

HTML

Chapter 2

Before we start

2

- HTML is the bone and organs, CSS is the skin, JS is the soul
- HTML: Defines the structure of the webpage. Do NOT control the style of webpage.
- Is HTML a programming language?
 - ▣ NO.

HTML是一种描述性语言，而不是编程语言

Text editor

3

- ❑ Things like MS Word, IDE and notepad is **NOT** a so-called (programming) text editor
- ❑ We recommend to use:
 - ▣ Atom
 - ▣ Sublime
 - ▣ VS Code
 - ▣ Notepad++

Outline

4

- HTML tags
 - ▣ Text
 - ▣ Links and images
 - ▣ Lists
 - ▣ Tables
 - ▣ Forms
 - ▣ Metadata and embedded content
- Syntax and Grammar of HTML
 - ▣ Validator
 - ▣ Conforming HTML documents

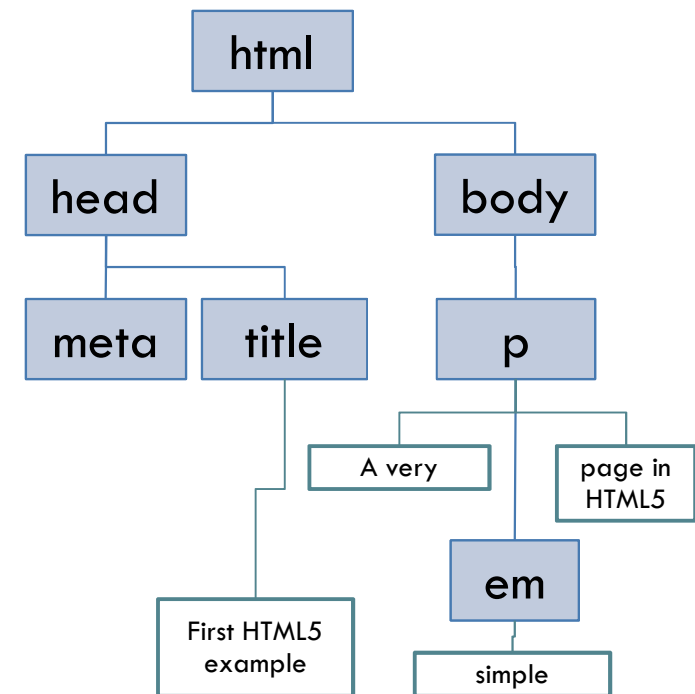
HTML document

5

- An HTML document is a text file with some markup tags that organize the data as a tree of elements

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>First HTML5 example</title>
  </head>
  <body>
    <p>A very <em>simple</em> page in HTML5</p>
  </body>
</html>
```

First example in HTML5



Example: XHTML version

6

- XHTML version of the same document
Extensible Hypertext Markup Language, 提供更严格、更可维护和更一致的Web文档标记方式

```
<!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" lang="en" xml:lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>First XHTML example</title>
  </head>
  <body>
    <p>A very <em>simple</em> page in XHTML</p>
  </body>
</html>
```

First example in valid XHTML

Markup tags

7

- An **element** starts with a **start tag** and ends with an **end tag**
- An empty element may combine the start and end tags
- **Attributes** in a start tag provides further information about an element
- Five characters have special meaning: `<` `>` `&` `'` `"`

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>First HTML5 example</title>
  </head>
  <body>
    <p>A very <em>simple</em>
      page in HTML5</p>
  </body>
</html>
```

Symbol	<	>	&	"	'
Entities	<	>	&	"	'

Basic structure

8

- An HTML doc has the root element `<html>`, which contains two children:
 - ▣ `<head>`, which describes the HTML doc
 - `<meta>` indicates the character encoding of the doc
 - `<title>` appears in title bar of the browser
 - Other elements ...
 - ▣ `<body>`, which provides the content of the HTML doc, usually shown in the browser window

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>...</title>
  </head>
  <body>
    ...
  </body>
</html>
```


Block and inline elements in <body>

9

- <body> contains some block elements.
- **Block elements** are displayed in rectangular blocks stacked from top to bottom
 - ▣ May contain text, inline elements and block elements
- **Inline elements** are displayed inline with the text flow
 - ▣ contain text and other inline elements.
 - ▣ Inline elements **cannot** contain block elements

```
<body>
  <h1>Structure of HTML</h1>
  <p>An HTML doc has a <em>tree structure</em>,
    where a parent element contains
    some child elements.</p>
</body>
```

Common block elements

10

- Common block elements:
 - `<p>` : paragraph
 - `<h1>`, `<h2>` ... `<h6>` : headings on different levels
 - `<blockquote>`
 - `<pre>` : preformatted content
 - Lists: ``, ``, ``, `<dl>`, `<dt>`, `<dd>`
 - Tables: `<table>`
 - Forms: `<form>`

Common inline elements

11

- Common inline elements:

- ▣ ``, `` : text emphasis
- ▣ `<abbr>` : abbreviation
- ▣ `<q>` : inline quotes
- ▣ `
` : line break
- ▣ `<code>` : inline computer code
- ▣ `<a>` : hyperlink
- ▣ `` : images
- ▣ `<object>`, `<iframe>`, `<video>`

Reminder: Semantic HTML

12

- You should use HTML according to the meaning of data.
 - ▣ Don't use HTML to adjust the presentation of data. Use CSS to format the data

Text blocks <h1>-<h6> <p>

13

- <body> contains some block elements

```
<body>
```

```
  <h1>Basic web markup and style</h1>
```

```
  <p>This chapter studies the basics of HTML and CSS</p>
```

```
  <h2>Structured text</h2>
```

```
  <p>Structure of HTML document. Common inline  
    and block elements for marking up text.</p>
```

```
  <h2>Text styles</h2>
```

```
  <p>Basic text styling in CSS. Style rules.</p>
```

```
</body>
```

Emphasis

14

- **** represents stress emphasis
 - ▣ Usually, is formatted as *italic*. You may customize the presentation by CSS.
 - ▣ In HTML 4.01, the <i> tag was used to render text in italics. However, this is not necessarily the case with HTML5. Style sheets can be used to format the text inside the <i> element.

```
<p>Compare these two sentences.</p>
```

```
<p>Call the doctor <em>now</em>!</p>
```

```
<p>Call the <em>doctor</em> now!</p>
```

Strong importance

15

- **** represents strong importance
 - ▣ Usually, is formatted as bold. You may customize the presentation by CSS.
 - ▣ has NO different meaning in HTML5.
 - ▣ But, the tag should be used as a LAST resort when no other tag is more appropriate.

```
<p><strong>Warning</strong>  
Guard dogs</p>
```



Inline quote <q> 인용

16

- Quote using inline elements <q>
 - ▣ May use CSS to specify suitable quotation marks

<p>Einstein said <q>Everything should be made as simple as possible, but not simpler</q>. Therefore, I think the midterm test should also be simple.</p>

Block quote <blockquote>

17

- The block element `<blockquote>` is used to quote a passage. The content should be block elements.

那这个cite会在哪里出现？

```
<blockquote cite="http://en.wikipedia.org/wiki/Css">  
<p>Cascading Style Sheets (CSS) is a style sheet  
language used to describe the presentation semantics (that  
is, the look and formatting) of a document</p>  
</blockquote>
```

Abbreviation <abbr>

18

- <abbr> marks up abbreviations with their long form.
- ▣ The long form (in title attribute) is usually shown as tooltip when the cursor is over the abbreviation.

```
<p>You can use <abbr title="structured query language">SQL</abbr> to query data from a relational <abbr title="Database Management System">DBMS</abbr>.</p>
```

Whitespace and line break

19

- In most cases, HTML treats spaces, tabs and carriage returns as the same **whitespace character**. In addition, multiple whitespace characters collapse to one.
- To enforce a line break within a paragraph, use the empty element **
** (only use this if the line break changes the meaning of the passage, e.g. as in a poem)
- To avoid whitespace collapsing, use the non-breaking space ** **

```
<p>You      cannot  
  add a line break  
like this. <br/>Use the br element to do so. </p>
```

You cannot add a line break like this.
Use the br element to do so.

Preformatted Text <pre>

20

- <pre> is a special block element that preserves spaces, tabs, and carriage return characters in content.
 - ▣ Similar to <p>, it requires inline content
 - ▣ Use it when the formatting has meaning. E.g. program source code
 - ▣ The first line break after <pre> is ignored.

```
<pre>
int sum=0;
for (int i=1; i<=10; i++)
    sum+=i; </pre>
```

Computer code <code>

21

- Use <code> to mark up inline computer code
行内代码

<p>To read student name and ID of all students,
use <code>SELECT name, studid from
student</code>.</p>

Attributes: class & id

22

- We often need to select one or more elements in a page from CSS and JavaScript
- The attribute **id** provides a unique identifier. Used to select a single element
- The attribute **class** lists which class(es) an element belong to
 - ▣ More than one element may belong to the same class
 - ▣ An element may belong to several classes
- You can also annotate elements with meaning using these attributes

Example

23

```
<p id="summary">  
  <code class="attr">id</code> and  
  <code class="attr">class</code> are  
  used to reference elements.  
</p>
```

```
<p id="first" class="important">..  
<p class="important revised">..  
<p class="revised">..  
<p id="last">..</p>
```

An element in several classes

24

- An element can also belong to more than one class

```
<p class="revised" >paragraph one</p>  
<p class="revised hilite" >paragraph two</p>  
<p class="hilite" >paragraph three</p>  
<p>paragraph four</p>
```

<!-- THIS IS INCORRECT -->

```
<p class="revised" class="hilite">paragraph two</p>
```


Adding structure to HTML content

25

Two elements are used to group HTML content

- `` – an inline element that can contain text and inline elements
- `<div>` – a block element that can contain text, inline elements, and block elements
- No other meaning is defined in HTML
 - ▣ Authors can assign their own meaning using the attribute `'id'` and `'class'`

Example of ``

26

- Use `` to group inline content, and add an id or class. Then you can format them with CSS or modify them in JavaScript.

```
<style type="text/css">
  span.typo { border-bottom: 1px dotted red; }
</style>

<p>HTML is <span class="typo">the</span>
predominant markup <span class="typo">language
</span> for web pages.</p>
```

HTML is the predominant markup language for web pages.

Grouping block elements with <div>

27

- Group several block elements with <div>
 - ▣ E.g. a heading and the few paragraphs after it form a section (may also use <section> in HTML5)

```
<body>
  <div class="section" id="section_html">
    <h2>HTML essential</h2>
    <p>The <em>structure</em> is important.</p>
    <p>another paragraph of this section...</p>
  </div>
  <div class="section" id="section_css">
    <h2>CSS essential</h2>
    <p>something about CSS ...</p>
  </div>
</body>
```

Use **class** to describe the kind of the group

Use **id** to identify a particular group

Links and images

28

- Hyperlink `<a>`
- Images
 - ▣ As content: inline image ``
 - ▣ For presentation only: `background-image`

Hyperlink <a>

29

```
<a href="http://www.w3.org">WWW Consortium</a>
```

- The *required* attribute **href** is a URL (absolute / relative)
- **<a>** is an inline element
 - ▣ The content of <a> can be text, image and some inline elements.
 - ▣ When the user clicks the link, the browser goes to the referred URL.
- Restrictions:
 - ▣ You *cannot* put block elements inside <a>. (But yes for HTML5)
 - ▣ You *cannot* put another <a> inside <a>.

Anchor and the id attribute

30

- An anchor can reference the element with the specific id on the page.

```
<a href="http://example.com/a.html#Overview"> Overview  
of HTML</a>
```

```
<!-- http://example.com/a.html -->  
<h2 id="Overview">Overview</h2>  
<p> ... </p>
```

Use of URL in HTML

31

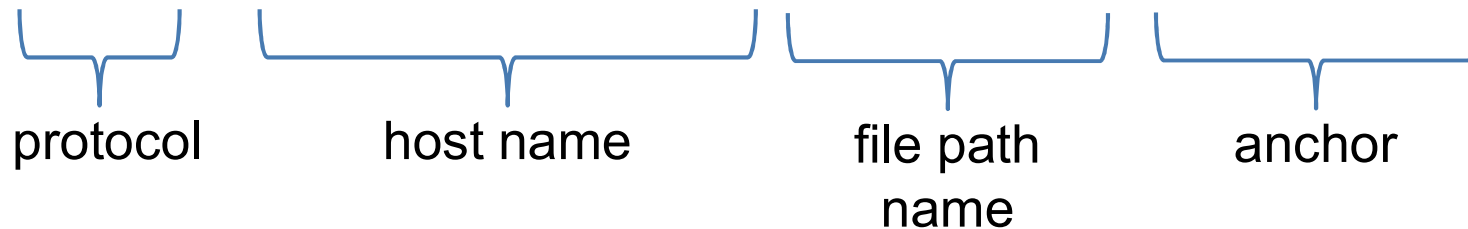
- An HTML doc refers to another web resource with a URL
 - ▣ Media displayed inside HTML content: ``, `<object>`, `<video>`
 - ▣ Background image specified in a style sheet
 - ▣ External style sheet and JavaScript library for the current page
 - ▣ A hyperlink `<a>`
- The URL can either be absolute or relative

Absolute URL

32

- An absolute URL must provide the protocol and host name
 - ▣ If absent, the file path name is usually assumed to be '/'. (The front page of the site)

`http://en.wikipedia.org/wiki/Xhtml#Overview`



Relative URL

33

- When a web page points to another page in the same website, we may use relative URL
- Relative URL omits the protocol, host name, and (maybe) a prefix of the file path name

```
<!-- this page is http://example.com/news/p1.html -->  
<a href="p2.html">Page 2</a>  
<a href="image/pic.png">a picture</a>  
<a href="#top">top of this page</a>  
<a href="/home.html">home</a>
```

```
http://example.com/news/p2.html  
http://example.com/news/image/pic.png  
http://example.com/news/p1.html#top  
http://example.com/home.html
```

Image

34

```

```

- is an empty inline element
- The image file is downloaded and displayed inline with other HTML content
- The *required* attribute **src** is a URL to the image data file
- The *required* attribute **alt** provides a description of the image.
 - ▣ Browsers show it before the image is ready to display
 - ▣ A screen reader may also read it out

Step 1. Check how images are arranged along text flow. Resize the window. 😊

Width and height of

35

```

```

- ❑ *Optional* attributes. The unit is pixel.
- ❑ Benefits: The browser can reserve enough space for the image. Layout of the page content will not change after downloading the image data
- ❑ You can also change the display size with CSS property **width** and **height**
- ❑ If the intrinsic size differs from the specified size, the browser resizes the image.

Common image formats

36

□ GIF and PNG

- Lossless compression, suitable for images with flat areas of plain color and text
- GIF: 256 (8bit) colors, animated GIF
- PNG: 48bit color, alpha transparency
- http://en.wikipedia.org/wiki/Portable_Network_Graphics

□ JPEG

- lossy compression suitable for images such as photographs



Lists

37

- A list of related items
- Three types:
 - ▣ Unordered list: `` ``
 - ▣ Ordered list: `` ``
 - ▣ Definition list: `<dl>` `<dd>` `<dt>`
- List styles

Unordered list

38

- Unordered list is a block element
- contains some (list item)
- Each may contain text, inline elements and block elements
- You can customize the bullets by CSS

```
<ul>  
  <li>HTML</li>  
  <li>CSS</li>  
  <li>Javascript</li>  
</ul>
```

- HTML
- CSS
- Javascript

Ordered list

39

- Ordered list is a block element
- contains some , similar to
- You can customize the numbering by CSS

```
<ol>  
  <li>Web standards</li>  
  <li>HTML</li>  
  <li>CSS</li>  
</ol>
```

```
1. Web standards  
2. HTML  
3. CSS
```

Nested lists

40

- `` can contain block elements, even another list.
- Notice how each level adds some indentation

```
<ol>
  <li>Web standards</li>
  <li>HTML
    <ul><li>syntax</li>
      <li>text</li><li>images</li>
    </ul>
  </li>
  <li>CSS
    <ul><li>selectors</li><li>properties</li></ul>
  </li>
</ol>
```

- 1. Web standards
- 2. HTML
 - syntax
 - text
 - images
- 3. CSS
 - selectors
 - properties

Definition List

41

- `<dl>` is a block element, which contains several items. Each item is composed of
 - ▣ a term `<dt>`
 - ▣ an explanation `<dd>` **define discription**

```
<dl>
  <dt>HTML</dt>
  <dd>the predominant markup ...</dd>
  <dt>CSS</dt>
  <dd>style sheet language ...</dd>
</dl>
```

HTML	the predominant markup language for web pages
CSS	style sheet language to describe presentation of web pages

Navigation menu

42

- Semantically, a navigation menu is a list of hyperlinks
 - ▣ We can use CSS to format the menu.

```
<p>List of lab in Chap 2</p>
<ul id="nav">
  <li><a href="lab2-1.html">Text markup</a></li>
  <li><a href="lab2-2.html">Text styles</a></li>
  <li><a href="lab2-3.html">Box model</a></li>
  <li><a href="lab2-4.html">Links & images</a></li>
  <li><a href="lab2-5.html">Lists</a></li>
</ul>
```

List of lab in Chap 2

- [Text markup](#)
- [Text styles](#)
- [Box model](#)
- [Links & images](#)
- [Lists](#)

Table

43

- HTML table is a collection of data arranged in rows and columns
 - ▣ Row centric: list table cells in rows
 - ▣ Merging cells: a cell can span multiple rows and columns
 - ▣ Additional info: caption
 - ▣ Additional structure: columns, row groups, column groups
- Table styles
 - ▣ Formatting the grid of cells
 - ▣ Formatting the internal of cells

Name	Test 1	Test 2
Peter	90	70
Mary	100	80

Basic syntax

44

- A `<table>` contains several rows `<tr>`, each row contains several cells. A cell is either a data cell `<td>` or heading cell `<th>`.

```
<table>
  <tr><th>Name</th><th>Course</th><th>Exam</th></tr>
  <tr><th>Peter</th><td>70</td><td>80</td></tr>
  <tr><th>Mary</th><td>80</td><td>85</td></tr>
  <tr><th>John</th><td>65</td><td>70</td></tr>
</table>
```

Name	Course	Exam
Peter	70	80
Mary	80	85
John	65	70

2D structure of tables

45

- HTML tables are used to represent two-dimensional data
 - ▣ A header cell `<th>` on the first row describes the data in a column
 - ▣ A header cell `<th>` on the first column describes the data in a row
 - ▣ May be specified with the attribute 'scope'

Name	Course	Exam
Peter	70	80
Mary	80	85
John	65	70

Table head, body and foot

46

- Rows in a long table may be grouped
 - ▣ `<thead>` : table head contains headers
 - ▣ `<tbody>` : table body for the main data
 - ▣ `<tfoot>` : table footer contains summary info
- Table head is shown before table body. Table foot is shown after.
 - ▣ Note: HTML4 and XHTML requires putting `<tfoot>` before `<tbody>` in HTML source
- There can be more than one `<tbody>`

Example

47

Name	Course	Exam
Peter	70	80
Mary	80	85
John	65	70
Average	71.7	78.3

```
<table>
  <thead>
    <tr><th>Name</th><th>Course</th><th>Exam</th></tr>
  </thead>
  <tbody>
    <tr><th>Peter</th><td>70</td><td>80</td></tr>
    <tr><th>Mary</th><td>80</td><td>85</td></tr>
    <tr><th>John</th><td>65</td><td>70</td></tr>
  </tbody>
  <tfoot>
    <tr><th>Average</th><td>71.7</td><td>78.3</td></tr>
  </tfoot>
</table>
```

Merging table cells

48

- A table cell can span multiple rows and columns using the attributes **rowspan** and **colspan**

Student		Mark
p1030001	Ann	80
p1030002	Boris	
p1030003	Cecilia	76
p1030004	Debra	

<th colspan='2'>

<td rowspan='2'>

Example

49

```
<table>
  <thead><tr><th colspan='2'>Student</th>
    <th>Mark</th></tr></thead>
  <tbody>
    <tr><td>p1030001</td><td>Ann</td>
      <td rowspan='2'>80</td></tr>
    <tr><td>p1030002</td><td>Boris</td></tr>
    <tr><td>p1030003</td><td>Cecilia</td>
      <td rowspan='2'>76</td></tr>
    <tr><td>p1030004</td><td>Debra</td></tr>
  </tbody>
</table>
```

Student		Mark
p1030001	Ann	80
p1030002	Boris	
p1030003	Cecilia	76
p1030004	Debra	

Notice that cell merging result in less table cells in some rows.

Table caption

50

- Add a caption to a table
 - ▣ To set the location of the caption, `table { caption-side: bottom }`

```
<table>  
  <caption>Mark sheet for COMP113</caption>  
  <thead> ... </thead>  
  <tbody> ... </tbody>  
  <tfoot> ... </tfoot>  
</table>
```

Mark sheet for COMP113

Name	Course	Exam	Grade
Peter	70	80	B+
Mary	80	85	A-
John	65	70	C
Average	71.7	78.3	B

Optional tags

51

- End tags for `<thead>`, `<tbody>`, `<tfoot>`, `<tr>`, `<th>` and `<td>` are generally **optional**

```
<table>
  <caption>Mark sheet for COMP113</caption>
  <thead>
    <tr><th scope='col'>Name<th scope='col'>Course
      <th scope='col'>Exam<th scope='col'>Grade
  <tbody>
    <tr><th scope='row'>Peter<td>70<td>80<td>B+
    <tr><th scope='row'>Mary<td>80<td>85<td>A-
    <tr><th scope='row'>John<td>65<td>70<td>C
  <tfoot>
    <tr><th scope='row'>Average<td>71.7<td>78.3<td>B
</table>
```

Row groups

52

- You can group rows into more than one groups using `<tbody>`

```
<table id='T1'>
  <thead> .. </thead>
  <tbody id='y1'> <tr>..<tr>..<tr>.. </tbody>
  <tbody id='y2'> <tr>..<tr>..<tr>.. </tbody>
  <tbody id='y3'> <tr>.. <tr>.. </tbody>
  <tbody id='y4'> <tr>.. </tbody>
</table>
```

Code	Course
COMP112	Programming I
COMP113	Web Technologies
COMP122	Data Structures and Algorithms
COMP212	Programming II
COMP221	Object Oriented Technologies
COMP222	Internet Programming I
COMP312	Internet Programming II
COMP321	Information System Implementation
COMP491	Final Year Project

Columns

53

- Assign an id or class to columns using `<col>` and `<colgroup>` elements
- The number of columns specified should agree with the table cells. (HTML validation can verify this)
- You can select the columns and apply CSS styles

1	2	3
1	2	3
1	2	3

1	2	3
1	2	3
1	2	3

```
<table>
  <col/><col class='mid' /><col/>
  <tbody>
    <tr><td>1<td>2<td>3</tr>...
  </tbody>
</table>
```

```
table col.mid {
  width: 5em;
  background-color: lightgreen;
  border: 2px solid red;
}
```

Form

54

- ☐ Form and form controls
- ☐ Adding style to forms
- ☐ Form validation

Forms

55

- Forms enable users to enter info for processing
 - A form contains various form controls to collect data
 - Form data may be processed by
 - Client-side JavaScript code, and/or
 - Server-side script at an URL

A screenshot of the Gmail login interface. At the top, it says "Sign in to Gmail with your" followed by the "Google Account" logo. Below this are two input fields: "Username:" and "Password:". Under the password field is a checkbox labeled "Stay signed in". A "Sign in" button is positioned below the checkbox. At the bottom, there is a blue hyperlink that reads "Can't access your account?".

Login form of Gmail

Form controls

56

Three main kinds of form controls

Text entry

Text box

text box

Text area

Multiline text entry

Action

Selection

Radio button

choice 1 ☒ choice 2 ☐ choice 3 ☐

Check box

choice 1 ☒ choice 2 ☒ choice 3 ☐

Pull-down menu

Mon

Scrolling list

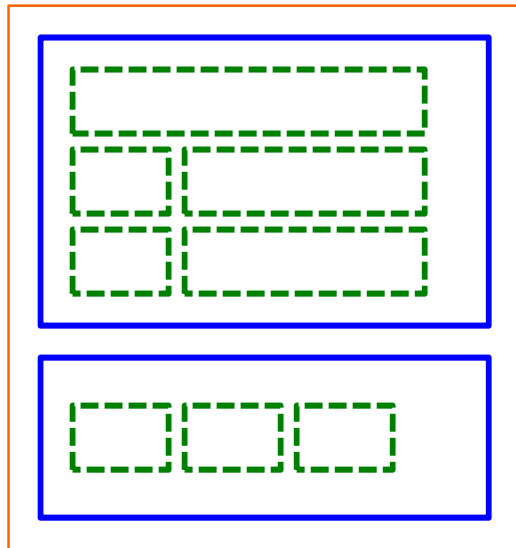
Mon
Tue
Wed
Thu

The form element

57

- **<form>** is a block element with three attributes
 - **action** (required) – URL of server side script to receive the form data when the form is submitted
 - **method** (optional) – HTTP method to send the data
 - get (default), or post
 - **enctype** (optional) – (encoding type) how form data is encoded as string
 - application/x-www-form-urlencoded (default) is suitable for small amount of data
 - multipart/form-data is suitable for file upload. Need method="post"

- Usually we organize content in a `<form>` in blocks



`<form>`



Block elements, e.g. `<p>`, `<div>`,
`<table>`, `<fieldset>`



Text and inline elements, e.g. `<a>`,
``, `<input>`, `<select>`,
`<textarea>`

Form data

59

- A form collects data from form controls as name-value pairs
 - E.g. user=peter, passwd=123
 - Each form control must have the attribute **name**
 - attributes **id** and **name** have different meaning
 - Users interact with a control to set its value
 - Controls specify their initial values in different ways: attributes **value**, **checked**, **selected** and element content

Usage of common form controls

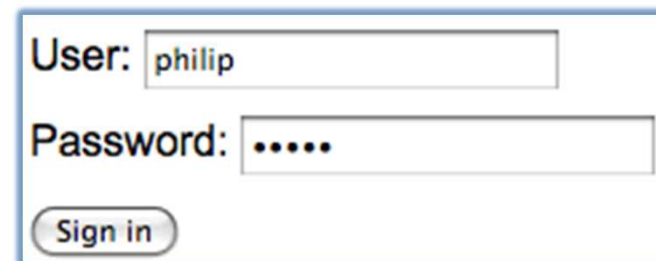
60

Task	UI	HTML element
Simple text entry	Text box	<code><input type="text" /></code>
Password entry	Text box	<code><input type="password" /></code>
Multiline text-entry	Text area	<code><textarea></code>
Yes / No	Checkbox	<code><input type="checkbox" /></code>
Choose one from a group	Radio button	<code><input type="radio" /></code>
	Pull-down menu	<code><select></code>
Choose many from a group	Checkbox group	<code><input type="checkbox" /></code>
	Scrolling list	<code><select multiple="multiple"></code>
Submit form	Submit button	<code><input type="submit" /></code>
Other action	General button	<code><input type="button"/></code>
		<code><button type="button"></code>

Simple text entry

61

- ❑ Use `<input type="text" />` for simple text entry.
- ❑ Optional attributes:
 - ❑ `size` : length in number of characters
 - ❑ `maxlength` : max number of char the user can enter
 - ❑ `value` : initial value
 - ❑ `placeholder`: display when the text field is empty
- ❑ Use `<input type="password" />` to mask the input

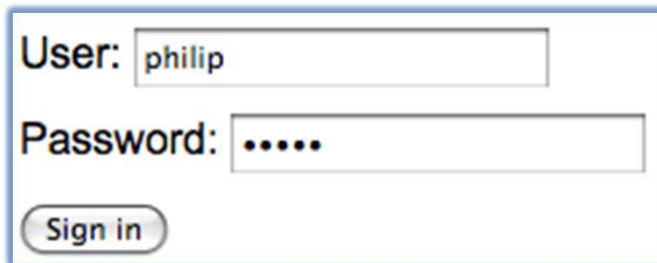


A login form with two input fields and a button. The first field is labeled "User:" and contains the text "philip". The second field is labeled "Password:" and contains five dots. Below the fields is a button labeled "Sign in".

Example

62

```
<form action="login.jsp" method="get">  
  <p>User: <input type="text" name="user" /> </p>  
  <p>Password: <input type="password" name="passwd" /> </p>  
  <p><input type="submit" value="Sign in"/> </p>  
</form>
```



User:

Password:

This form will send user=philip, passwd=12345

Label

63

- `<label>` links a text label to a control
 - When you click the label, the form control gains the input focus
 - Associate the label and the form control by the attribute `for` and `id`.

```
<p><label for="fname">Name</label>
<input type="text" name="fname" id="fname" />
</p>
<p><label for="fstudid">Student ID</label>
<input type="text" name="fstudid" id="fstudid" />
</p>
```



Note: the 'name' attribute is used for form submission only. It is not required to have the same value as the 'id' attribute.

Label, a shorthand

64

- If the `<label>` and its associated form control are near, you can put the control inside the `<label>`
 - ▣ Don't need to assign id to the control

```
<p>
  <label>Name
    <input type="text" name="fname" /> </label>
</p>
<p>
  <label>Student ID
    <input type="text" name="fstudid" /> </label>
</p>
```



Multiline text entry

65

- Use `<textarea>` to enter more than one lines of text
 - ▣ `<textarea name="mesg" rows="3" cols="40">initial value</textarea>`

```
<p><textarea name="mesg" rows="3" cols="40"></textarea> </p>
<p><input type="submit" name="action" value="Send" />
<input type="submit" name="action" value="Save as draft" />
<input type="submit" name="action" value="Discard" /></p>
```

New message

How are you recently?

Philip

Send

Save as draft

Discard

*This form will send
mesg=How+are+you+recently%3F%0D%0A%0D%0APhilip
action=Send*

Hidden field

66

- No user interface shown. Send data when the form is submitted
 - ▣ `<input type="hidden" name="customerid" value="123"/>`
- Useful to carry form data from one page to another

File upload

67

- A control that allows the user to select a file to upload
 - ▣ `<input type="file" name="uploadfile"/>`
 - ▣ The file is uploaded when the user submits the form
- Add attribute `enctype="multipart/form-data"` to `<form>`
 - ▣ The default encoding type “application/x-www-form-urlencoded” is not suitable for large amount of data transfer

New type of <input>

68

```
<p>Mobile: <input type="tel" name="mobile" /> </p>  
<p>Email: <input type="email" name="email" /> </p>
```

- HTML5 defines some new input types, but not all of them are implemented.
 - ▣ email, url, tel, number, color, date, ...
- Firefox support: <http://hacks.mozilla.org/2010/11/firefox-4-html5-forms/>
- Opera support: <http://dev.opera.com/articles/view/new-form-features-in-html5/>
- Current state: <http://wufoo.com/html5/>

Checkbox

69

- Use a checkbox to ask a yes/no question.
 - ▣ `<input type="checkbox" name="agree" />`
 - ▣ To check it by default, add the boolean attribute `checked`
 - ▣ If checked, send `name=on`, or `name=value` if attribute `value` is set
 - ▣ If not checked, don't send anything.

Do you agree?

```
<input type="checkbox" name="agree" value="yes" checked="checked" />
```

Do you agree? ☒

This form will send agree=yes

Radio button

70

- Use radio buttons to select one choice from a group
 - ▣ `<input type="radio" name="color" value="red"/>`
 - ▣ Add the boolean attribute `checked` to preselect a choice
 - ▣ If checked, send name=value

Choose a color:

Red `<input type="radio" name="color" value="red" checked"/>`

Green `<input type="radio" name="color" value="green"/>`

Blue `<input type="radio" name="color" value="blue"/>`

Choose a color: Red ☒ Green ☐ Blue ☐

This form will send color=red

Checkbox group

71

- A checkbox group allows a user to select more than one item

What pets do you own?

<input type="checkbox" name="pets" value="dog" />Dog

<input type="checkbox" name="pets" value="cat" />Cat

<input type="checkbox" name="pets" value="bird" />Bird

<input type="checkbox" name="pets" value="fish" />Fish

What pets do you own?

☒ Dog ☐ Cat

☐ Bird ☒ Fish

This form will send pets=dog, pets=fish

Adding labels to radio buttons

72

- When you click the label associated with a radio button, the form control is checked
 - ▣ Improve accessibility: make your radio button easier to check.
 - ▣ also work for checkbox

Choose a color: Red ☒ Green ☐ Blue ☐

```
<label>Red
  <input type="radio" name="color" value="red" checked="checked"/>
</label>
<label>Green
  <input type="radio" name="color" value="green"/></label>
<label>Blue
  <input type="radio" name="color" value="blue"/></label>
```


Pull-down menu

73

- A compact control to select one option from many
 - ▣ `<select name="month" > <option> ... </select>`
 - ▣ Use the boolean attribute `selected` to preselect an option
 - ▣ Send `name=content` of the selected option, or the option's attribute `value` if present.

```
<select name="month">  
  <option selected="selected">Month:</option>  
  <option>Jan</option>  
  <option>Feb</option> <option>Mar</option> ...  
</select>
```

Birthday: Oct 1

Month:
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

Example

74

```
<select name="month">  
  <option>Jan</option>  
  <option>Feb</option>  
  <option>Mar</option> ...  
</select>
```

*This form will send
month=Jan, if the user
selects 'Jan'*

```
<select name="month">  
  <option value="1">Jan</option>  
  <option value="2">Feb</option>  
  <option value="3">Mar</option> ...  
</select>
```

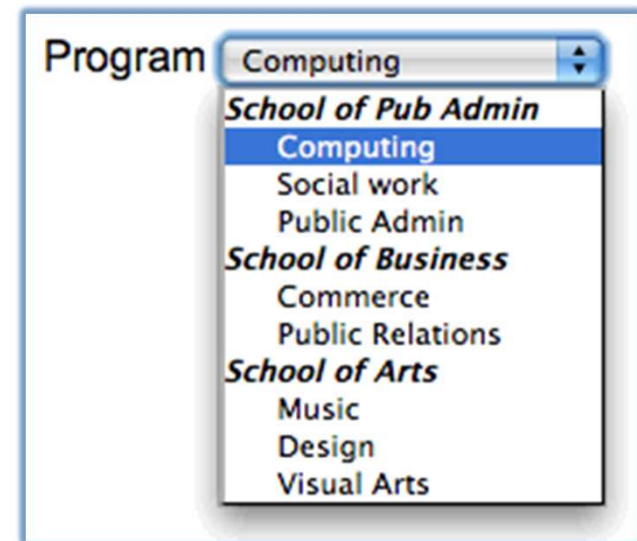
*This form will send
month=1. The attribute
value is useful when
the option content is
too verbose.*

Option group

75

- To improve accessibility, you can break a large number of options into groups.

```
<label for="fprog">Program</label>
<select name="fprog" id="fprog">
  <optgroup label="School of Pub Admin">
    <option>Computing</option>
    <option>Social work</option>
    <option>Public Admin</option>
  </optgroup>
  <optgroup label="School of Business">
    <option>Commerce</option>
    <option>Public Relations</option>
  </optgroup> ...
</select>
```



Optional tags in option groups

76

- The end tags of both `<option>` and `<optgroup>` are optional

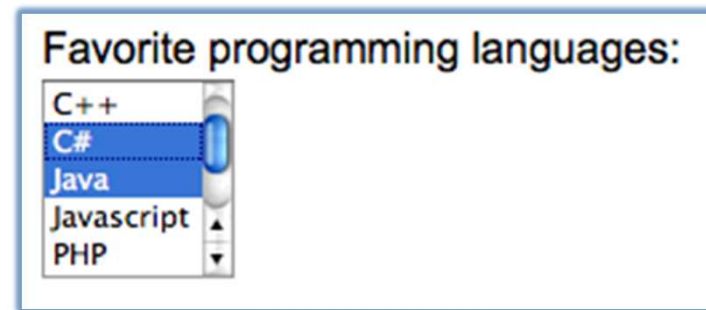
```
<label for="fprog">Program</label>
<select name="fprog" id="fprog">
  <optgroup label="School of Pub Admin">
    <option>Computing</option>
    <option>Social work</option>
    <option>Public Admin</option>
  </optgroup>
  <optgroup label="School of Business">
    <option>Commerce</option>
    <option>Public Relations</option>
  </optgroup> ...
</select>
```

Scrolling list

77

- A compact control to select *multiple options* from many
 - Boolean attribute **multiple**
 - **size** is the number of options shown

```
<select name="proglang"  
  multiple="multiple" size="5">  
  <option>C++</option>  
  <option>C#</option> ...  
</select>
```



Summary of selection controls

78

	A few options. You want to display all.	Too many options. You want to hide some.
Select one option only	Radio button <input type="radio" />	Pull-down menu <select >
Select multiple options	Check box group <input type="checkbox" />	Scrolling list <select multiple size="5">

```
Red <input type="radio" name="color"
value="red" checked />
Green <input type="radio" name="color"
value="green" />
Blue <input type="radio" name="color"
value="blue" />
```

```
<select name="month">
  <option value="1" selected >
    Jan</option>
  <option value="2">Feb</option>
  <option value="3">Mar</option> ...
</select>
```

Field set

79

- `<fieldset>` is a block element that groups related form controls
 - ▣ `<legend>` provides description of the group
 - ▣ You may style the border of the fieldset and text style of legend

```
<fieldset>
  <legend>Student info</legend>
  Name <input ... /> ...
  Student ID <input .../> ...
  Program <select> ...
</fieldset>
...
```

Student info

Name

Student ID

Program

Interests

Select activities you are interested in

Drama ☒ Basketball ☒ Photography ☐

submit

Buttons

80

- HTML defines three types of buttons
 - ▣ Submit buttons: submits a form
 - ▣ Reset buttons: resets all controls to their initial values
 - ▣ Push buttons: no default behavior. Web authors associate client-side scripts to the buttons.
- Two ways to make a button `type=submit / reset / button`
 - ▣ `<input type="submit" value="content" />` uses the attribute value as button content
 - ▣ `<button type="submit"> content </button>` allows images inside button

Submit button

81

- The user presses this button to submit the form
 - ▣ `<input type="submit" name="action" value="Send"/>`
 - ▣ When pressed, the form collects data from form controls and send it to server
 - ▣ Pressed button also sends name=value
 - ▣ Use several submit buttons with different values to distinguish user intention

Note: the 'value' is shown in the button

New message

How are you recently?
Philip

Send Save as draft Discard

Example

82

New message

How are you recently?
Philip

```
<form action="#">
<fieldset>
  <legend>New message</legend>
  <p><textarea name="mesg" rows="3" cols="30"></textarea></p>
  <p><input type="submit" name="action" value="Send" />
    <input type="submit" name="action" value="Save as draft" />
    <input type="submit" name="action" value="Discard" /></p>
</fieldset>
</form> ...
```

Reset button and Push button

83

- When the user presses the reset button, all form controls return to their initial values
 - ▣ `<input type="reset" value="start over"/>`
 - ▣ The form is not submitted
 - ▣ Not used much in modern web pages
- You can define a push button that triggers a JavaScript function when clicked
 - ▣ `<input type="button" value="check spelling" onclick="alert('to be implemented'); "/>`

<button>

84

- The `<button>` element can use both image and text as content

- `<button type="submit" name="action" value="save">Save</button>` submits form and send action=save.
- `<button type="button">Save</button>` triggers JavaScript script associate to the click event of the button.

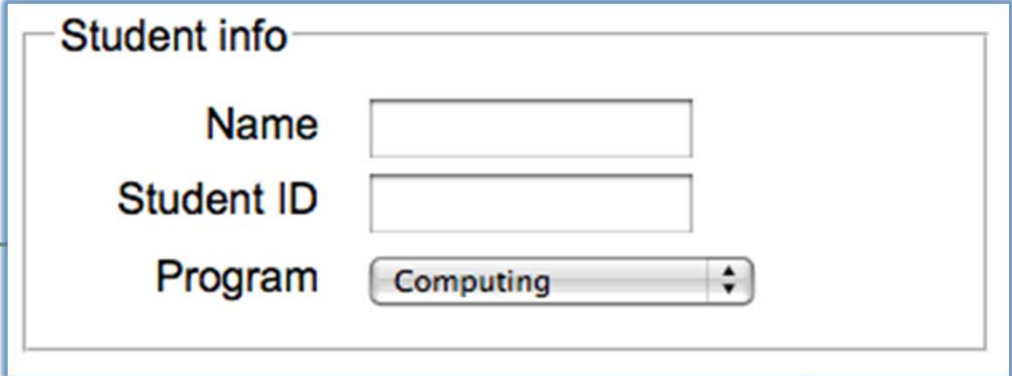


```
<button type="button">
Ok</button>
<button type="button">
Cancel</button>
```

Formatting form in a grid

85

- A common method to align form labels and controls is `<table>`
 - ▣ Easy to use, but makes the HTML code confusing
 - ▣ Pure CSS method in next chapter



Student info

Name

Student ID

Program

```
<form action="#">
```

```
<table>
```

```
<tr><td><label ...></td><td><input ... /></td></tr>
```

```
<tr><td><label ...></td><td><input ... /></td></tr>
```

```
<tr><td><label ...></td><td><select... ></td></tr>
```

```
</table>
```

```
</form>
```

Form validation

86

- HTML5 and CSS3 define some new features for form validation
 - ▣ **required** attribute – the user must enter non-empty value
 - ▣ Input elements of type **email** and **url** validate whether the entry is an email address / URL
 - ▣ **pattern** attribute uses JavaScript regular expression to specify criteria for valid entry
 - ▣ CSS pseudo-class **:valid** and **:invalid** allows different styles for input fields with valid and invalid entries.

Further reading

87

- More attributes for form controls
 - **tabindex** determines the order that the input focus shifts from control to control when the user presses the Tab key
 - **disabled** disables a control. The form will not send its value when submitted
 - **readonly** makes a control read-only. The form will send its value when submitted.

Metadata and embedded content

88

- Metadata: data about the page
 - ▣ Metadata in <head>
 - ▣ Links: CSS, JavaScript, RSS, icons, ...
 - ▣ Schema.org
- Embedded content
 - ▣ <iframe> - integrates external HTML content
 - ▣ <object> - embeds objects that are handled by plugins
 - ▣ <video> - new element in HTML5 for embedding video

Functions of <head>

89

- provide data about the page
- link to external resources (which augment the current doc)
- link to other documents

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta ... />
    <link ... />
    <script ...> </script>
  </head>
  <body> ... </body>
</html>
```

Elements in <head>

90

These elements are usually not rendered in the browser window

- `<title>` : usually appears in title bar of browser window
- `<meta>` : metadata, information about the html page
- `<link>` : link to external resources and other docs
- `<base>` : base address of URLs in the html page
- `<style>` : embedded style sheet
- `<script>` : add behavior. Usually JavaScript
 - ▣ Embedded code
 - ▣ Link to external library

Metadata <meta>

91

- Metadata, (data about data) describes a web page, e.g. keywords, description, author.
- Also affects action of search engine crawlers
 - ▣ robot exclusion protocol: <http://www.robotstxt.org/>

```
<meta name="keywords" content="html, css, tutorial" />
<meta name="description" content="a tutorial of web design using html and
css, prepared for IPM" />
<meta name="author" content="Philip" />

<!-- instructs search engines not to index this page, but to follow links from
the page. -->
<meta name="robots" content="noindex, follow" />
```

Character encoding in <meta>

92

- Web authors can specify the character encoding of an HTML doc with <meta>

```
<meta charset="UTF-8" />
```

```
<!-- HTML4 and XHTML version -->  
<meta http-equiv="Content-Type"  
      content="text/html; charset=UTF-8" />
```

External resources and documents

93

- A web browser usually needs to load some external resources to display an HTML doc
 - ▣ E.g. CSS style sheet, Favicon
- In addition, an HTML doc may be related to some other doc
 - ▣ E.g. several docs in a sequence, news feed
- The `<link>` element is used in both cases

<link>

94

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

- The <link> element defines a link to an external resource or another document
 - **rel** – its relationship with the current doc
 - **type** – its MIME type. Assume "text/css" for rel=stylesheet
 - **href** – URL of the resource / document

Common external resources

95

- Common external resources:
 - ▣ `rel="stylesheet"` : external CSS style sheet
 - ▣ `rel="shortcut icon"` : an icon used in address bar and bookmark

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

```
<link rel="shortcut icon" href="/favicon.ico" />
```

Links to other doc

96

- The `<link>` element can also link to another document that has some relationship with the current one.
 - ▣ In a sequence of documents, `rel="prev"`, `rel="next"`
 - ▣ `rel="alternate"` : an alternative version of the current document. Usually used for linking to a news feed in RSS or ATOM format.
 - ▣ Ref. <http://reference.sitepoint.com/html/link/rel>

```
<link rel="alternate" type="application/rss+xml"
href="rss.xml"/>
```


<script>

97

- The <script> element loads an external script file and executes it on the current doc
 - ▣ **type** – MIME type of script. Default is text/javascript
 - ▣ **src** – URL of the script file

```
<script type="text/javascript" src="menu.js" ></script >
```

- The <script> element can also embed script in the HTML doc. And it can also appear in <body>.

```
<script>  
  window.onload = function () { alert('done'); };  
</script >
```

Schema.org

98

- A common vocabulary for metadata markup defined by Google, Yahoo! and Bing
 - ▣ Enables rich snippets in Google
 - ▣ Example adopted from <http://schema.org>

```
<div itemscope itemtype="http://schema.org/Movie">  
  <h1 itemprop="name">Avatar</h1>  
  <span>Director:  
    <span itemprop="director">James Cameron</span>  
    (born August 16, 1954)</span>  
  <span itemprop="genre">Science fiction</span>  
  <a href="avatar-trailer.html" itemprop="trailer">Trailer</a>  
</div>
```

Inline frame <iframe>

99

- An inline frame integrates external HTML content into a web page
 - ▣ the iframe works like a small browser window
 - ▣ Displayed in an inline box with the given dimension
 - ▣ Common usage: embed another HTML page from the same site, a YouTube movie, Facebook like button, Google maps

```
<iframe src="http://www.ipm.edu.mo/"  
width="600" height="300" > </iframe>
```

Objects

100

- The `<object>` element embeds an object of any MIME `type` in an HTML doc
 - ▣ Content is displayed in an inline box. You can specify `width` and `height`.
 - ▣ A web browser usually runs *external programs* (known as *plugins*) to render the object.
 - ▣ Common usage: Flash animation, Silverlight animation, Windows media
 - ▣ To check installed plugins in Firefox: <about:plugins>
 - ▣ The older `<embed>` has similar functions

<object> basic syntax

101

```
<object type="application/pdf" data="test.pdf"  
      width="600" height="300">  
</object >
```

- The <object> element downloads and displays an external data file of certain MIME type
 - ▣ type – MIME type of the object
 - ▣ data – URL of the object data
 - ▣ width and height – dimension of the object

<object> general syntax

102

```
<object type="video/quicktime" data="test.mov"
width="320" height="240">
  <param name="src" value="test.mov" />
  <param name="controller" value="true" />
  Alternative content
</object >
```

- One or more <param> element inside <object> provides parameters for plugins.
 - ▣ Some browsers require to pass the data file URL as a <param>
- If the browser cannot display the object
 - ▣ it shows the **alternative content** inside <object>
 - ▣ it does not download the data file

Flash movie in an <object>

103

```
<object type="application/x-shockwave-flash"  
  data="pacman.swf" width="300" height="400">  
  <param name="movie" value="pacman.swf"/>  
  Cannot display the flash movie  
</object>
```

- Modern browsers can embed flash movie in the standard <object> element.
 - ▣ Compare this with the old embed code used by YouTube.

You may notice that the URL of the flash movie data file is specified twice. Firefox requires the data attribute, whereas IE requires the name param. The reason of such different implementation stems from ambiguity of the HTML4 spec on the object element.

Web video

104

- No common video format across different OS
 - ▣ E.g. Windows uses Windows Media, MacOS X uses Quicktime
- But the Flash player is wide-spread in desktop computers and it can play the Flash Video format (*.flv)
- Flash becomes a default online video format in many sites
 - ▣ But mobile devices have incomplete supports for Flash video...
- HTML5 introduces the `<video>` element
 - ▣ Ideally with a default video format supported by all browsers, but unfortunately no consensus at the moment
 - ▣ Possible to provide multiple versions of different video format

<video>

105

```
<video src="plant.ogv" width="300" height="400" controls>  
</video>
```

- the `<video>` element displays a video
 - ▣ The attribute `src` is an URL to the video file
 - ▣ Dimension in pixel: `width` and `height`
 - ▣ The boolean attribute `controls` indicates whether to show the default controls or not
 - ▣ The attribute `poster` is an URL to an image file. This image is displayed before playing the video.

No universal video format on the Web

106

- To play a video file, the browser has to be able to handle three aspects of the file
 - ▣ Container format. E.g. MPEG4 (.mp4), Ogg (.ogv), WebM (.webm)
 - ▣ Video codec. E.g. H.264, Theora, VP8
 - ▣ Audio codec. E.g. mp3, aac, Vorbis
- No single combination is supported in all platforms

	Firefox	Chrome	Safari on iPad	Internet Explorer
.ogv (Ogg+Theora+Vorbis)	Y	Y	N	N
.mp4 (MPEG4+H.264+aac)	Depends	Y, but not in the future	Y	Y
.webm (WebM+VP8+Vorbis)	Y	Y	N	N

Multiple video sources

107

```
<video controls width="300" height="400">
  <source src="plant.mp4"
    type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"' />
  <source src="plant.webm" type='video/webm; codecs="vp8, vorbis"' />
  <source src="plant.ogv" type='video/ogg; codecs="theora, vorbis"' />
  <p>Show this fallback message if none of the sources are playable</p>
</video>
```

- ❑ To allow users on different platforms to see your video, you have to encode it in multiple formats
- ❑ Specify a list of video files of different formats in `<source>` elements
- ❑ The browser checks the `type` attribute of each source element and try to find a format that is playable

Further readings

108

- About HTML5 video. Read the background of the formats
 - ▣ <http://www.html5rocks.com/en/tutorials/video/basics/>
 - ▣ http://en.wikipedia.org/wiki/HTML5_video
 - ▣ <http://diveintohtml5.org/video.html>

HTML5 Semantic Elements

109

- HTML5 offers new semantic elements to define different parts of a web page:

<code><article></code>	<code><figure></code>	<code><mark></code>	<code><time></code>
<code><aside></code>	<code><footer></code>	<code><nav></code>	
<code><details></code>	<code><header></code>	<code><section></code>	
<code><figcaption></code>	<code><main></code>	<code><summary></code>	



- See here:
 - https://www.w3schools.com/html/html5_semantic_elements.asp

HTML5 Custom Elements

110

- Custom Elements is a capability for creating your own custom HTML elements.
- Use JavaScript to code their behavior, and use CSS to style them.
- See:
 - ▣ https://developer.mozilla.org/en-US/docs/Web/Web_Components/Custom_Elements

Syntax and Grammar of HTML

111

- The HTML5 specification defines the HTML language and how to use it correctly
- Conforming HTML documents (valid documents)
 - No syntax errors
 - No grammatical errors: Restrictions on content models and on attribute values

HTML syntax

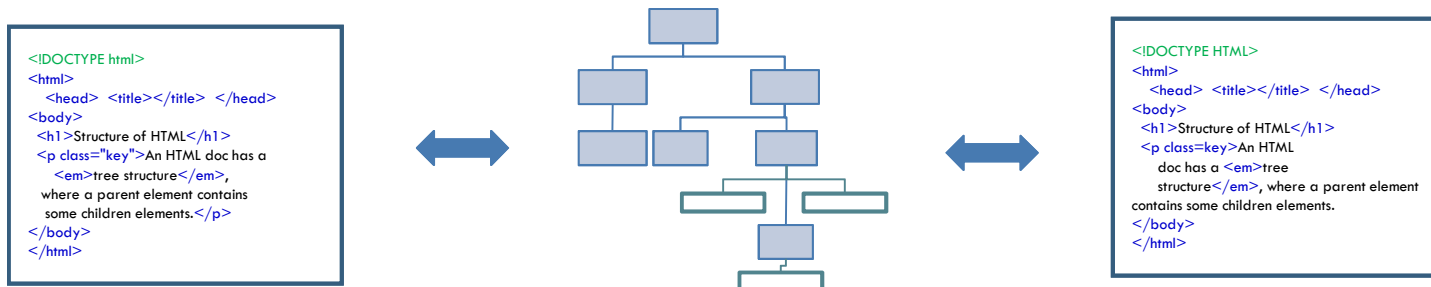
112

- HTML5 syntax is mostly compatible with XHTML and HTML4
 - ▣ Possible to write an HTML document as XML
 - ▣ Several shorthand convention from HTML4
 - Optional start tags and end tags
 - Unquoted attributes
 - Boolean attributes
 - Void (self-closing)
- The HTML5 spec also defines how to handle syntax error consistently

HTML doc ↔ DOM tree

113

- The HTML parser in a web browser converts an HTML doc of correct syntax to a DOM tree
 - ▣ CSS style rules describe style properties of each element in the tree
 - ▣ JavaScript codes access and modify the tree
- Small syntactical changes of HTML docs result in no change of the DOM tree
 - ▣ For the browsers, they are the same doc

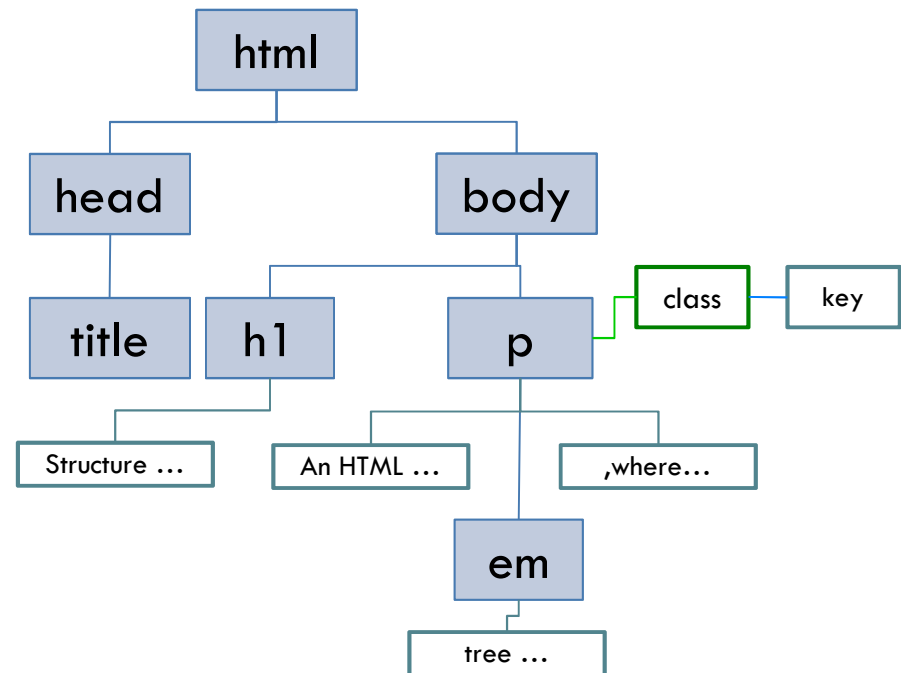


Example

114

- The DOM tree is a tree of elements and text

```
<!DOCTYPE html>
<html>
  <head> <title></title> </head>
<body>
  <h1>Structure of HTML</h1>
  <p class="key">An HTML doc has a
    <em>tree structure</em>,
    where a parent element contains
    some children elements.</p>
</body>
</html>
```



DOCTYPE declaration and root element

115

- A HTML5 document starts with a **DOCTYPE declaration**
 - ▣ Required for legacy reasons. When omitted, browsers tend to use a different rendering mode (commonly known as quirk mode) that is incompatible with some specifications.
 - ▣ Case-insensitive
- This is followed by a single root element, **<html>**

```
<!DOCTYPE html>
<html>
...
</html>
```

HTML elements

116

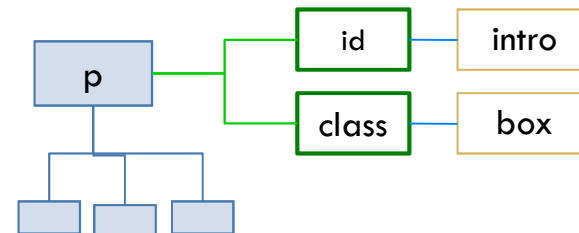
- Several kinds of elements:
 - ▣ **Void elements** can't have any content (e.g. ``)
 - ▣ **Normal elements** may contain text and other elements (e.g. `<p>`)
 - The characters '`<`' and '`&`' have special meaning and should be escaped as `<` and `&`;
 - ▣ **Raw text elements** (`<style>` and `<script>`) can have text. (don't need to escape '`<`' in most cases)
 - ▣ Others ...
- Element names in tags are case-insensitive

Syntax of elements, 1

117

```
<p id='intro' class="box">...content...</p>
```

- A normal element has a start tag and an end tag
 - ▣ Element type (tag name) appears in both tags
- Its content is between the start tag and end tag
 - ▣ The content may be empty, text, some elements, or mixed content.
- Attributes of the element are included in the start tag
 - ▣ Value of an attribute may be quoted
 - ▣ Attributes are separated by space



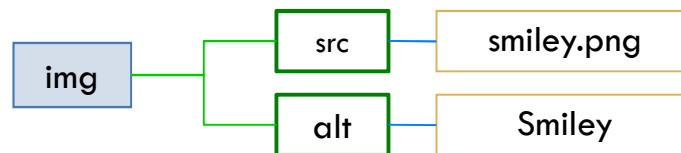
Syntax of elements, 2

118

```

```

- A void element has no end tag.
 - ▣ The character '/' is optional in HTML5, but required in XML.



```

```

More syntax rules, 1

119

- Non-void elements must be closed

```
<p>It <em>is important to close normal elements</p>
```

- Elements must nest properly

```
<p>Order of nesting is <em>also important!</p></em>
```

- Tag names are case-insensitive in HTML5, but case-sensitive in XHTML

```
<P>Possible, but <em>not</EM> encouraged. </P>
```

More syntax rules, 2

120

- An attribute can be specified only once in an element

`<p class="important" class="revised">` Write the two classes in a single attribute! `</p>`

- Attributes can only appear in start tags

`<p>` Don't put attributes in end tags. `</p class="error">`

- Attribute names are case-insensitive in HTML5, but case-sensitive in XHTML

`<p ID="oh">` Possible, but `<em cLASs="badhabit">` not `` encouraged. `</p>`

More syntax rules, 3

121

- ❑ In XHTML, you must quote attribute values
- ❑ In HTML5, you may omit quotes if the attribute value is a simple word.

```
<p class=important>Ok!</p>
```

```
<p class=important revised>Not ok!</p>
```

- ❑ Use HTML entities to write quote character " and ' in attribute values

```
<p id='John&apos;intro'>This is John's introduction</p>
```

More syntax rules, 4

122

- Boolean attributes represents true / false value
- True value represented by presence of the attribute

```
<input type="checkbox" checked name="A" />
```

```
<input type="checkbox" checked="checked" name="A" />
```

```
<input type="checkbox" checked="" name="A" />
```

- False value represented by absence of the attribute.

```
<input type="checkbox" name="A" />
```

Predefined HTML entities

123

- Five characters have special meaning in HTML5. When they appear in text content or attribute value, replace them with **HTML entities**.

- ▣ also defined in XML
- ▣ HTML5 relaxes the rules. You can write `>` `"` `'` & in most cases.

Symbol	Explanation	Entities
<	Less than	<
>	Greater than	>
&	Ampersand	&
"	Quote	"
'	Apostrophe	'

- ▣ Additional HTML entities:
http://www.w3schools.com/tags/ref_entities.asp

HTML comment

124

- You can add **comment** without affecting the display of content.
 - ▣ A comment cannot appear inside a markup tag

```
<!-- This is a comment -->
```

- You can also *comment out* HTML code.

```
<body>  
  <!-- <p>first paragraph, hidden by comment</p> -->  
  <p>second paragraph</p>  
</body>
```

Optional tags

125

- In some cases, start tags and/or end tags may be omitted.
 - ▣ Even when omitted, the elements are still in the DOM tree
 - ▣ For details, refer to <http://dev.w3.org/html5/spec/Overview.html#optional-tags>

```
<body>  
  <p>first paragraph</p>  
  <p>second paragraph</p>  
</body>
```

```
<ul>  
  <li>item 1</li>  
  <li>item 2</li>  
</ul>
```

Grammar of HTML

126

- In addition to syntax, the HTML5 specification also defines proper usage of elements
 - what elements are available
 - what attributes an element can use
 - Meaning (semantics) of elements and attributes
 - Content model: what an element can contain
- A **valid** HTML document (or **conforming** HTML document) observes all these grammar rules
 - Check validity using <http://validator.w3.org/>

Why validation?

127

- HTML5 defines how to handle/correct invalid doc. However, validation is still useful to reveal potential problems:
 - ▣ Possible typo (e.g. `..)`
 - ▣ Unclear author's intent (e.g. `<h1>Overview</h2>`)
 - ▣ Conflict in expressed semantics (e.g. `` contains elements other than ``)
 - ▣ Likely misunderstanding of specification (e.g. `<input disable='false' ../>`)
 - ▣ scripts failing in hard-to-debug ways (e.g. two elements having the same value for 'id')

A valid HTML5 doc template

128

- A valid HTML5 document with empty body

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title></title>
    <meta charset="utf-8" />
  </head>
  <body> </body>
</html>
```


Semantics of HTML elements

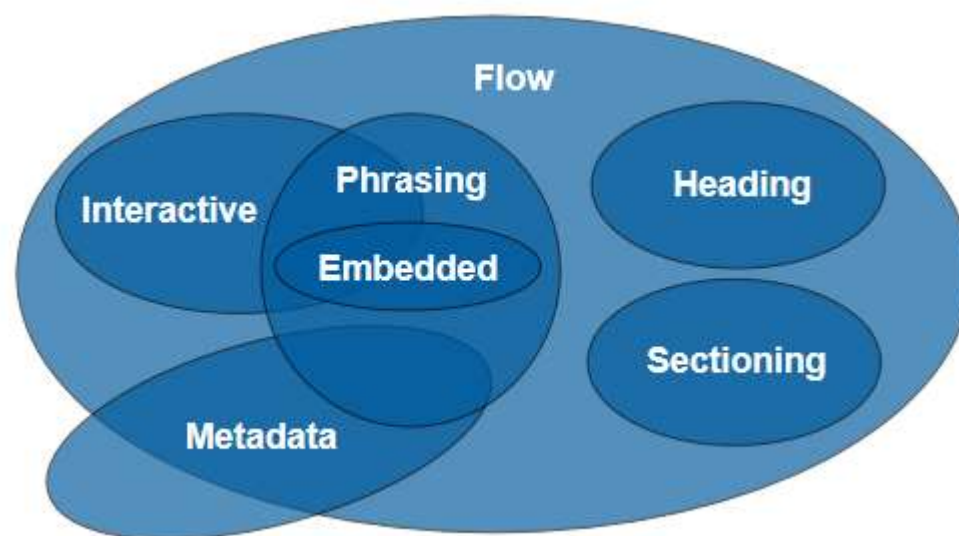
129

- You can read the HTML5 specification for the meaning of elements and attributes
 - ▣ E.g. use `` to stress some words, not to make them italics
 - ▣ W3C HTML5 spec: <http://dev.w3.org/html5/spec/Overview.html>
 - ▣ HTML5 Doctor quick ref:
<http://html5doctor.com/element-index/>
- Content model describes the expected content of an element
 - ▣ E.g. `` must contain ``, and `` can only appear inside `` or ``.

Categories of HTML elements

130

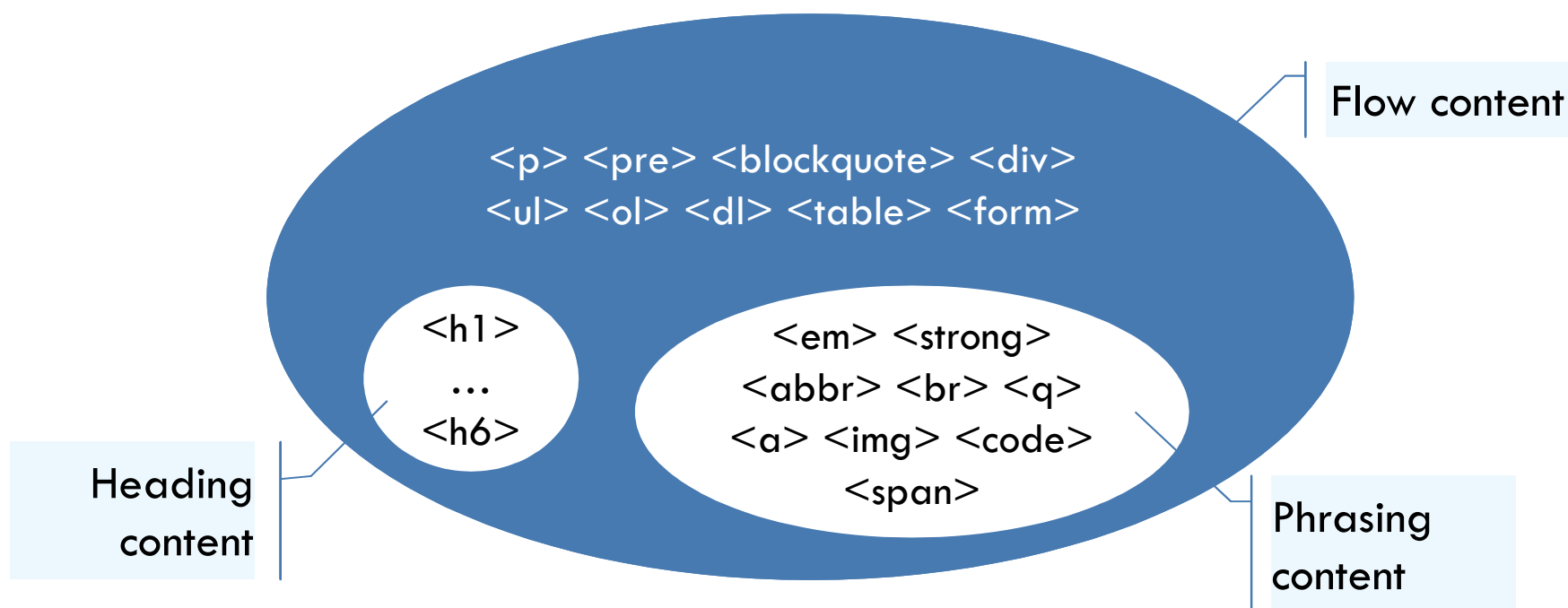
- Each HTML element falls into zero or more categories
- Content model describes elements of which categories are expected in the content of an element.
- <http://w3c.github.io/html/dom.html#kinds-of-content>



Categories of basic elements

131

- Most elements that are used in the body of documents belong to 'flow content'
- Some of these also belong to 'phrasing content', which is the text and inline markup inside a paragraph



Examples, 1

132

□ What can `<body>` contain?

▣ Check: <http://dev.w3.org/html5/spec/Overview.html#the-body-element>

▣ Are the following HTML fragments valid?

```
<body>
  <h1>..  
</h1>
  <p>..  
</p><p>..  
</p>
  <h2>..  
</h2>
  <pre>..  
</pre>
</body>
```

```
<body>
  Can I write text
  <em>directly</em>
  inside body?
</body>
```

```
<body>
  <li>..  
</li>
  <li>..  
</li>
</body>
```

Examples, 2

133

□ What can <div> contain?

▣ Check: <http://dev.w3.org/html5/spec/Overview.html#the-div-element>

▣ Are the following HTML fragments valid?

```
<div>
  <h1>..

# 


```

```
<div>
  Can I write text
  <em>directly</em>
  inside?
</div>
```

```
<div>
  <div>
    <div>..
```

Examples of content model

134

□ Are the following fragments conforming? If not, why?

```
<p>Todo:
  <ul>
    <li>first</li>
    <li>second</li>
  </ul>
</p>
```

```
<p>something
  <p>some inside</p>
</p>
```

```
<strong>
  <p>..</p>
</strong>
```

```
<pre>a <a href='a.html'>
link</a> inside </pre>
```

```
<a href='b.html'>nested
  <a href='a.html'> link</a>
allowed? </a>
```