

## Lecture Note 3 Client-Side Development I

By A/Prof Heng Tao SHEN  
School of Information Technology and Electrical Engineering  
The University Of Queensland

### Outline

- How to write a web page
  - HTML (this lecture)
  - CSS (this lecture)
  - JavaScript (next lecture)
- How to publish what you have done (Next Lecture)
  - Finding a host vs. Hosting by yourself
  - Applying a domain name
- Advance issues (Next Lecture)
  - Cross browser compatibility
  - Different browser versions of the same family

P. 2

### HTML (1/3)

- What is HTML?
  - Hypertext Markup Language
  - Provides a means to describe the structure of text-based information in a document
    - By denoting certain text as links, headings, paragraphs, lists, ...
- Note:
  - Markup Language is different from Programming Language
    - Programming Language (e.g. C):
      - Tells the computer what to do.
        - » `printf("Some information");`
    - Markup Language (e.g. HTML):
      - Tells the computer what is the meaning of a piece of text by using "tag"
        - » `<p>Some Texts</p>`

P. 3

### HTML (2/3)

- Why you need to learn HTML?
  - You may argue
    - We can use those What You See Is What You Get (WYSIWYG) editors (e.g. MS Expression Web, Dreamweaver, etc) to write the web page!
      - I also highly recommend you to use these software for development...
    - Why we have to learn it?
  - Well... Several reasons:
    - Machine's interpretation may not be what you have in your mind
      - Have you edited HTML directly? Why most editors allow you to do that?
    - Machine generated code is not as good as human generated one
      - And you often want to make explicit choices about document format, vendor-independence, standard-compliance...
    - Understanding of HTML is necessary for advanced users
      - Making some interactive elements (E.g. JSP, PHP, ASP, ...)
  - ...

P. 4

### HTML (3/3)

- The standard of HTML:
  - HTML 2.0 (1995)
  - HTML 4.0 (1997)
  - **HTML 4.01** (1999)
  - XHTML 1.0 (2000, a reformulation of HTML 4.01 using XML 1.0)
  - XHTML 1.1 (2001, reformulated from Modularization of XHTML)
  - XHTML 2 (no such standard, expired in 2009)
  - HTML 5 (working draft in 2008, would be the sole next-generation HTML standard, including both XML and non-XML serializations)
  - XHTML 5 (an update to XHTML 1.x, defined alongside HTML5)
  - W3C suggests that most authors use the HTML syntax, rather than the XHTML syntax
- The focus of this lecture:
  - We will not discuss every element in details in this course
  - We will focus on the concept of writing an HTML document

P. 5

### Tag

- About Tag:
  - It is a good habit to close all tags:
    - Good: `<p>A paragraph</p><p>Another paragraph</p>`
    - Bad (but valid): `<p>A paragraph<p>Another paragraph`
  - A tag can contains attribute:
    - `<a href="http://www.google.com">Google</a>`
  - If nothing appears within a tag, then this is also possible:
    - Good and valid: `<meta content="en-us"></meta>`
    - Good and valid: `<meta content="en-us"/>`
  - Tags are **NOT** case sensitive
    - All these have the same effect:
      - `<body></body>`, `<BODY></BODY>`, `<Body></Body>`, ...

P. 6

## Using WYSIWYG Editor

- Since ITEE has Microsoft Expression Web 3.0, we will use it as our default WYSIWYG editor in this course.
  - Used to be FrontPage 2003
  - Others (like Dreamweaver) are very similar.

P.7

## Basic Structure of a HTML Document (1/2)

- A HTML document should be look like this:
  - `<html>`  
`<head></head>`  
`<body></body>`  
`</html>`
  - Head (`<head></head>`)
    - Contains information about the document (will not be rendered)
      - Where to find files? How the document to be rendered? What is the language and character set? How to be found by search engines? ...
  - Body (`<body></body>`)
    - Contains information that you want to display
    - Tells the computer how to render the information

P.8

## Basic Structure of a HTML Document (2/2)

- We can have nested structure within any tag.
  - An example:
    - ```
<html>
<head>
<title>INFS3202</title>
</head>
<body>
<p>Welcome to the course of INFS3202.</p>
<p>I hope you enjoy this course.</p>
Please ask question if you have any problem. =>
</body>
</html>
```
  - Try it out by yourself!

P.9

## Head (1/2)

- Six major tags within Head:
  - Title (`<title>Some data</title>`)
    - The title of your page. Display in the "browser's top border".
      - `<title>INFS3202</title>`
      - `<title>The University of Queensland</title>`
  - Meta (`<meta>`)
    - Some important information to let the browser knows.
      - `<meta http-equiv="Content-Type" content="text/html; charset=big5" />`
        - » Set the character encoding to Traditional Chinese (Big5)
      - `<meta http-equiv="Content-Type" content="text/html; charset=shift_jis" />`
        - » Set the character encoding to Japanese (shift\_jis)
    - Content types are case-insensitive.
    - Examples of content types include "text/html", "image/png", "image/gif", "video/mpeg", "text/css", and "audio/basic"...

P.10

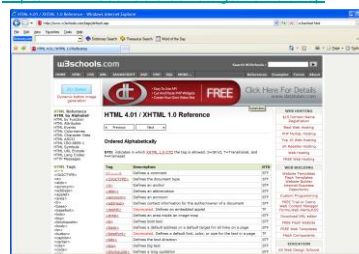
## Head (2/2)

- Style (`<style>Some rendering styles</style>`)
  - Define some rendering styles. We will discuss it shortly in CSS
- Link (`<link>`)
  - Link to an external file. We will discuss it shortly in CSS
- Script (`<script>Some Programming Scripts</script>`)
  - Within the script tag is some programming language (e.g. JavaScripts).
  - We try to ask the computer to "run some programs". We will discuss it in the [next lecture](#)
- Base (`<base href="an URL">`)
  - Base URL gives the base URL for dereferencing relative URLs
    - E.g., `<base href="http://www.google.com/">` `<IMG SRC="Icons/logo.gif" ALT="Some logo">`
    - The full URL for the image is
      - » `http://www.google.com/Icons/logo.gif`
  - Otherwise, the document URL is used

P.11

## Body (1/2)

- Too many different tags can be used in the body!!!
  - <http://www.w3schools.com/tags/default.asp>



- We only discuss some useful/tricky ones.

P.12

## Body (2/2)

- The 9 major elements that will be covered in this lecture:
  - Paragraph (<p>)
  - Line break (<br>)
  - Font (<font>)
  - Anchor (<a>)
  - Image(<img>)
  - Table (<table>)
  - Layer (<div>)
  - Span (<span>)
  - List (<ul> or <ol>)
- After learning (and of course familiar with) the above elements (together with css), you can properly build a very professional commercial web site
  - Although without interaction component yet!

P. 13

## Paragraph

- Write a paragraph! <p>
  - There is an obvious space between two paragraphs.
- Attribute commonly used:
  - Align
    - justify / left / right / center

```

<p align="justify">This course teaches the concepts and skills for building sophisticated database-driven Web Information Systems (WIS). It provides a comprehensive and systematic introduction to the latest WIS technologies. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of WIS, client-side and server-side technologies and web services.</p>
<p align="right">The primary aim of this course is to introduce, and cover part of in depth, aspects of Web-based information systems. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of Web Information Systems (WIS), client- and server-side technologies, and web services.</p>
<p align="center">Feedback is essential to effective learning and students can expect to receive appropriate and timely feedback on all assessment. For a detailed explanation of the feedback you are entitled to, you should consult the policy on Student Access to Feedback on Assessment.</p>

```

This course teaches the concepts and skills for building sophisticated database-driven Web Information Systems (WIS). It provides a comprehensive and systematic introduction to the latest WIS technologies. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of WIS, client-side and server-side technologies and web services.

The primary aim of this course is to introduce, and cover part of in depth, aspects of Web-based information systems. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of Web Information Systems (WIS), client- and server-side technologies, and web services.

Feedback is essential to effective learning and students can expect to receive appropriate and timely feedback on all assessment. For a detailed explanation of the feedback you are entitled to, you should consult the policy on Student Access to Feedback on Assessment.

P. 14

## Line Break

- Break a line into two (<br />)

```

<p align="justify">This course teaches the concepts and skills for building sophisticated database-driven Web Information Systems (WIS). It provides a comprehensive and systematic introduction to the latest WIS technologies. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of WIS, client-side and server-side technologies and web services.</p>
<br />
<p align="justify">This course teaches the concepts and skills for building sophisticated database-driven Web Information Systems (WIS). It provides a comprehensive and systematic introduction to the latest WIS technologies. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of WIS, client-side and server-side technologies and web services.</p>

```

This course teaches the concepts and skills for building sophisticated database-driven Web Information Systems (WIS). It provides a comprehensive and systematic introduction to the latest WIS technologies. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of WIS, client-side and server-side technologies and web services.

This course teaches the concepts and skills for building sophisticated database-driven Web Information Systems (WIS). It provides a comprehensive and systematic introduction to the latest WIS technologies. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of WIS, client-side and server-side technologies and web services.

P. 15

## Font (1/2)

- Set the style of the fonts (<font>)
- Always use the GUI to help you to do this instead of typing the code yourself, unless you cannot do so.

```

<p align="justify"><font size="1">This</font> course teaches the concepts and skills for building </font> sophisticated database-driven Web Information Systems (<font size="1">WIS</font>). It provides a comprehensive and systematic introduction to the latest <font color="#FFFFFF">WIS</font> technologies. </font>It has a focus on the latest technologies for WIS development. It <font face="Bell MT">covers</font> concepts <font face="Arial">fundamentals</font> of WIS, <font color="#FFFFFF">client-side</font> and server-side technologies and web services.</p>

```

This course teaches the concepts and skills for building sophisticated database-driven Web Information Systems (WIS). It provides a comprehensive and systematic introduction to the latest WIS technologies. It has a focus on the latest technologies for WIS development. It covers concepts fundamentals of WIS, client-side and server-side technologies and web services.

P. 16

## Font (2/2)

- Certainly, you can specify multiple attributes for a same tag
  - <font face="Arial" color="#FFFFFF" size="6">Some text.</font>

P. 17

## Anchor

- Usually used as a link (<a>)
- Attribute commonly used:
  - href
    - Your target URL
  - target
    - "\_blank" open a new window for the link
- Example:
  - Open in the same window:
    - <a href="http://www.google.com">To Google</a>
  - Open in a new window:
    - <a href="http://www.google.com" target="\_blank">To Google</a>

P. 18

## Image

- Add an image (<img>)
  - There are many attributes. Try it out yourself!

P. 19

## Table (1/2)

- Draw a table (<table>)
  - It is one of the most useful elements
    - It is very easy to understand. Yet, believe it or not, it is also one of the most difficult element to use in practice.
      - Different browser has different rendering mechanism. It is extremely difficult to make all browsers display a table exactly the same.
      - For building a complex web site, machine generated code can seldom generate precisely what you want to have.
  - It consists of row (<tr>) and column (<td>)

```

10 <table border="1" width="100%">
11 <tr>
12 <td>1</td>
13 <td>2</td>
14 <td>3</td>
15 </tr>
16 <tr>
17 <td>4</td>
18 <td>5</td>
19 <td>6</td>
20 </tr>
21 <tr>
22 <td>7</td>
23 <td>8</td>
24 <td>9</td>
25 </tr>
26 </table>

```

1	2	3
4	5	6
7	8	9

P. 20

## Table (2/2)

- Play with the Table by yourself. You will never know how tricky is Table unless you are working in some real project.

P. 21

## Layer

- Besides Table, Layer (<div>) is another extremely useful "layout element" in HTML

```

10 <body>
11 <div style="position: absolute; width: 100px; height: 100px; z-index: 1; left: 26px; top: 44px; border-style: solid; border-color: #FFFF00; background-color: #FF0000" id="layer1">
12 This is a layer
13 </div>
14 <div style="position: absolute; width: 100px; height: 100px; z-index: 1; left: 57px; top: 41px; border-style: solid; border-color: #000000; background-color: #000000" id="layer2">
15 This is another layer</div>
16 </div>
17 </body>

```

- The z-index property specifies the stack order of an element.



P. 22

## Span

- Span (<span>) is a very useful rendering element which is frequently used in conjunction with CSS
  - Usual usage:
    - <span class="some\_style"><any\_tag>some data</any\_tag></span>
  - A text with a span element that can be styled with CSS
- The <span> tag provides no visual change by itself
- The <span> tag provides a way to add a hook to a part of a text or a part of a document.
  - When the text is hooked in a span element you can add styles to the content, or manipulate the content with for example JavaScript.

P. 23

## Others

- Other tags that are also very useful:
  - Bold
    - <b>Some text</b>
  - Italic
    - <i>Some text</i>
  - Underline
    - <u>Some text</u>
  - List (note – you can have nested list)

```

<p>An ordered list:</p>
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<p>An unordered list:</p>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

```

An ordered list:

1. Coffee
2. Tea
3. Milk

An unordered list:

- Coffee
- Tea
- Milk

P. 24

## CSS

- What is CSS?
  - Cascading Style Sheet
  - Used to describe the presentation (that is, the look-and-feel) of a document written in a mark-up language (e.g. HTML).
- Since HTML can already specify the look-and-feel of the elements, why do we need to learn/have CSS?
  - Well, consider the following situation...
    - You have written a web page for your client
      - It consists of 100 <font> tags
    - Your client suddenly asks you to change the font size of the page
      - Traditionally, you need to edit 100 <font> tags!!!
        - » Edit 100 lines at least. Time consuming... Error... Etc...
      - Yet, with CSS, it is possible that you just need to edit 1 line ONLY!!!

P. 25

## Writing CSS (1/4)

- Define the style in the head section:
  - E.g. You want to define a class called "majorFont":
 

```
<head>
<style type="text/css">
.majorFont{
  font-family: "Arial";
  font-size: 9pt;
}
</style>
</head>
```
  - And then, in the body section, you may write:
    - <p class="majorFont">Some text</p>
    - <span class="majorFont"><p>Some text</p></span>
    - <table class="majorFont">.....</table>

P. 26

## Writing CSS (2/4)

- You can have more than one class in the tag!
  - E.g. You have defined another class called "majorColor"
 

```
<style type="text/css">
.majorColor{
  color: #FFFFFF;
}
</style>
```
  - Both have the same effect:
    - <p class="majorFont majorColor">Some text</p>
    - <span class="majorColor"><p class="majorFont">Some text</p></span>

P. 27

## Writing CSS (3/4)

- If you have multiple class in a tag, then the latest one may override the properties of those previous.
  - E.g. two class:
 

```
<style type="text/css">
.font{
  font-family: "Arial";
  font-size: 9pt;
  color: #CDCDCD;
}
.color{
  color: #FFFFFF;
}
</style>
```
  - Any differences between the following two cases?
    - <p class="font color">Some text</p>
    - <p class="color font">Some text</p>

P. 28

## Writing CSS (4/4)

- Although it is possible to write the inline style CSS
  - For example:
 

```
<p style="font-family: "Arial"; font-size: 9pt;
color: #CDCDCD;">Some text</p>
```
- It is not recommended to do so (unless the style is very unique for that component, such as layer).
  - Why?

P. 29

## More About CSS

- CSS is not just designed for writing HTML more easily.
- CSS is designed for rendering the components in the HTML page more efficiently and provide more flexibility.
  - Traditional HTML tags/attributes may not be able to support some CSS rendering style.
- Too many styles for CSS.
  - Impossible to discuss them one by one.
  - Reference:
    - <http://www.w3schools.com/css/>

P. 30

### Linking An External CSS File (1/4)

- Why do we need to link the CSS file externally?
  - Well, consider the following situation...
    - You have developed a web site for your client
      - It consists of 100 HTML pages
    - Your client suddenly asks you to change the font size of all 100 HTML pages...
      - If you embedded CSS in the HTML document, you still need to edit 100 pages!
      - Yet, with external CSS, it is possible that you just need to edit 1 line ONLY!!!

P.31

### Linking An External CSS File (2/4)

- In order to link...
  - Create a file with CSS contents:
 

```
.font{
    font-family: "Arial";
    font-size: 9pt;
}
.color{
    color: #FFFFFF;
}
```

    - **Note that no <style> is necessary!!!**
  - Save the file as xxx.css
    - E.g. I save it as style.css
  - In the head section, write:
 

```
<head>
<link rel="stylesheet" href="style.css"
type="text/css" />
</head>
```

P.32

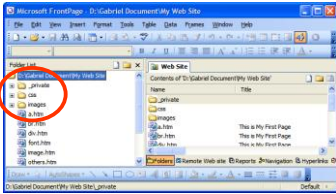
### Linking An External CSS File (3/4)

- You may include as many CSS files as you like!
  - Many CSS files. Each for some specific issue:
 

```
<head>
<title>An Example</title>
<link rel="stylesheet" href="environment.css"
type="text/css" />
<link rel="stylesheet" href="font.css" type="text/css" />
<link rel="stylesheet" href="table.css" type="text/css" />
<link rel="stylesheet" href="link.css" type="text/css" />
</head>
```

P.33

### Linking An External CSS File (4/4)

- For better resource management, we usually store CSS files into a directory called "css":
 

- And remember to change the path of the CSS files:

```
<link rel="stylesheet" href="css/environment.css" type="text/css" />
<link rel="stylesheet" href="css/font.css" type="text/css" />
<link rel="stylesheet" href="css/table.css" type="text/css" />
<link rel="stylesheet" href="css/link.css" type="text/css" />
```

P.34

### One Useful Style (1/2)

- Can you change the link color or style using HTML only (without CSS)?
  - Impossible!
  - Yet, it is possible with the help of CSS!
    - But it is very tricky...



P.35

### One Useful Style (2/2)

- CSS changing the link style:
 

```
.link01 a:link{ /* unvisited link */
    text-decoration: none;
    color: #00FFFF;
}
.link01 a:visited{ /* visited link */
    text-decoration: line-through;
    color: #00FFFF;
}
.link01 a:hover{ /* mouse over link */
    text-decoration: underline;
    color: yellow;
}
.link01 a:active{ /* selected link */
    text-decoration: overline;
    color: #00FFFF;
}
```
- To use it:
  - <a href="..." class="link01">Some text</a>

P.36

## Summary

- We have covered:
  - Major HTML elements
  - How to use CSS
- Next lecture:
  - Programming in HTML – JavaScript
  - Some advanced issues on client-side development

P. 37

## References

- HTML tutorial:
  - <http://www.w3schools.com/html/DEFAULT.asp>
- CSS tutorial:
  - <http://www.w3schools.com/css/>
- Some useful development tools:
  - CSS table and border:
    - <http://www.somacon.com/p141.php>
  - Layer positioning:
    - <http://www.barelyfitz.com/screencast/html-training/css/positioning/>

P. 38