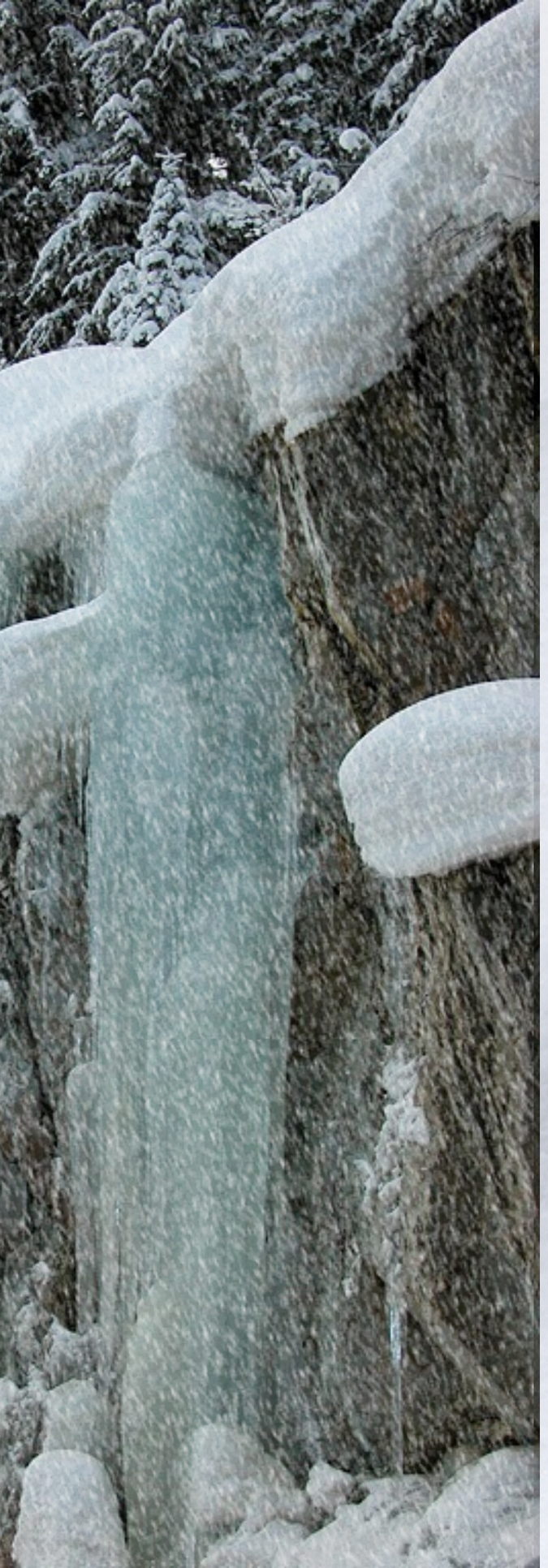
Lecture 2
HTML

Play



Outline

- HTML Overview
- Form
- Web Standards

Hypertext Markup Language (HTML)

- 1993: Initial official proposed description of HTML submitted to the IETF standards group.
- 1995: HTML 2 becomes an official standard language by a publication called RFC 1866.
- 1996-97: HTML 3.2 standardizes various features including forms, tables, image maps, and internationalization.
- 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

HTML

- Hypertext Markup Language
  - describes the content and structure of information on a Web page
- Intended to be maximally portable
  - Logical markup
  - Graceful degradation of presentation

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">

Why HTML became 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

XHTML Basics

- Very few real changes from HTML
- But more strict

- All tags are in lowercase
- All tags must be closed
  - Empty tags
  - Paired tags

XHTML 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

- the header describes the page and the body contains the page's contents
- an HTML page is saved into a file ending with extension .html

```
<?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>
        information about the page
    </head>
    <body>
        page contents
    </body>
</html>
```

XHTML Basics

- Tags
  - Elements
  - Attributes
- Entities
- Comments

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

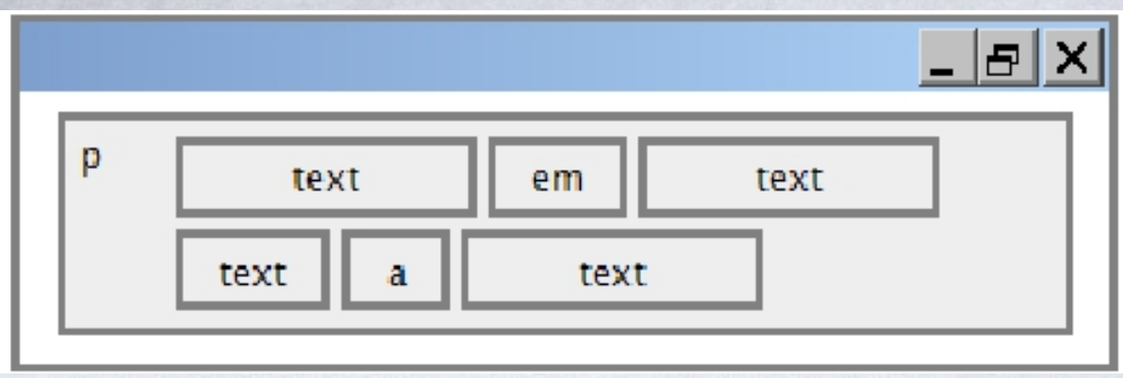
Character Encoding

- 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)

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



Paragraph: <p>

- placed within the body of the page

```html
<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>
```

```output
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 theall-singing, all-dancing crap of the world.
```

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 />

Horizontal rule: <hr>

- headings to separate major areas of the page (block)

```
<p>First paragraph</p>
<hr />
<p>Second paragraph</p>
```

First paragraph

Second paragraph

- should immediately closed with />

More about HTML tags

- some tags can contain additional information called attributes
  - syntax: <element **attribute1**=“value1” **attribute2**=“vaule2”>content</element>
  - example: Next page
- some tags don’t contain content; can be opened and closed in one tag
  - syntax: <element **attribute1**=“value1” **attribute2**=“vaule2” />
  - example: <hr />
  - example:

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="imges/gollum.jpg" alt="Gollum form LOTR" />
```

- 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
- the **title** attribute specifies an optional tooltip
- images/gandalf.jpg vs. /images/gandalf.jpg

```
<a href="http://theonering.net/">
        <img src="images/gandalf.jpg" alt="Gandalf from LOTR"
title="You shall not pass! />
</a>
```

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

Unordered list: <ul>,

- ul represents a bulleted list of items (block)
- li represents a single item within the list (block)

```
<ul>
    <li>No shoes</li>
    <li>No shirt</li>
    <li>No problem!</li>
</ul>
```

html

- No shoes
- No shirt
- No problem!

output

More about unordered lists

a list can contain other lists:

```
<ul>
    <li>Simpsons:
        <ul>
            <li>Homer</li>
            <li>Marge</li>
        </ul>
    </li>
    <li>Family Guy:
        <ul>
            <li>Peter</li>
            <li>Lois</li>
        </ul>
    </li>
</ul>
```

- Simpsons:
  - Homer
  - Marge
- Family Guy:
  - Peter
  - Lois

Ordered list:

- ol represents a numbered list of items (block)

```
<p>RIAA business model:</p>
<ol>
    <li>Sue customers for copying music</li>
    <li>???</li>
    <li>Profit!</li>
</ol>
```

RIAA business model:

1. Sue customers for copying music
2. ???
3. Profit!

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~!

Definition list: <dl>, <dt>, <dd>

dl represents a list of definitions of terms(block)

dt represents each term, and dd its definition

```
<dl>
   <dt>newbie</dt><dd>one who does not have mad skills</dd>
   <dt>own</dt><dd>to soundly defeat
       (e.g. I owned that newbie!)</dd>
   <dt>frag</dt> <dd>a kill in a shooting game</dd>
</dl>
```

newbie

one who does not have mad skills

own

to soundly defeat (e.g. I owned that newbie!)

frag

a kill in a shooting game

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	&#1048;
" &	" &

- 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;amp;ie=utf- 8&amp;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

Computer code: <code>

- code: a short section of computer code (usually rendered in a fixed-width font)

```
<p> The <code>ul</code> and <code>ol</code> tags make lists. </p>
```

The ul **and** ol **tags make lists.**

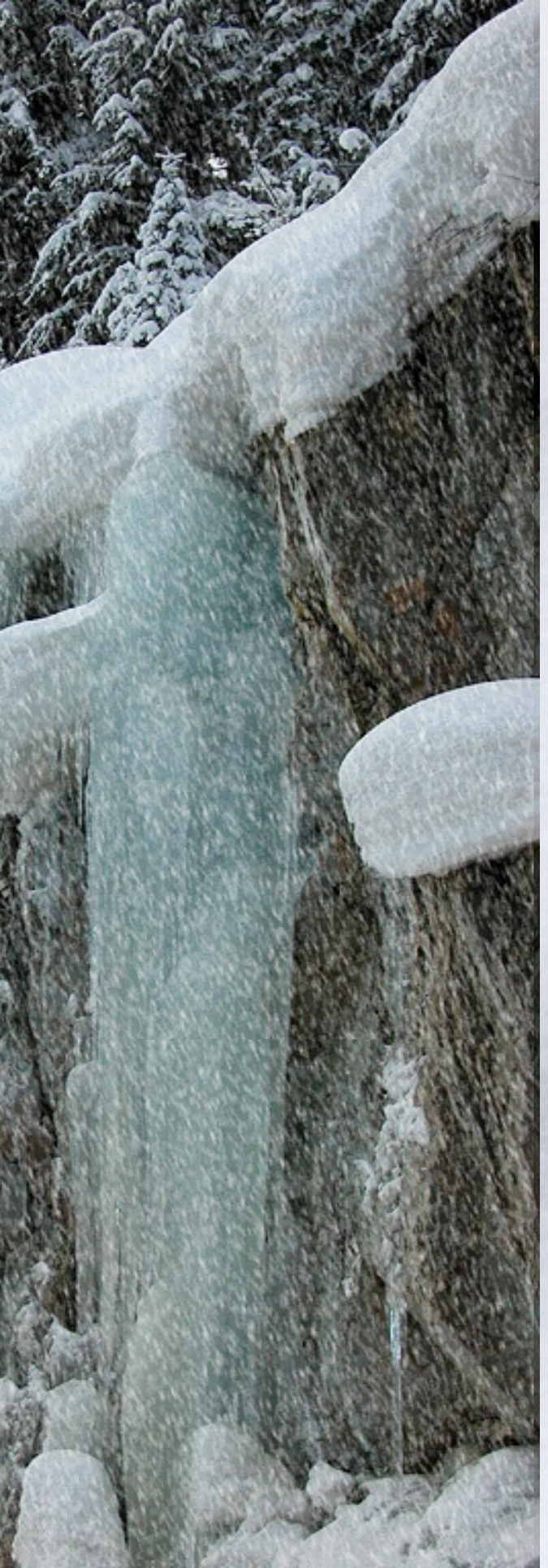
Preformatted text: <pre>

a large section of pre-formatted text (block)

```
<pre>
            Steve Jobs speaks loudly
                reality distortion
              Apple fans bow down
</pre>                                          HTML
```

```
Steve Jobs speaks loudly
    reality distortion
Apple fans bow down                             output
```

- displayed with exactly the whitespace / line breaks given in the text
- shown in a fixed-width font by default
- how would it look if we had instead enclosed it in code tags?



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 he 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.

```HTML
<form action="http://www.google.com/search">
  <div>
    Let's search Google:
    <input name="q" />
    <input type="submit" />
  </div>
</form>
```

output

Let's search Google: [ ] 提交查询

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

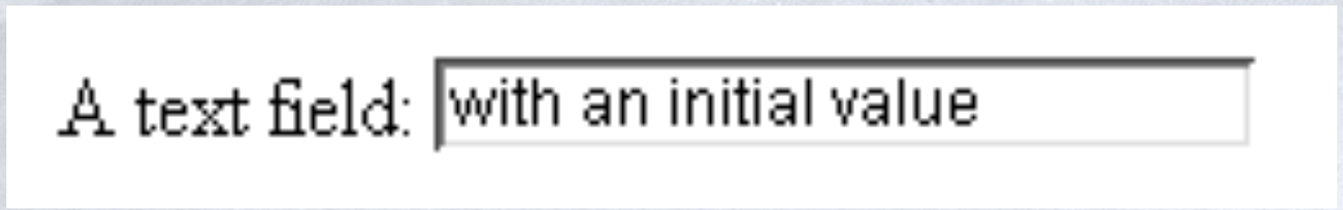
The <input> tag

- 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

Text input

A text field:

```
<input type="text" name="textfield" value="with an initial value">
```



A multi-line text field

```
<textarea name="textarea" cols="24" rows="2">Hello</textarea>
```

A password field:

```
<input type="password" name="textfield3" value="secret">
```

- Note that two of these use the input tag, but one uses textarea

Buttons

- A submit button:

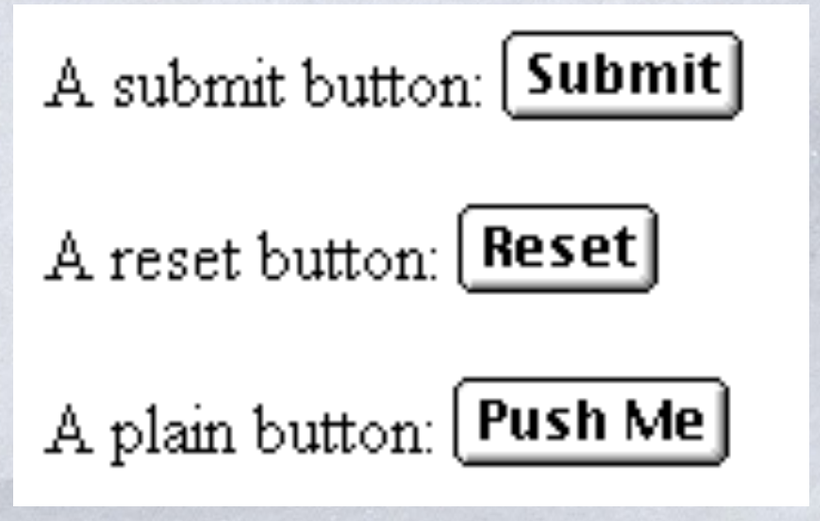
```
<input type="submit" name="Submit" value="Submit">
```

- A reset button:

```
<input type="reset" name="Submit2" value="Reset">
```

- A plain button:

```
<input type="button" name="Submit3" value="Push Me">
```



submit: send data
reset: restore all form elements to their initial state
button: take some action as specified by JavaScript

- Note that the type is input, not “button”

Checkboxes

- A checkbox:

```
<input type="checkbox" name="checkbox”
       value="checkbox" checked>
```

A checkbox: ☑

type: "checkbox"
name: used to reference this form element from JavaScript
value: value to be returned when element is checked

Note that there is no text associated with the checkbox—you have to supply text in the surrounding HTML

Radio buttons

- Radio buttons:
  <input type="radio" name="radiobutton" value="myValue1">
  male
  <input type="radio" name="radiobutton" value="myValue2" checked>
  female

- If two or more radio buttons have the same name, the user can only select one of them at a time
  - This is how you make a radio button “group”
- If you ask for the value of that name, you will get the value specified for the selected radio button
- As with checkboxes, radio buttons do not contain any text

Drop-down menu or list

- A menu or list:

```
<select name="select">
    <option value="red">red</option>
    <option value="green">green</option>
    <option value="BLUE">blue</option>
</select>
```

- Additional arguments:
  - size: the number of items visible in the list (default is "1")
  - multiple: if set to "true", any number of items may be selected (default is "false")

Hidden fields

- <input type="hidden" name="hiddenField" value="nyah">
  <-- 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

Reset buttons

- when clicked, returns all form controls to their initial values
- specify custom text on the button by setting its value attribute

```
Name: <input type="text" name="name" /> <br />
Food: <input type="text" name="meal" value="pizza" /> <br />
<label>Meat? <input type="checkbox" name="meat" /></label> <br />
<input type="reset" />
```

HTML

Name:

Food: pizza

Meat?

重置 提交查询

output

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

Text labels: <label>

- associates nearby text with control, so you can click text to activate control
- can be used with checkboxes or radio buttons
- either wrap the input elements or target input elements with id specified via the “for” attribute
- label element can be targeted by CSS style rules
- reasons for preferring label than text:
  - functionality: can be directly clicked on
  - styling: can be styled by CSS rules
  - accessibility: screen reader will read it when selected

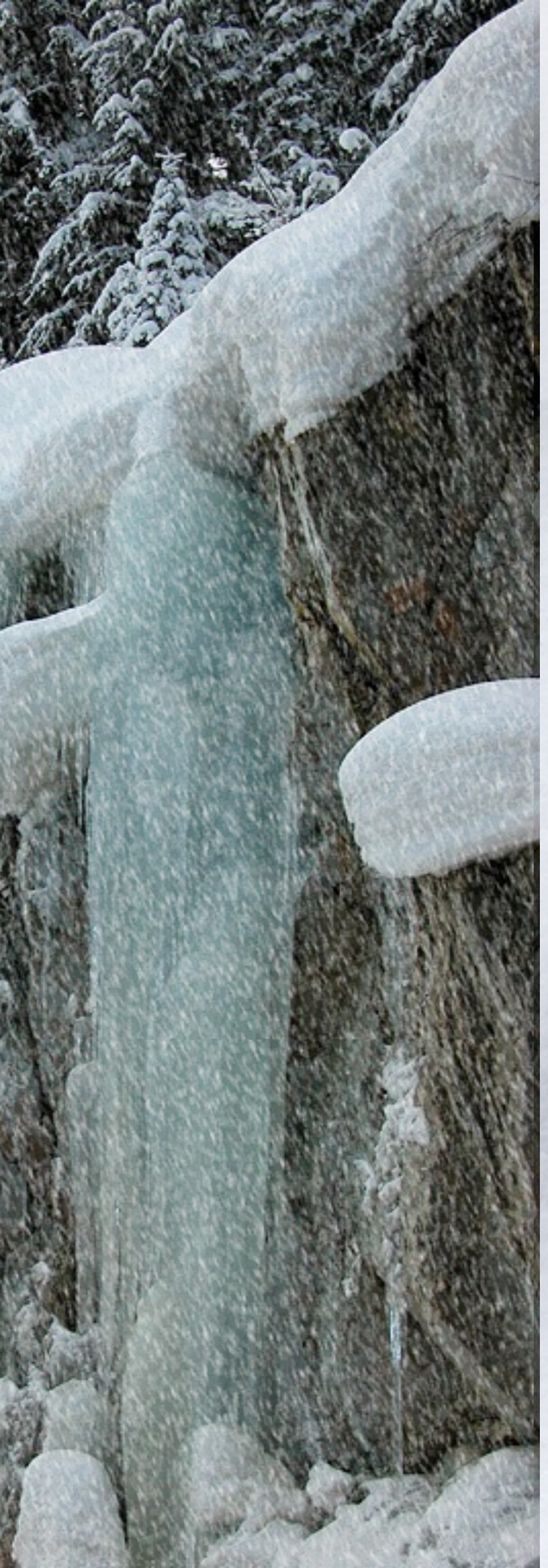
```html
<label><input type="radio" name="cc" value="visa" checked="checked" /> Visa</label>
<label><input type="radio" name="cc" value="mastercard" /> MasterCard</label>
<label><input type="radio" name="cc" value="amex" /> American Express</label>
```

HTML

output

A complete example

```
<html>
<head>
<title>Get Identity</title>
<meta http-equiv="Content-Type" content="text/html;
          charset=iso-8859-1">
</head>
<body>
<p><b>Who are you?</b></p>
<form method="post" action="">
  <p>Name:
    <input type="text" name="textfield">
  </p>
  <p>Gender:
    <input type="radio" name="gender" value="m">Male
    <input type="radio" name="gender" value="f">Female</p>
  </form>
</body>
</html>
```



Outline

- Basic HTML
- More HTML Elements
- Form
- Web Standards

Web standards

- It is important to write proper XHTML code and follow proper syntax.
- 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.

XHTML versions

- XHTML 1.0 (W3C Recommendation)
  - HTML 4.01 with XML syntax
  - XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset
- XHTML 1.1 (W3C Recommendation)
  - module-based XHTML
- XHTML 1.2
  - improved Semantic Web support through RDFa
  - draft, not widely adopted
- XHTML 2.0
  - not backward compatible
  - developing
- XHTML 5
  - part of HTML5 specification (ongoing)

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

Where are the tools?

- HTML Tidy
  - http://sourceforge.net/projects/tidy/
- The validator
  - http://validator.w3.org/

W3C XHTML Validator

```
<p>
   <a href="http://validator.w3.org/check/referer">
      <img src="http://www.w3.org/Icons/valid-xhtml11"
      alt="Validate" />
   </a>
</p>
```

validator.w3.org

- check your HTML code to make sure it follows the official XHTML syntax
- more picky than the browser, which may render bad XHTML correctly

Thanks!!!