Webdesign

A quick introduction to XML, XHTML and CSS

XML

XML is a textual format design to be readable by humans and machines without knowing anything else but XML. An XML document is a Tree consisting of so-called Nodes. Each Node is of a certain Type, e.g. Element or Text. Elements may have child nodes. Typical child nodes of an Element include Attributes, further Elements and Text. The following is a sample XML Document:

```
<element>
      <empty-element attribute=”value” />
      TEXT
</element>
```

Since it’s a tree there may be only one top-level element. The top-level element is called the document element. Most XML Documents contain an XML-Declaration, which may be omitted. A typical XML-Declaration is this:

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

The values are default values which are assumed by default by an XML-Processor if the XML-Declaration is omitted. Such a special element like the XML-Declaration is called a Processing Instruction (which is another Type of Node). The most popular Processing Instruction is xml-stylesheet, which may look like this:

```
<?xml-stylesheet type=”text/css” href=”my.css”?>
```

A typical pitfall are empty elements. Since XML requires the document to be parse-able without additional information you explicitly have to tell, that the element does not have content and thus there will be no ending tag. This is done by inserting a slash at the end of the elements declaration:

```
<i-am-empty />
```

The following expressions are equivalent:

```
<i-am-empty />
<i-am-empty/>
<i-am-empty></i-am-empty>
```

XHTML

XHTML is “a reformulation of HTML in XML”. The document element of an XHTML document is typically `<html>` and consists of exactly two subordinate elements (we say: it has two child nodes, or: it has two children). These are `<head>` and `<body>` in exactly that order. The actual content of an XHTML document is to be found within the body, while the head contains meta-information like associated stylesheets, links to alternate versions or the title of the document. A typical head may look like this:

```
<head>
      <title>Hello, World!</title>
      <link rel=”stylesheet” type=”text/css” href=”my.css” />
</head>
```

This links the file “my.css” as a Cascading StyleSheet with the current document. The result will be, that the style-rules from my.css (located in the same directory as the document) will be applied when rendering the document.

The primary intention of XHTML is to apply semantics to text, not style (since that’s what Stylesheets are for). Since XHTML-Documents describe, well, documents they describe headings, paragraphs, tables. The following elements should give you a basic idea:

Element	Full Name
`<h1>, <h2>, <h3>, <h4>, <h5>, <h6>`	heading
`<p>`	paragraph
`<ul>, <ol>`	unordered list / ordered list
`<li>`	list item
`<table>, <tr>, <td>`	table, table row, table cell
`<dl>`	definition list
`<dt>, <dd>`	term, definition

These elements express a certain meaning, for example the following piece of code defines the terms “XML” and “XHTML”:

```
<dl>
      <dt>XML</dt>      <dd>eXtensible Markup Language</dd>
      <dt>XHTML</dt>    <dd>A reformulation of HTML in XML</dd>
</dl>
```

Most elements can be refined by defining certain attributes. While elements carry meaning, most attributes are somewhat functional (since you want to achieve some kind of special behavior).

Attribute	Meaning	Sample
`id`	A unique name within this document	`<h1 id=”firstHeading”>`
`class`	The element belongs to a certain set of things	`<p class=”quote”>`
`title`	An alternate rendering of the element (typically displayed as a little yellow box when hovering over the elements content)	`<img title=”My cat” />`
`lang`	The language of the elements content	`<p lang=”en-gb” />`
`xml:lang`	The XHTML-equivalent for HTMLs lang	`<p xml:lang=”en-gb” />`

By far the most interesting attributes for our purposes are `id` and `class`.

There are elements which rely on special attributes. These, and their attributes, are:

`<img src=”me.png” alt=”A picture of myself” />`	
`src`	Path to the source-file which is to be included as an image.
`alt`	Alternate text if the image can not be displayed.
`<a href="http://de.wikipedia.org/wiki/Rugby_Union">Rugby</a>`	
`href`	“hyper-reference”, the linked resource.
`rel`	The kind of relation between this and the target.
`<form method=”post”>`	
`method`	The HTTP-Method used for sending the form (get or post)
`action`	What to do when sending (typically a resource to send the data to)
`<input type=”text” />`	
`type`	text, password, checkbox, radio, submit or hidden.
`name`	The name for this field when submitting the data.

CSS

Cascading StyleSheets are used to define the visual layout of HTML and XML documents. The name was chosen since when you apply a style to an element that style cascades to the elements children (the children inherit the style of their parents). The syntax of CSS is easy and straight forward:

```
<selectors> {
      (<rule-name>: <rule-definition>;)*
}
```

For example:

```
body {
      background-color: black;
      color: white;
}
```

Selectors are a comma-separated list of tag-names, class-references, id-references or combinations of selectors. The following example matches all elements of the name “h4”, all elements having the name “a” which are contained in a table, all paragraphs which belong to the class “quote” and the element h1 with the id “firstHeading”.

```
h4, table a, p.quote, h1#firstHeading
```

Some basic rules are:

Rule	Takes	Sample
color	<color>	color: red; color: #ffa500; color: rgb(50%, 50%, 50%);
font-family	<font-name>	font-family: “Times New Roman”
font-style	italic or normal	font-style: italic;
font-weight	bold or normal	font-style: bold;
font-size	<size>	font-size: 10px; font-size: 12pt;
text-align	left, center, right, justify	text-align: justify;
text-transform	none, lowercase, uppercase, capitalize	text-transform: capitalize;

These rules affect the styling of the text (color is the color of the text).

There is a kind of rules which affect the layout of the document. Elements styled with CSS are laid out as blocks. Blocks may have background-colors, borders, margins, paddings and a custom position.

<table>
<tr><th>Rule</th><th>Takes</th><th>Sample</th></tr>
<tr><td>background-color</td><td><color></td><td>background-color: white;</td></tr>
<tr><td>display</td><td>inline, block, table, table-row, table-cell, list-item, none</td><td>div { display: block; }
span { display: block; }
head { display: none; }</td></tr>
<tr><td>width</td><td><size></td><td>table { width: 80%; }
input[type=text] { width: 200px; }</td></tr>
<tr><td>height</td><td><size></td><td rowspan="3">div {
  position: absolute;
  top: 100px;
  left: 200px;
  width: 300px;
  height: 150px;
}</td></tr>
<tr><td>position</td><td>absolute, relative, static, fixed</td></tr>
<tr><td>left, top, right, bottom</td><td><size></td></tr>
<tr><td>margin</td><td><size>{1,4}</td><td rowspan="3">p {
  margin: 10px;
  padding: 5px 10px;
  border: 1px solid green;
}</td></tr>
<tr><td>padding</td><td><size>{1,4}</td></tr>
<tr><td>border</td><td><size> <border-style> <color></td></tr>
</table>