Lecure 2
HTML

Outline

- HTML Overview
- Form
- Web Standards

HTML

- Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications.
- Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

Wikipedia

HTML timeline

- 1989: Berners-Lee wrote a memo proposing an **Internet**-based **hypertext** system.
- 1995: HTML 2 becomes an official standard language by a publication called RFC 1866.
- 1997: HTML 4 is proposed by W3C standard body, adding style sheets, scripting, frames, embedding objects, internationalization, and accessibility for disabilities.
- 1999: HTML 4.01 the last major version of the language is published by W3C. A majority of pages on the Web today still use it as their started language.
- 2001-01: XHTML, HTML based on XML
- 2014 html5
- 2016 html5.1
- 2017 html5.2

HTML and XHTML

- HTML was originally an application Standard Generalized Markup Language (SGML)
  - SGML is a very flexible markup language
  - requires a relatively complex, lenient, and generally custom parser
- XHTML:
  - an application of XML, a more restrictive subset of SGML
  - true XHTML documents allow for automated processing to be performed using standard XML tools

Markup Languages

- Markup:
  - Embedded codes in documents
  - Codes are called ‘tags’
  - Codes
    - Describe the structure documents
    - Include instructions for processing
- Markup language:
  - Computer language for describing syntax of tags
  - May be used with other tools to specify rendering

Logical Markup

- Logical markup:
  - Describes parts of document
  - Does not specify how to render
- Example:
  - This is very important
  - This is very important

Logical Markup

- Presentation is client's 'decision'
- When client cannot present then there is graceful degradation
  - <img alt="image description" src="foo.gif">

Markdown

- Markdown 是一种轻量级标记语言，它允许人们使用易读易写的纯文本格式编写文档。
- Markdown 语言在 2004 由约翰·格鲁伯（英语：John Gruber）创建。
- Markdown 编写的文档可以导出 HTML 、Word、图像、PDF、Epub 等多种格式的文档。
- Markdown 编写的文档后缀为 .md, .markdown。
- Markdown 应用
  - Markdown 能被使用来撰写电子书，如：Gitbook。
  - 当前许多网站都广泛使用 Markdown 来撰写帮助文档或是用于论坛上发表消息。例如：GitHub、简书、reddit、Diaspora、Stack Exchange、OpenStreetMap 、SourceForge等。

```
# h1 标题
## h2 标题
### h3 标题

## 文本样式

**This is bold text**

__This is bold text__

*This is italic text*

_This is italic text_

~~Strikethrough~~

## 代码

Inline `code`

Indented code

    // Some comments
    line 1 of code
    line 2 of code
    line 3 of code


Block code "fences"

```
Sample text here...
```

Syntax highlighting

```  js
var foo = function (bar) {
  return bar++;
};

console.log(foo(5));
```
```

h1 标题

h2 标题

h3 标题

文本样式

This is bold text

This is bold text

This is italic text

This is italic text

~~Strikethrough~~

代码

Inline `code`

Indented code

```
// Some comments
line 1 of code
line 2 of code
line 3 of code
```

Block code "fences"

```
Sample text here...
```

Syntax highlighting

```js
var foo = function (bar) {
  return bar++;
};

console.log(foo(5));
```

how to write (x)HTML

- Editors
- Generator
  - https://github.com/Microsoft/ailab/tree/master/Sketch2Code
  - https://sketch2code.azurewebsites.net/

YOUR SKETCH

YOUR HTML

YOUR SKETCH

YOUR HTML

Workflow
Wireframes
Pictures
App
Cloud
AI model
AI service
Markup generator
User draws on a whiteboard a UI screen sketch
A Windows app takes pictures and submit them to AI Service
AI service runs the image against a prebuilt model
AI service generates HTML or any other markup language
Results are displayed in the App
1
2
3
4
5

Sketch2Code Architecture Diagram

HTML Basics

- 3 Parts to an XHTML or HTML document
  - DOCTYPE
    - What DTD are you using
  - Head
    - Meta information
    - Only <title> is required
  - Body
    - Text to render

Structure of XHTML 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">
        <head>
            <title>Hello World</title>
        </head>
        <body>
            page contents
        </body>
</html>
```

```
 <!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

Block and inline elements (explanation)

- block elements contains an entire large region of content
  - examples: paragraphs, lists, table cells
  - the browser places a margin of whitespace between block elements for separation
- inline elements effects a small amount of content
  - examples: bold text, code fragments, images
  - the browser allows many inline elements to appear on the same line
  - must be nested inside a block element

Comments: <!-- ... -->

- comments to document your HTML file or “comment out” text

```
<!-- My web page, by Student SS 330, Spring 2048 -->
<p>SS courses are <!-- NOT --> a lot of fun!</p>
```

HTML

```
SS courses are a lot of fun!
```

output

- comments are still useful for disabling sections a page
- comments cannot be nested and cannot contain a –
- many web pages are not thoroughly commented (or at all)
  - comment is a communicative approach, to explain your designs and purposes to your colleagues, or even yourself sometime later.
  - comment is not for browsers of end users, but for developers and designers.

Page title: <title>

- describes the title of the Web page

```
<title>Chapter 2: HTML Basics</title>
```

- placed within the **head** of the page
- displayed in the Web browser's title bar and when bookmarking the page

Page meta data: <meta> Programming

describe meta data of the Web page

```
<meta name="description" content="introduction of XXX" />
```

```
<meta http-equiv="Content-Type" content="text/html;
charset=gbk" />
```

- placed within the **head** of the page
- chartset is very significant in pratice, and we often use utf-8 for language other than English
  - character encoding and decoding, where, when, and how?

Web page metadata: <meta>

```
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<meta name="description" content="Authors' web site for
Building Java Programs." />
<meta name="keywords" content="java, textbook" />
```

- name
  - author
  - description
  - keywords
  - generator
  - revised
- http-equiv
  - content-type
  - expires
  - refresh

Character Encoding

- manipulating and representing of characters in computer
  - the bit length of the character
  - and the proper visual symbol
  - encoding vs. decoding

- chartset
  - ASCII(basic 7b, extension 1B),
  - iso-8859-1/latin-1 (West Europe,1B)
  - GB2312 (2B, Simplified Chinese)
  - GBK(2B, S. & T. Chinese)
  - BIG5 (2B, Traditional Chinese)
  - GB18030 (1,2,4B, Eastern Asia)
  - *Unicode (650 languages)*
    - *UTF-8 (1,2,3,4B , Chinese 3B)*
    - *UTF-16 (2B, 4B, Chinese 2B)*
    - *UTF-32 (4B, future)*
  - *UCS*
    - *UCS-2 (2B, comparable with UTF-16)*
    - *UCS-4 (4B, future)*

Paragraph: <p>

- placed within the body of the page

```
<p>You're not your job. You're not how much money you have in the bank. You're not the car you drive. You're not the contents of your wallet. You're not your khakis. You're the all-singing, all-dancing crap of the world.</p>
```

html

You're not your job. You're not how much money you have in the bank. You're not the car you drive. You're not the contents of your wallet. You're not your khakis. You're the all-singing, all-dancing crap of the world.

output

Line break:

- forces a line break in the middle of a block element (inline)

```
<p>Teddy said it was a hat, <br /> So I put it on.</p> <p>Now Daddy's
sayin', <br /> Where the heck's the toilet plunger gone?</p>
```

Teddy said it was a hat,
So I put it on.
Now Daddy's sayin',
Where the heck's the toilet plunger gone?

- br should be immediately closed with />

Links: <a>

- links, or “anchors”, to other pages (inline)

```
<p>
    Search
    <a href=“http://www.google.com/”>Google</a> or our
    <a href=“lectures.html”>Lecture Notes</a>
</p>
```

Search Google or our Lecture Notes.

- uses the href attribute to specify the destination URL
  - can be absolute (to another Web site) or relative (to another page on this site)
- anchors are inline elements; must be placed in a block element such as p or h1

Links: <a>

- hover a link in a browser
- descriptiveness

  Click here to check your course schedule

  Please check your course schedule

  Course Schedule (please check yours before March 15!)”

- What’s principle applied here?
- Kind.
  - you are kind to your Web page readers by making the page descriptive, which in turn let them understand easier

Images: <img>

- inserts a graphical image into the page (inline)

```
<img src="url" alt="some_text">
```

- the src attribute specifies the image URL
- XHTML also requires an alt attribute describing the image

More about images

- if placed inside an anchor, the image will become a link
-

```
<a href="http://theonering.net/">
      <img src="hh.jpg" alt="some-text" width="304"
height="228">
</a>
```

Image Formats

- GIF
  - 透明、动画、无损、256
- JPEG
  - 有损
- PNG
  - 真彩色和调色板、透明、无损、
- webp
  - 有损、压缩率高

Phrase elements: ,

- em: emphasized text (usually rendered in italic)
- strong: strongly emphasized text

```
<p>
    HTML is <em>really</em>,
    <strong>REALLY</strong> fun!
</p>
```

- rendered in bold

HTML is *really*, **REALLY** fun!

- as usual, the tags must be properly nested for a valid page
- em vs. i, strong vs. b

Nesting tags

- Bad:

```
<p>
    HTML is <em>really,
    <strong>REALLY</em> lots of </strong> fun!
</p>
```

- tags must be correctly nested
  - a closing tag must match the most recently opened tag
- the browser may render it correctly anyway, but it is invalid XHTML

Table: <table>, <tr>, <td>, <th>, <caption>

```
<table>
<caption>Smart Guys</caption>
<tr><th>name</th><th>gender</th></tr> <tr><td>Bill</td><td>male</td></tr> <tr><td>Susan</td><td>female</td></tr>
</table>
```

Smart Guys

name	gender
Bill	male
Susan	female

- Never use Table for layout~!

Quotations: <blockquote>

- a lengthy quotation (block)

```
<p>As Lincoln said in his famous Gettysburg Address:</p>
<blockquote>
<p>Fourscore and seven years ago, our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.</p>
</blockquote>
```

As Lincoln said in his famous Gettysburg Address:

> *Fourscore and seven years ago, our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.*

output

Inline quotations: <q>

- a short quotation (inline)

```
<p>Quoth the Raven, <q>Nevermore.</q></p>
```

Quoth the Raven, “Nevermore”.

- Why not just write the following?
  - Quoth the Raven, "Nevermore."

We don’t use “ mark for two reasons:

- XHTML shouldn’t contain literal quotation mark characters; they should be written as "
- using <q> allows us to apply CSS styles to quotations

HTML character entities

- a way of representing any Unicode character within a Web page

character (s)	entity
<>	< >
é è ñ	é è ñ
TM ©	™ ©
π δ Δ	π δ Δ
И	И
" &	" &

- How would you display the text & on a web page?

HTML-encoding text

```
&lt;p&gt; &lt;a href=&quot;http://google.com/search?q=marty&amp;ie=utf- 8&amp;aq=t&quot;&gt; Search Google for Marty &lt;/a&gt; &lt;/p&gt;
```

HTML

```
<p> <a href="http://google.com/search?q=marty&ie=utf- 8&aq=t"> Search Google for Marty </a> </p>
```

output

- To display the link text in a Web page, its special characters must be encoded as shown above

Outline

- HTML overview
- Form
- Web Standards

What are forms?

- <form> is just another kind of HTML tag
- HTML forms are used to create (rather primitive) GUIs on Web pages
  - Usually the purpose is to ask the user for information
  - The information is then sent back to the server
- A form is an area that can contain form elements
  - The syntax is: <form parameters> ...form elements... </form>
  - Form elements include: buttons, checkboxes, text fields, radio buttons, drop-down menus, etc
    - Other kinds of HTML tags can be mixed in with the form elements
  - A form usually contains a Submit button to send the information in the form elements to the server
  - The form's parameters tell JavaScript how to send the information to the server (there are two different ways it could be sent)
  - Forms can be used for other things, such as a GUI for simple programs

Form example

- must wrap the form's controls in a block element such as div, fieldset, etc.

```
<form>
    <div class="form-group">
        <label for="inputEmail">Email</label>
        <input type="email" class="form-control" id="inputEmail" placeholder="Email">
    </div>
    <div class="form-group">
        <label for="inputPassword">Password</label>
        <input type="password" class="form-control" id="inputPwd" placeholder="Password">
    </div>
    <div class="checkbox">
        <label><input type="checkbox"> Remember me</label>
    </div>
    <button type="submit" class="btn">Login</button>
</form>
```

The <form> tag

- The <form arguments> ... </form> tag encloses form elements (and probably other HTML as well)
- The arguments to form tell what to do with the user input
  - action="url" (required)
    - Specifies where to send the data when the Submit button is clicked
  - method="get" (default)
    - Form data is sent as a URL with ?form_data info appended to the end
    - Can be used only if data is all ASCII and not more than 100 characters
  - method="post"
    - Form data is sent in the body of the URL request
    - Cannot be bookmarked by most browsers
  - target="target"
    - Tells where to open the page sent as a result of the request
    - target= _blank means open in a new window
    - target= _top means use the same window

The <input> tag

- Most, but not all, form elements use the input tag, with a type="…" argument to tell which kind of element it is
  - type can be text, checkbox, radio, password, hidden, submit, reset, button, file, or image
- Other common input tag arguments include:
  - name: the name of the element
  - value: the “value” of the element; used in different ways for different values of type
  - readonly: the value cannot be changed
  - disabled: the user can’t do anything with this element
  - Other arguments are defined for the input tag but have meaning only for certain values of type

examples

- <input type="text" name="firstname" />
- <input type="radio" name="sex" value="male" /> Male
- <input type="checkbox" name="bike" />

Think of <input>

- So many types of input, why NOT use elements instead?
  - <input type=¨text〃 ... /> → <text/> or <text></text>
  - <input type=¨checkbox〃 ... /> → <checkbox ... />
- In fact, it is just a bad design decision when form was firstly designed and introduced into html in 1996, and we follow it so far ...,
- another flaw
  - checked=¨checked〃 ..., is it weird?
- Lessons:
  - Reality is never, ever perfect
  - BUT we will try out best to make it perfect

Hidden fields

- <input type="hidden" name="hiddenField" value="xxx">
  <-- right there, don't you see it?

A hidden field: <-- right there, don't you see it?

- What good is this?
  - All input fields are sent back to the server, including hidden fields
  - This is a way to include information that the user doesn’t need to see (or that you don’t want her to see)
  - The value of a hidden field can be set programmatically (by JavaScript) before the form is submitted

Outline

- Basic HTML
- Form
- Web Standards

Web standards

- Why use XHTML and Web standards?
  - more rigid and structured language
  - more interoperable across different web browsers
  - more likely that our pages will display correctly in the future
  - can be interchanged with other XML data SVG (graphics), MathML, MusicML, etc.
- HTML5

XHTML 1.0 vs HTML 4.01

- all tags must be closed
- all tags must be correctly nested
- all tag attributes must be enclosed in quotation marks
- the & character can't be used on its own, and use & instead
- tags are case-sensitive and must be all in lowercase
- attributes cannot be minimized any more
- XHTML document must start with new XML declaration
  - <?xml version="1.0" encoding="UTF-8"?>
- different **DOCTYPE** declaration
  - <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
- the **<html>** tag requires an **xmlns** attribute

Ref.

- https://www.w3schools.com/
- https://www.khanacademy.org/computing/computer-programming/html-css/intro-to-html/pt/html-basics
- https://classroom.udacity.com/nanodegrees/nd001-cn-preview/parts/63e29129-49ba-41cf-87d6-f98246da3f24/modules/166575d1-e43b-4cbf-90da-26c36c6b3c96/lessons/cf47a211-fbc7-41fa-a9c5-3c0c6613eb05/concepts/74229205980923

Thanks!!!