:

Consider each of the following html innovations:

Issue	For	Against
FRAMES		
IFRAMES		
<font face='times'> tag replaced with <span style='font-family:times'>		
<strong>, <em> to replace <b> and <i>		
<ul type='a'> replaced by <ul style='list-style- type:lower-alpha'>		