WT UNIT - 1

Introduction to XHTML
origins and evolution of HTML and XHTML, basic syntax, standard XHTML document structure, basic text markup tags, images, hypertext links, lists, tables, forms, frames, syntactic differences between HTML and XHTML, introduction to HTML 5.

Cascading Style Sheets (CSS)
Introduction, levels of style sheets, style specification formats, selector forms, property value forms, font properties, list properties, color, alignment of text, box model, background images, positioning.

HTML- HyperText Markup Language

- HTML is defined with the use of the Standard Generalized Markup Language (SGML), language for defining markup languages
- Describes the general form and layout of documents
- An HTML document is a mix of content and controls
- Controls are tags and their attributes
- Tags often delimit content and specify something about how the content should be arranged in the document
- Attributes provide additional information about the content of a tag

XHTML- Extensible Hypertext Markup Language

- XHTML is a stricter, more XML-based version of HTML
- XML is a markup language where all documents must be marked up correctly (be "well-formed").
- <html>, <head>, <title>, and <body> are **mandatory**

CSS- Cascading Style Sheets

- A language for defining stylesheets that was developed for HTML
- Provide the means to control and change presentation of HTML documents
- Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents

XML

- A meta-markup language
- Used to create a new markup language for a particular purpose or area
- Because the tags are designed for a specific area, they can be meaningful
- No presentation details
- A simple and universal way of representing data of any textual kind

JavaScript

- A client-side HTML-embedded scripting language

- Only related to Java through syntax
- Dynamically typed and not object oriented
- Provides a way to access elements of HTML documents and dynamically change them

Origins and evolution of HTML and XHTML

- In late 1994 Tim Berness Lee stared W3C World wide web Consortium to develop and distribute standards for Web technologies starting with HTML
- Hypertext Markup Language
- Developed for the delivery of hypertext on the WWW
- Built using SGML
- ASCII “Markup Language”

HTML Versions

- First HTML standard was HTML 2.0 was released in 1995
- HTML 3.2 was released in Jan 1997.
- HTML 4.0 – Dec1997

  Introduced many new features and deprecated many older features
- HTML 4.01 - 1999 - A cleanup of 4.0
- XHTML 1.0 - 2000

  Just 4.01 redefined using XML, instead of SGML
- XHTML 1.1 – 2001

  Modularized 1.0, and drops frames
- XHTML 2.0 – 2010 development abandoned
- HTML 5.0 2008

  Working Draft (not a W3C Recommendation yet)
- HTML and XHTML syntax

  October 2018 W3C Recommendation

HTML VS XHTML

- HTML has lax syntax rules, leading to sloppy and sometime ambiguous documents
- XHTML syntax is much more strict, leading to clean and clear documents in a standard form
- Even minor syntax errors will prevent a document labelled as XML from being rendered fully,whereas they would be ignored in the HTML syntax
- HTML compatible with most legacy Web browsers

Basic syntax

- Elements are defined by tags (markers)
- Tag format:
  - Opening tag: <name>
  - Closing tag: </name>
- The opening tag and its closing tag together specify a container for the content they enclose
  - E.g. <p> Hello </p>
- Not all tags have content
  - E.g. <hr>
- The container and its content together are called an element
- If a tag has attributes, they appear between its name and the right bracket of the opening tag
- E.g. <img src = “c10.jpg” />
- Comment form: <!-- … -->
- Browsers ignore comments, unrecognizable tags, line breaks, multiple spaces, and tabs

- Tags are just suggestions to the browser, even if they are recognized by the browser

XHTML Document structure

- An XHTML document is composed of 3 parts:

1.a line containing HTML version information , encoding and document type
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

2.html tag and declarative header section

- ○ <html xmlns = "http://www.w3.org/1999/xhtml"› xmlns is the XHTML name space
- ○ <head> tag consists o <title>tag
- ○ The <title>tag is used to give the document a title
- ○ normally displayed in the browser's window title bar

3.a body containing the document's actual content

- ○ Delimited with the <body> tag
- ○ After document type declaration, the remainder of an XHTML document is contained by the xhtml element

Basic text markup

Paragraph Elements: <p>
Text is normally placed in paragraph elements
The browser puts as many words of the paragraph's content as will fit in each line
<p>
Greetings!
</p>

Line breaks: <br/>
Horizontal rules: <hr/>
Preserving white space <pre>

Headings
Six sizes, 1 - 6, specified with <h1> to <h6> 1, 2, and 3 use font sizes that are larger than the default font size
4 uses the default size
5 and 6 use smaller font sizes

Blockquotes: <blockquote>
To set a block of text off from the normal flow and
appearance of text
Browsers often indent, and sometimes italicize

Font Styles and Sizes (can be nested)
Boldface: <b>
Italics: <i>
Smaller: <small>
Larger: <big>(not supported in HTML5)
Monospace: <code> (not supported in HTML5)

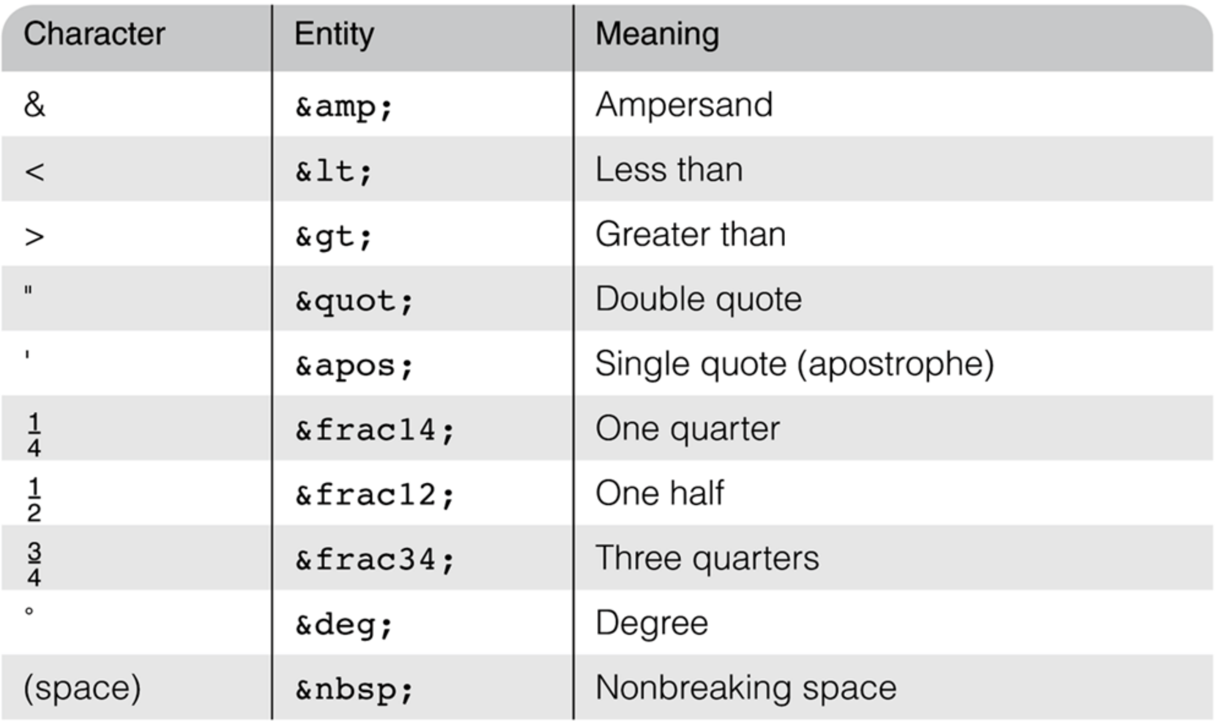
Example: The <big>sleet <big> in <big> <i> Crete </i><br /> lies </big> completely
</big> in </big> the street

Display: The sleet in Crete lies completely in the street

<marquee> It is used to scroll the text </marquee>
Used for scrolling text

Subscripts: <sub> Superscripts: <sup> Example: x<sub>2</sub><sup>3</sup> Display: x2

Character Entities



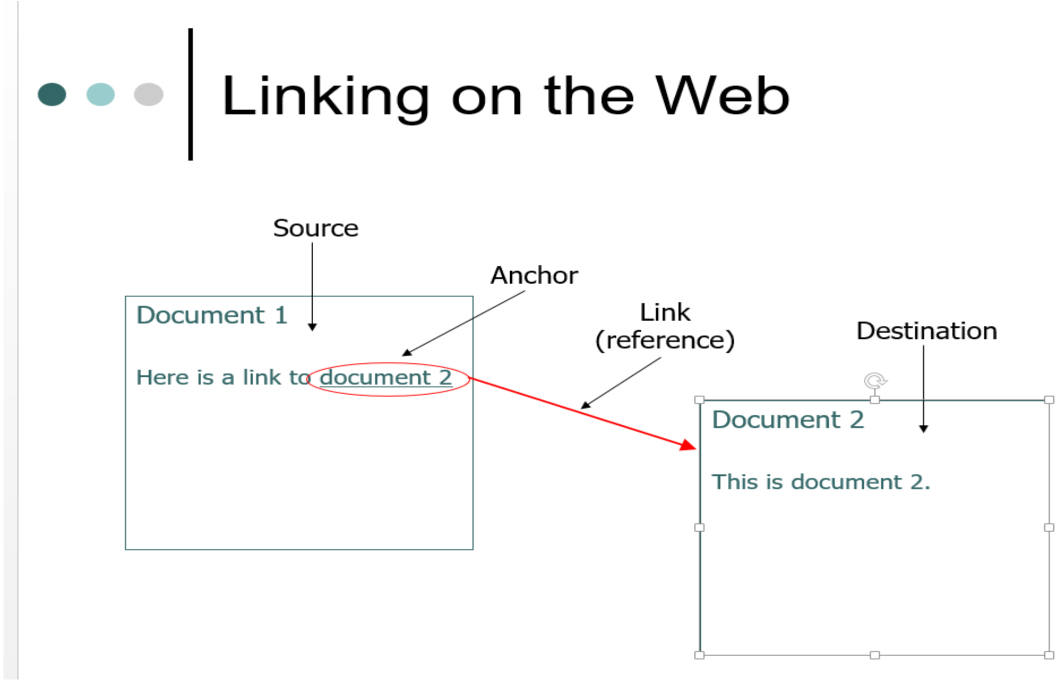
Character	Entity	Meaning
&	&amp;	Ampersand
<	&lt;	Less than
>	&gt;	Greater than
"	&quot;	Double quote
'	&apos;	Single quote (apostrophe)
¼	&frac14;	One quarter
½	&frac12;	One half
¾	&frac34;	Three quarters
°	&deg;	Degree
(space)	&nbsp;	Nonbreaking space

Meta tag:

- Meta element is used to provide additional information about a document.
- The meta tag has no content; rather, all of the information provided is specified with attributes.
- The two attributes that are used to provide information are name and content.
- <meta name = "keywords" content = "binary trees,linked lists, stacks" />

Images

- All modern web browsers can display images inline (i.e. embedded in the text)
- GIF (Graphic Interchange Format)
    - 8-bit color (256 different colors)
- JPEG (Joint Photographic Experts Group)
    - 24-bit colour (16 million different colours)
- Portable Network Graphics (PNG)
    - Relatively new
    - Designed for transferring images on the Internet
- Images are inserted into a document with the <img> tag with the src attribute
- The alt attribute describes about the image
    - Non-graphical browsers
    - Browsers with images turned off
- <img src = “logo.jpg" alt = “University of Nottingham Logo" />
- The <img> tag has other optional attributes, including width and height (in pixels)



Linking on the Web

Hypertext Links

- Hypertext is the essence of the Web!
- A link is specified with the href (hypertext reference) attribute of <a> (the anchor tag)
- The content of <a> is the visual link in the document
- <a href="target.html">This is a link</a>
- Relative addressing of targets is easier to maintain and more portable than absolute addressing

Targets within documents

- If the target is not at the beginning of the document, the target spot must be marked
- Target labels can be defined in many different tags with the id attribute
- <h1 id = "baskets"> Baskets </h1>
- The link to an id must be preceded by a pound sign (#)
  - target is in the same document,
- <a href = "#baskets"> Baskets </a>
  - target is in a different document
- <a href = "myAd.html#baskets"> Baskets </a>

Image Hyperlinks

- Links can include images in their content
- <a href = "c210data.html">
- <img src = "smallplane.jpg" alt = "Small picture of an airplane ">Info on C210 </a>

Unordered lists

- The list is the content of the <ul> tag
- List elements are the content of the <li> tag

```
<h3> Some Common Single-Engine Aircraft </h3>
<ul>
        <li>    Cessna        Skyhawk      </li>
        <li>    Beechcraft    Bonanza      </li>
        <li>    Piper  Cherokee     </li>
</ul>
```

Ordered lists:

- The list is the content of the <ol> tag
- Each item in the display is preceded by a sequence value

```
<h3> Cessna 210 Engine Starting Instructions </h3>
<ol>
        <li> Set mixture to rich </li>
        <li> Set propeller to high RPM </li>
        <li> Set ignition switch to "BOTH" </li>
        <li> Set auxiliary fuel pump switch to "LOW PRIME" </li>
        <li> When fuel pressure reaches 2 to 2.5
        PSI, push starter button </li>
</ol>
```

Nested lists:

- Any type list can be nested inside any type list
- The nested list must be in a list item

```
<ol>
        <li>    Single-Engine Aircraft
        <ol>
                <li>    Tail    wheel  </li>
                <li>    Tricycle       </li>
        </ol>   <br>
        </li>
        <li>    Dual-Engine  Aircraft
        <ol>
                <li>    Wing-mounted        engines        </li>
                <li>    Push-pull     fuselage-mounted   engines        </li>
        </ol>
        </li>
</ol>
```

Definition lists(or glossaries):

- List is the content of the <dl> tag
- Terms being defined are the content of the <dt> tag
- The definitions themselves are the content of the

<dd> tag

```
<dl>
        <dt>   152    </dt>
        <dd>   Two-place     trainer </dd>
        <dt>   172    </dt>
        <dd>   Smaller four-place   airplane </dd>
</dl>
```

Syntactic Differences between HTML & XHTML

- Case sensitivity
- Closing tags
- Quoted attribute values
- Explicit attribute values
- id and name attributes

- Element nesting

Tables

- A table is a matrix of cells
    - The cells can include almost any element
    - Some cells have row or column labels and some have data
- A table is specified as the content of a <table> tag
- Each row of a table is specified as the content of a <tr> tag
- The row headings are specified as the content of a <th> tag
- Each data cell of a row is specified as the content of a <td> tag

Table Basic Formatting

- A border attribute in the <table> tag specifies a border between the cells • If border is set to "border", the browser's default width border is used
    - The border attribute can be set to a number, which will be the border width
    - E.g. border = "4"
    - Without the border attribute, the table will have no lines!
- Tables are given titles with the <caption> tag, which can immediately follow <table>

rowspan and colspan attributes

- Table cells are sized to fit the data they contain
- You can create larger data cells by using these attributes:
    - rowspan – set in the <th> or <td> tag to specify that the cell must span some number of rows
    - colspan – set in the <th> or <td> tag to specify that the cell must span some number of columns

align and valign attributes

- The align attribute controls the horizontal placement of the contents in a table cell
- Values are left, right, and center
- Default for labels is center, for data it is left
    - align is an attribute of <tr>, <th>, and <td>
- The valign attribute controls the vertical placement of the contents of a table cell • Values are top, bottom, and center (default)
    - valign is an attribute of <th> and <td>

cellpadding and cellspacing attributes

- The cellspacing attribute of <table> is used to specify the distance between cells in a table The cellpadding attribute of <table> is used to specify the spacing between the content of a cell and the inner walls of the cell

Table Sections

- Tables naturally occur in two and sometimes three parts
- Header – <thead>
    - First section, contains header information such as column names
- Body – <tbody>
    - Table's primary data
    - If a document has multiple tbody elements, they are separated by thicker horizontal lines
- Footer – <tfoot>
    - E.g. for calculation results and footnotes

Forms

- A form is the usual way information is communicated from a browser to a server
- HTML has tags to create a collection of objects that implement this information gathering
- These objects are called controls or widgets
- (e.g., radio buttons and checkboxes)
- When the Submit button of a form is clicked, the form’s values are sent to the server

The <form> Tag

- All of the components (widgets) of a form are defined in the content of a <form> tag
- The only required attribute of <form> is action
  - specifies the URL of the application that is to be called when the Submit button is clicked
  - action = "http://www.cs.nott.ac.uk/cgibin/survey.pl"
  - if the form has no action, the value is set to the empty string (“ ”)
- The method attribute of <form> specifies one of the two possible techniques of transferring the form data to the server, get and post

get and post

- get or post, used to pass the form data to the server. default is get
- Browser attaches the query string to the URL of the HTTP request, so the form data is transmitted to the server together with the URL
- The browser inserts a question mark at the end of the actual URL just before the first character of the query string so that the server can easily find the beginning of the query string.
- The get method can also be used to pass parameters to the server when forms are not involved.
- One disadvantage of the get method is that some servers place a limit on the length of the URL string and truncate any characters past the limit.
- So, if the form has more than a few controls, get is not a good choice
- post method is used, the query string is passed by some other method to the form-processing program.
- There is no length limitation for the query string with the post method, better choice when there are more than a few controls in the form.
- There are also some security concerns with get that are not a problem with post

The <input> Tag

- Many controls are created with the <input> tag The type attribute of <input> specifies the kind of control being created
- Text
  - Creates a horizontal box for text input
  - Default size is 20 characters; it can be changed with the size attribute
  - If more characters are entered than will fit, the box is scrolled (shifted) left
  - maxlength – max number of characters
- <input type = "text" name = "Phone" size = "12" >

Password

- type = “password”
- Only bullets or asterisks displayed by browser

Checkboxes

- to collect multiple choice input
- Every checkbox requires a value attribute, which is the widget's value in the form data when the checkbox is 'checked'
- A checkbox that is not 'checked' contributes no value to the form data
- By default, no checkbox is initially 'checked' • To initialize a checkbox to 'checked', the checked attribute must be set to "checked"

Radio Buttons

- collections of checkboxes in which only one button can be 'checked' at a time
- Every button in a radio button group MUST have the same name
- If no button in a radio button group is selected the browser often selects the first one

The <select> Tag - Menus

- There are two kinds of menus
  - those that behave like checkboxes and
  - those that behave like radio buttons (the default)
- Menus that behave like checkboxes are specified by including the multiple attribute, which must be set to "multiple"
- The name attribute of <select> is required
- The size attribute of <select> can be included to specify the number of menu items to be displayed (the default is 1)
- If size is set to > 1 or if multiple is specified, the menu is displayed as a scrolled list
- Each item of a menu is specified with an <option> tag
- whose pure text content (no tags) is the value of the item
- An <option> tag can include the selected attribute, which when assigned "selected" specifies that the item is pre-selected

The <textarea> Tag

- Creates a multiline text area
- Usually include the rows and cols attributes to specify the size of the text area
- Default text can be included as the content of <textarea>
- Scrolling is implicit if the area is overfilled

Submit and Reset Buttons

- Both are created with <input>
- <input type = "reset" value = "Reset Form">
- <input type = "submit" value = "Submit Form">
- Reset clears all controls to their initial states
- Submit has two actions:
  - Encode the data of the form
  - Request that the server execute the serverresident program specified as the value of the action attribute of <form>
- A Submit button is required in every form

Frames

- Rectangular sections of the display window, each of which can display a different document
- Uses of frames
  - Navigation systems (table of contents)
  - Static title

- Advertising

- Frames are not supported in XHTM5 Use discouraged because they:
  - break the fundamental paradigm of the web
  - The page is the basic unit of the web represented by a URL
  - cause usability problems

Framesets

- The <frameset> tag specifies the number of frames and their layout in the window
- <frameset> takes the place of <body>
- Cannot have both!
- <frameset> must have either a rows attribute or a cols attribute, or both (usually the case)
  - The possible values for rows and cols are:
  - numbers – size in pixels
  - percentages – % of total browser window size
  - asterisks – remainder of window

Frame tag

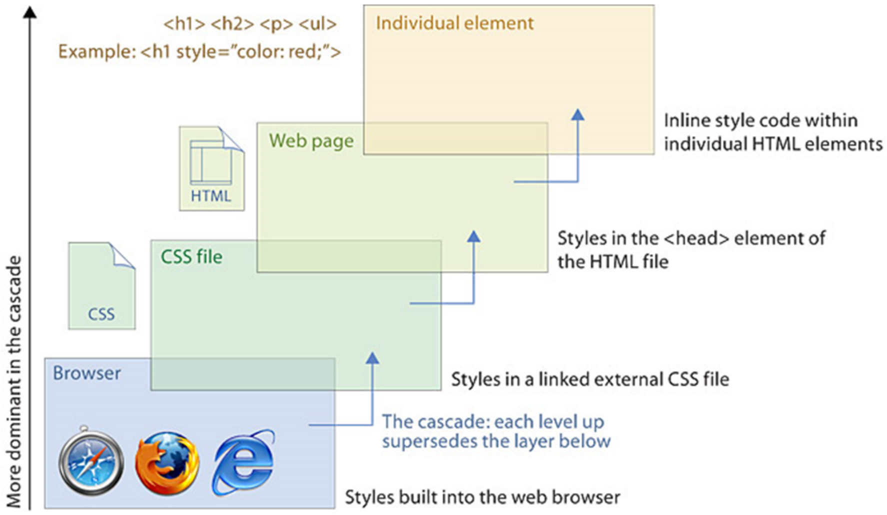
- The <frame> tag specifies the content of a frame
- The first <frame> tag in a <frameset> specifies the content of the first frame, etc.
  - Row-major order is used
  - Frame content is specified with the src attribute
  - Without a src attribute, the frame will be empty
- (such a frame CANNOT be filled later) If <frameset> has fewer <frame> tags than frames, the extra frames are empty

Cascading Style Sheets (CSS)

CSS

- CSS provides the means to control and change presentation of HTML documents
- Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents
- CSS1 specification – 1996 CSS2 specification - 1998
- CSS level 2 revision 1 (“CSS 2.1”) April 12, 2016
- CSS3 7 June 2011
- CSS4 March 24. 2017
-
- Levels of Style Sheets
- There are three levels of style sheets
  - Inline - specified for a specific occurrence of a tag and apply only to that tag
  - This is fine-grain style, which defeats the purpose of style sheets - uniform style
  - Document-level style sheets - apply to the whole document in which they appear
  - External style sheets - can be applied to any number of documents
- When more than one style sheet applies to a specific tag in a document, the lowest level style sheet has precedence

CSS cascade hierarchy



<h1> <h2> <p> <ul>
Example: <h1 style="color: red;">
Individual element
Inline style code within individual HTML elements
HTML
Web page
Styles in the <head> element of the HTML file
CSS
CSS file
Styles in a linked external CSS file
Browser
The cascade: each level up supersedes the layer below
Styles built into the web browser
More dominant in the cascade

Levels of Style Sheets

- Inline style sheets appear in the tag itself
- Document-level style sheets appear in the head of the document
- External style sheets are in separate files, potentially on any server on the Internet
    - Written as text files with the MIME type text/css
    - A <link> tag is used to specify that the browser is to fetch and use an external style sheet file
- <link rel = "stylesheet" type = "text/css" href = "http://www.wherever.org/example.css">

Inline Style Specification

- Style sheet appears as the value of the style attribute General form:

  style = "property_1: value_1;
  Property_2: value_2;
  … property_n: value_n;"

Document Style Specification

- Style sheet appears as a list of rules that are the content of a <style> tag
- The <style> tag must include the type attribute, set to "text/css"
    - <style type = "text/css"> rule list </style> Form of the rules:
    - selector {property_1:value_1;
    - property_2:value_2; …; property_n:value_n;}

External Style Sheet Specification

- An external style sheet is used to define the style for many HTML pages.
- To use an external style sheet, add a link to it in the <head> section of each HTML page using <link>tag
- The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.
-
- @import url(filename) can be used instead of
- <link> tag inside <style> tag and I should be the first statement in it Ex: import.xhtml

Selector Forms

1. Simple Selector
The selector is a tag name or a list of tag names, separated by commas Examples:
h1 {font_size: 24pt;} h2, h3 {font_size: 20pt;}

2. Contextual selectors
Apply style only to elements in specified position in body of document body b i {font_size: 30pt} apply only for bold and italics text

3 .Class Selector
Used to allow different occurrences of the same tag to have different style specifications
A style class has a name, which is attached to a tag name

```
    p.normal {property-value list}
    p.warning {property-value list}
```

The class you want on a particular occurrence of a tag is specified with the class attribute of the tag

```
    <p class = "normal"> ... </p>
    ...
    <p class = "warning"> ... </p>
```

4. Generic Selectors
A generic class can be defined if you want a style to apply to more than one kind of tag
A generic class must be named, and the name must begin with a period
.really-big { … }
Use in body of doc like normal style class

```
    <h1 class = "really-big"> … </h1> ...
    <p class = "really-big"> … </p>
```

5. id Selectors
An id selector allow the application of a style to one specific element General form:
#specific-id {property-value list}

```
    e.g. #section3 {font-size: 20} In XHMTL doc:
    <h2 id = “section3”>
    3. Properties for sale
    </h2>
```

6. Universal Selectors
Universal Selectors are denoted by an asterisk(*) applies its style to all elements in the document
Type the code in css file an use it in the xthml file
Universal Selectors not often used

7. Pseudo Classes
Pseudo classes are styles that apply when something happens, rather than because the target element simply exists
Names begin with colons
hover class applies when the mouse cursor is over the element
focus class applies when an element has focus

Properties

- There are different properties in 12 categories:
  - Background
  - Border and outline
  - Dimension
  - Font
  - Generated content
  - List Margin
  - Padding
  - Positioning
  - Print
  - Table
  - Text

Property Value Forms

- Keywords - left, small, …
  - Length - numbers, maybe with decimal points
- Units:
  - px – pixels
  - in – inches
  - cm – centimeters
  - mm – millimeters
  - pt – points
  - pc - picas (12 points)
  - em - height of the letter 'm' (relative length)
  - ex-height - height of the letter 'x' (relative length)
- No space is allowed between the number and the unit specification, e.g. 1.5 in is illegal!

Property Value Forms

- Percentage - just a number followed immediately by a percent sign
- URL values
- url(protocol://server/pathname)

Colors

- Color name, e.g. white
- Hex form: #XXXXXX, e.g. #FFFFFF
- rgb(n1, n2, n3), e.g. rgb(255, 255, 255) • Numbers can be decimal (0-255) or percentages
- Property values are inherited by all nested tags, unless overridden

Font Properties

- font-family
  - Value is a list of font names - browser uses the first in the list it has
- font-size
  - Possible values: a length number or a name, such as smaller, xx-large, etc.
- font-style
  - italic, oblique (useless), normal
- font-weight
  - degrees of boldness bolder, lighter, bold, normal

- font - for specifying a list of font properties
    - font: bolder 14pt Arial Helvetica
    - Order must be: style, weight, size, font name(s)

Text Decoration

- The text decoration property is used to specify some special features of text.
- The available values are line-through, overline, underline

List Properties

<u>list-style-type</u>

On unordered lists list-style-type can be used to specify the shape of the bullets

disc (default), square, or circle

Set it on either the <ul> or <li> tag

```
<h3> Fruit </h3>
 <ul>
        <li style = "list-style-type: disc"> Apple </li>
        <li style = "list-style-type: square"> Orange </li>
        <li style = "list-style-type: circle"> Pear </li>
</ul>
```

- list-style-image can be use to make images as shape
- On ordered lists list-style-type can be used to change the sequence values I

Text and Background Colour

- The color property specifies colour of text
- <style type = “text/css”> th.red {color: red} th.orange {color: orange}
- </style>
- The background-color property specifies the background colour of elements

Background Images

- The background-image property
- background-repeat property
    - Possible values: repeat (default), norepeat, repeat-x, or repeat-y
- background-position property
    - Possible values: top, center, bottom, left, or right

Text Alignment

- The text-indent property allows indentation
    - Takes either a length or a % value
- The text-align property has the possible values, left (the default), center, right, or justify   Ex: textalign.xhtml
- Sometimes we want text to flow around another element - the float property
    - values of left, right, and none (the default)

The <span> and <div> tags

- One problem with the font properties is that they apply to whole elements, which are often too large
- Solution: a new tag to define an element in the content of a larger element – <span>
- The Purpose of <span> is to change properly values of part of a line of content
- <style type = "text/css">

- .bigred {font-size: 24pt; font-family: Ariel; color: red}
- </style>

…

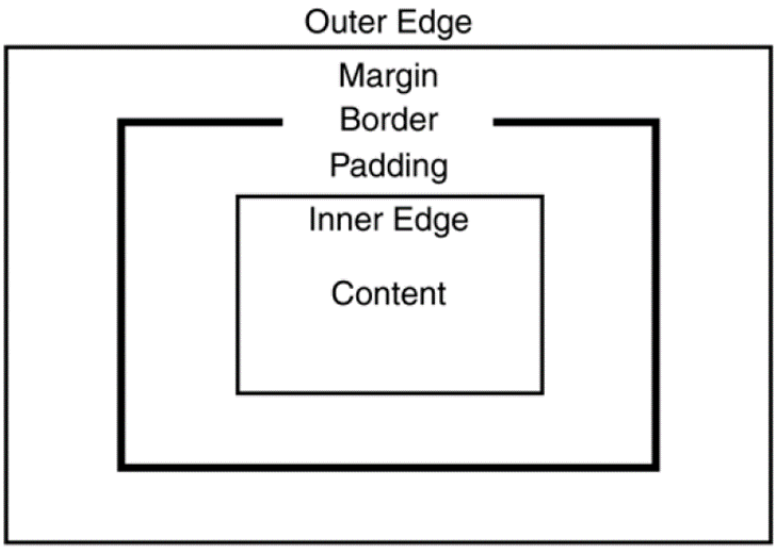
<p> Now is the <span class = "bigred"> best time </span> ever! </p>

The <span> and <div> tags

- The <span> tag is similar to other HTML tags, they can be nested and they have id and class attributes
- Another tag that is useful for style specifications:
- **<div>**
- Used to create document sections (or divisions) for which style can be specified
- e.g., a section of five paragraphs for which you want some particular style

The Box Model

- All XHTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
- The CSS box model is essentially a box that wraps around every XHTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



Borders

- The Border properties are border-style, border-width, border-color
- Every element has a border-style property.
- It Controls whether the element has a border and if so, the styles of the borders are dotted, dashed and double.
- The styles of one of the four sides of an element can be set with border-top-style, border-bottom-style, borderleft-style , border-right-style
- border-width can be used to specify thickness of a border with possible values of thin, thick, medium or length value in pixels
- Width of the four borders are border-top-width,borderbottom-width,border-left-width,border-right-width
- Color of border controlled by border-color it can be specified for any of the four borders border-top-color,borderbottom-color,border-left-color,border-right-color Ex: boderspro.xhtml

Margins and Padding

- Margin is the space between border of an element and elements neighbor
- Padding is the space between content of an element and its border
- Margin properties are named as margin which are applied to all the four sides of an element margin-left,
- margin-right, margin-top, margin-bottom
- Padding properties are named as padding which are applied to all the four sides of an element padding–left, padding–right, padding–top, padding-bottom Ex: marpadpro.xhtml

Conflict Resolution

- When two or more rules apply to the same tag there are resolutions for deciding which rule applies.
- In-line style sheets have precedence over document style sheets. Document style sheets have precedence over external style sheets.
- Within the same level there can be conflicts a tag may be used twice as a selector
- h3{color:red;} body h3 {color: green;}
- A tag may inherit a property and also be used as a selector.
- Style sheets can have different sources
- The browser itself may set some style
- eg: In FX2 min font size can be set in Tools-Options- Advanced The author of a document may specify styles.
- The user, through browser settings, may specify styles.
- Individual properties can be specified as important or normal.
- Eg: p.special{font-style: italic !important; font-size :14}
- This property is known as weight of a specification
- Conflict resolution is a multistage sorting process.
- The first step in the process is to gather the style specifications from the three possible levels of style sheets.
- These specifications are sorted into order by the relative precedence of the style sheet levels.
- This is done according to the following rules, in which the first has the highest precedence.
- from highest to lowest
    - 1. Important declarations with user origin
    - 2. Important declarations with author origin
    - 3. Normal declarations with author origin
    - 4. Normal declarations with user origin
    - 5. Any declarations with browser (or other user agent) origin Tie-Breakers
- Conflict resolution by Specificity (high to low)
    - 1. id selectors
    - 2. Class and pseudo-class selectors
    - 3. Contextual selectors
    - 4. General selectors
- Position
- Essentially, later has precedence over earlier.
- Most recently seen specification is the one which gets more precedence.
- Sorting process to resolve the style specification is known as cascade.

HTML5 new features

- The video and audio elements for media playback
- The canvas element for drawing New content specific elements, e.g:
    - article, footer, header, nav, section New form controls, e.g.:
    - calendar, date, time, email, url, search Better support for local offline storage

Browser Support

- HTML5 is not yet an official standard, and no browser has full HTML5 support
- Canvas support - Firefox 3.0+, Safari 3.0+, Chrome 3.0+, Opera 10.0+
- HTML5 Storage support – IE8+, Firefox 3.5+, Safari 4.0+, Chrome 4.0+, Opera 10.5+
- Upcoming Internet Explorer 9 will be supporting most HTML5 features