Web page structure

Chapter 3

Outline

- A. Adding structure to HTML documents
  - Generic elements: <div> and <span>
  - HTML5 sectioning elements
- B. Syntax and Grammar of HTML
  - Syntax of HTML5
  - Correct syntax -> document tree
  - HTML5 defines standard error correction
  - Grammar
  - Validator and Valid HTML documents

A. Adding structure to HTML content

- Additional structure with <div> <span>
- HTML5 sectioning elements:
  - <header>, <footer>
  - <section>, <article>
  - <nav>, <aside>

Adding structure to HTML content

Two elements are used to group HTML content

- <span> – 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 <span>

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

```
<style>
    span.typo { border-bottom: 1px dotted red; }
</style>

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

Grouping block elements with <div>

- Group several block elements with <div>
  - E.g. a heading and the few paragraphs after it form a chapter

```
<body>
  <div class=”chapter" id=”chap1">
    <h1>HTML essential</h1>
    <p>The <em>structure</em> is important.</p>
    <p>another paragraph of this section…</p>
  </div>
  <div class=”chapter" id=”chap2">
    <h1>CSS essential</h1>
    <p>something about CSS …</p>
  </div>
</body>
```

Use class to describe the kind of the group

Use id to identify a particular group

Content of <div>

- <div> can contain both inline and block content.
- <div> can be nested. Useful to create hierarchical structure.

```
div.box { border: 1px solid green; }

<div class=‘box’>
   <div>first line</div>
   <div><img … /></div>
   <div><img … /></div>
</div>
```

div.box draws a green box to enclose the content.

The images and the text are separated into several <div>. This arranges them as a column.

<p> has similar effects, but <div> has no predefined margin.

Example: invalid usage

□ The following fragments are invalid..

```
<p>.. <div>div inside p</div> </p>
```

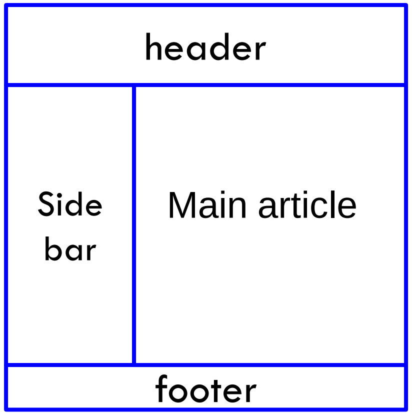
```
<span>.. <div>div inside inline element</div> </span>
```

```
<span>.. <h1>block element inside span</h1> </span>
```

Common sections in web pages

- □ Many web sites use similar high-level structure. Traditionally, these high-level sections are coded as <div>
- □ This example assumes that the main content of the page is a newspaper article, and the sidebar provides some related info about the article.

```
<body>
   <div id=‘header’>..</div>
   <div id=‘sidebar’ >..</div>
   <div id=‘main’>..</div>
   <div id=‘footer’>..</div>
</body>
```

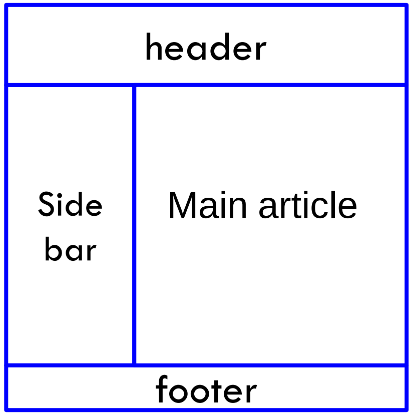


Sectioning elements in HTML5

- These elements provide more meaning than <div>
  - <header> – a header usually contains logo, title, and navigation menu
  - <footer> – a footer usually contains the author, copyright data, related links, and contact info
  - <aside> – consists of info that is tangentially related to the content around. Usually arranged as sidebar
  - <nav> – a navigation consists of links to other doc or parts in this doc
  - <section> – consists of some related content, typically with a heading. Examples include chapters and sections in a book.
  - <article> – a composition that forms an independent part of a page, e.g. forum post, magazine or newspaper article
    Note: an article may contain several sections. And you can also use a section to group several related articles.

Examples

```
<body>
   <header>..</header>
   <aside id=‘sidebar’>..</aside>
   <article id=‘main’>..</article>
   <footer>..</footer>
</body>
```

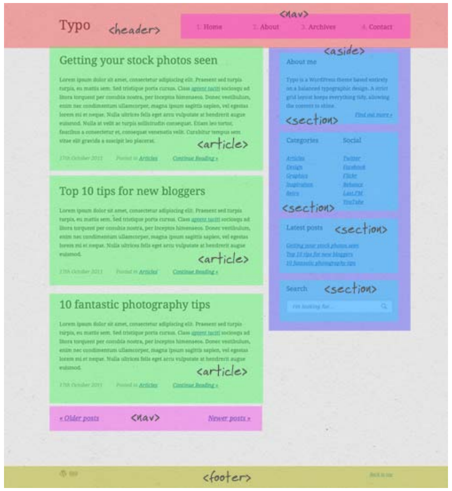


header

Side bar

Main article

footer



http://line25.com/tutorials/create-a-typography-based-blog-layout-in-html5

B. Syntax and Grammar of HTML

- The HTML5 specification defines the HTML language and how to use it correctly
- Valid document (Conforming HTML documents)
  - No syntax errors
    - E.g. <p class="review">…</p>
  - No grammatical errors
    - Correct element and attribute names, suitable content of elements
    - E.g. <a href='b.html'><code>push</code></a>

Topics in HTML5 syntax

- DOCTYPE declaration, <html>, <head>, <body>
- Text encoding
- Tags. Attributes. Special characters. Entities
- Syntax of elements and attributes
- More syntax rules: open and close tags. Void element. Nesting. Boolean attributes
- <! – Comment -->
- Optional tags

Basic structure of HTML

- An HTML5 document must start with the line <!doctype html>
- .. followed by an <html> element, which contains <head> and <body>

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

Character encoding

- An HTML document is a text file encoded using a character encoding.
- Character encoding converts a text character into a sequence of bytes
  - Common encoding: ISO-8859 (Western) (similar to ASCII), Big5 (Traditional Chinese), GB2312 (Simplified Chinese), UTF-8 (Unicode)
  - Example: 中 = (Big5) A4A4 = (Gb) D6D0 = (utf-8) E4B8AD
- A web browser must know the character encoding to read the content correctly.
- HTML documents may use <meta> tag to specify the character encoding
  - .. placed inside <head>

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

Markup tags

- 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>
  <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	&lt;	&gt;	&amp;	&quot;	&apos;

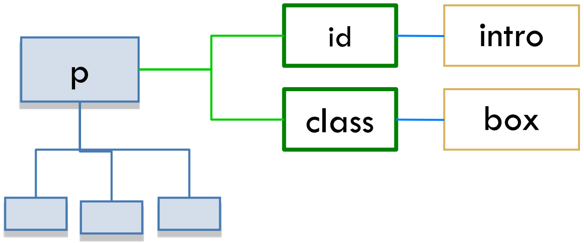
HTML elements

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

Syntax of elements, 1

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

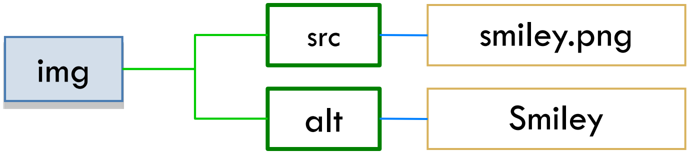
- A normal element has a start tag and an end tag
  - Element type (tag name) appears in both tags
- Its content is between the start tag and end tag
  - The content may be empty, text, some elements, or mixed content.
- Attributes of the element are included in the start tag
  - Value of an attribute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



Syntax of elements, 2

```
<img src="smiley.png" alt="Smiley" />
```

- A void element has no end tag.
  - The character '/' is optional in HTML5.



```
<img src="smiley.png" alt="Smiley" >
```

More syntax rules, 1

- 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

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

More syntax rules, 2

- □ 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

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

More syntax rules, 3

- 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;s intro'>This is John's introduction</p>
```

More syntax rules, 4

- 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" />
```

HTML comment

- 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

- 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

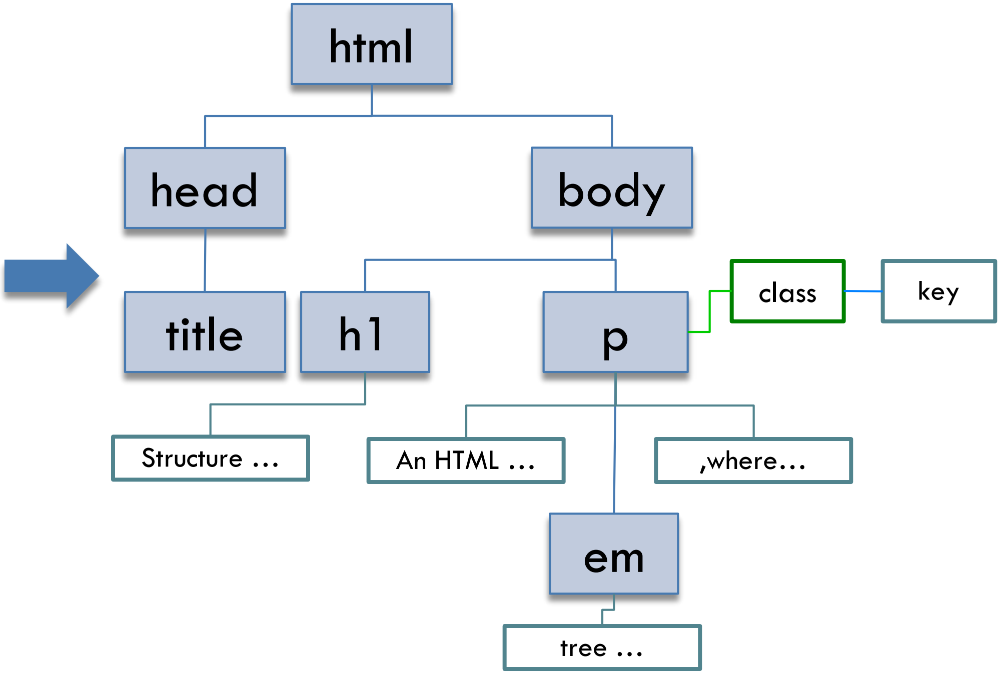
```
<body>
  <p>first paragraph~~</p>~~
  <p>second paragraph~~</p>~~
</body>
```

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

HTML doc → document tree

- After web browser converts an HTML doc (with no syntax error) to a document tree,
  - CSS style rules describe style properties of each element in the tree
  - JavaScript codes access and modify the tree

```
<!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>
```



Importance of correct syntax

- If there are syntax errors in the HTML document, the web browser will correct most errors automatically
  - The HTML5 spec defines how to handle syntax error consistently
  - However, this auto correction is performed silently, and the resultant document tree may be different from what would be expected by the HTML author

Grammar of HTML

- 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/

Browsers handle some grammar errors

- HTML5 defines how to handle some grammar errors and change the structure of the document tree.
  - The change may not be what would be expected by the HTML author…

```
<p>Todo:
  <ul>
    <li>…</li>
  </ul>
</p>
```

→

```
<p>Todo:</p>
<ul>
   <li>…</li>
</ul>
<p></p>
```

HTML validation

- Validation checks an HTML document against HTML grammar and syntax rules.
- Useful to reveal potential problems:
  - Possible typo (e.g. <a herf='#home'>..</a>)
  - Unclear author's intent (e.g. <h1>Overview</h2>)
  - Conflict in expressed semantics (e.g. <ul> contains elements other than <li>)
  - 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

- A valid HTML5 document with empty body

```
<!doctype html>
<html>
    <head>
       <title>something</title>
       <meta charset="utf-8" />
    </head>
    <body>   </body>
</html>
```

Semantics of HTML elements

- You can read the HTML5 specification for the meaning of elements and attributes
  - W3C HTML5 spec: http://www.w3.org/TR/html5/
  - HTML5 Doctor quick ref: http://html5doctor.com/element-index/
- Content model describes the expected content of an element
  - E.g. <ul> must contain <li>, and <li> can only appear inside <ul> or <ol>.

Examples

- Assume the following fragment is inside the body of an HTML document. The resultant HTML file is not valid. How to correct them?

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

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

```
<p><h1>heading</h1>
  paragraph</p>
```

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

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

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