

# Setup and Environment

## Web Architecture and Information Management [./] Spring 2009 — INFO 190-02 (CCN 42509)

Erik Wilde, UC Berkeley School of Information

2009-01-26



[<http://creativecommons.org/licenses/by/3.0/>]

This work is licensed under a [CC Attribution 3.0 Unported License](http://creativecommons.org/licenses/by/3.0/) [http://creativecommons.org/licenses/by/3.0/]

## Contents

• Abstract	2
• Assignments	3
• 1 Web Server Configuration	
◦ Web Servers	5
◦ File Handling	6
◦ Apache	7
◦ XAMPP	8
◦ Apache Configuration	9
• 2 HTML Basics	
◦ Structured Documents	11
◦ 2.1 Elements	
▪ HTML's Main Building Blocks	13
▪ Element Markup	14
◦ 2.2 Element Structures	
▪ Element Nesting	16
▪ Element Names	17
▪ End Tags	18
▪ Overlapping Elements	19
▪ Element Markup Errors	20
◦ 2.3 Attributes	
▪ Element Metadata	22
▪ Attribute Syntax	23
▪ Attribute Markup Errors	24
• Summary	25

## Abstract (2)

This lecture provides a hands-on overview of the various tools that are required to create and publish Web pages. This includes authoring a Web page, transferring the page onto your Web space on the Web server, and validating the Web page. For a more detailed look at the Web page in the context of the browser, the *Firebug* extension allows Firefox users to understand in detail how a Web page is structured and styled.

## Assignments (3)

- 12 Assignments over the course of the semester
  - first assignment was ungraded, the rest is graded
  - all assignments have to be satisfactory, top 10 will be counted
  - if you attend labs, you may skip one assignment
- Assignment grading is based on 0-5 points
  - 0 means not turned in
  - 1 means turned in but unsatisfactory
  - 2 means the bare minimum
  - 3 means ok but not great
  - 4 means good job but not quite perfect
  - 5 means everything solved perfectly

# Web Server Configuration

---

## Web Servers (5)

---

- Web servers do more than just “deliver files”
- They receive a request for acting on a resource
  - this may be a simple file retrieval (read file from disk, send over network)
  - additional information is available from the request's [header fields](#) [Web Foundations (URI and HTTP); HTTP Header Fields (1)]
  - the request URI may contain additional [query information](#) [Web Foundations (URI and HTTP); Query Information (1)]
  - the request may transmit complex data (such as a form submission)
- Processing can mean anything, it is transparent for the client
  - selection of a particular file
  - processing/transformation of a file
  - starting a process that creates a resource
- The result of processing yields a *resource representation*

## File Handling (6)

---

- Accessing files on your computer uses local programs
  - program installation associates file types with a program
  - multiple programs may be associated with the same file type
  - programs usually compete to be the *default association*
- Accessing documents with a browser uses browser knowledge
  1. some types are handled by the browser (HTML, GIF, JPEG, PNG, ...)
  2. some types are handled by browser plug-ins (Flash, ActiveX, ...)
  3. some types may be associated with external programs (PDF, ...)
  4. some types may not be known to the browser at all
- Program associations work different locally and on the Web
  - locally, a file is selected and the OS manages a list of programs
  - on the Web, a link is clicked and the browser handles the content

## Apache (7)



- [Apache](http://httpd.apache.org/) [http://httpd.apache.org/] is the most popular server on the Internet
- at least this can be said for a *general-purpose servers*
- more specialized scenarios (e.g., Google) need specialized servers
- Apache is open-source and available for various platforms
- Apache is the *A* in the popular [LAMP](http://en.wikipedia.org/wiki/LAMP_(software_bundle)) [http://en.wikipedia.org/wiki/LAMP\_(software\_bundle)] suite
- *Linux* as a free and open-source operating system
- *Apache* as the Web server processing incoming requests
- *MySQL* as the database for managing large amounts of data
- *PHP* as a simple language for server-side scripts
- (some people claim that the *P* stands for *Perl*)
- Many software packages can be easily installed in LAMP environments
- [IMP](http://www.horde.org/imp/) [http://www.horde.org/imp/] allows you to set up Web-based email access
- [mantis](http://www.mantisbt.org/) [http://www.mantisbt.org/] provides bug and issue tracking

## XAMPP (8)

- [XAMPP](http://www.apachefriends.org/en/xampp.html) [http://www.apachefriends.org/en/xampp.html] is a software installation package
  - available for a variety of platforms (Windows, Linux, MacOS, Solaris)
  - [Apache](#) [Apache (1)] as Web server
  - MySQL as a database
  - PHP and Perl as programming languages
  - control panel for ease of use
- Why running a Web server on the local computer?
  - browsing local files using HTTP
  - testing server configurations locally
  - testing server-based code locally
  - running a local database and a Web-based interface for it
- XAMPP is not required but recommended for this course



## Apache Configuration (9)

- Web servers need some configuration do to their job
- Configuration controls various aspects of server behavior
  - who am I?

```
<VirtualHost 128.32.78.26:80>
  ServerAdmin webmaster@ischool.berkeley.edu
  ServerName ischool.berkeley.edu

  RedirectMatch permanent ^/~(.*)$
  http://people.ischool.berkeley.edu/~$1
  RedirectMatch permanent ^(.*)$
  http://www.ischool.berkeley.edu$1
```

- where to look for documents to serve

```
# DocumentRoot: The directory out of which you will
# serve your
# documents. By default, all requests are taken from
# this directory, but
# symbolic links and aliases may be used to point to
# other locations.
#
DocumentRoot "/var/www/html"

<Directory /home/*/public_html>
  Options Indexes SymLinksIfOwnerMatch MultiViews
+Includes
  AllowOverride All
  XBitHack on
  DirectoryIndex index.html index.htm index.php
```

# HTML Basics

## Structured Documents (11)

- *Hypertext Markup Language (HTML)* is the language of the Web
  - *Hypertext* because the Web is a hypermedia system
  - *Markup* because documents are encoded using text
  - *Language* because HTML is used for communications
- *Markup Languages* are different from most file formats
  - many computer formats are binary encoded and not “just text”
  - *markup* allows structured documents to be encoded *as just text*
- Web data formats use markup as well as other encodings
  - *HTML* and *XML* are markup languages
  - *JavaScript* is also exchanged textually (but it's not markup)
  - images and other multimedia content is encoded as binary files

# Elements

---

## HTML's Main Building Blocks (13)

---

- HTML pages have a *header* and a *body*
  - the *header* contains information about the page
  - the *body* contains the actual document content
- Elements are used to mark up structures of the HTML document
  - block-level elements for paragraphs, lists, tables, ...
  - inline elements for text structures (links, emphasis, ...)
  - elements with specific semantics (e.g., embedding images)
- HTML defines rules for how these elements can be combined
  - additionally, HTML defines default layouts for these elements
  - some elements also require special handling (e.g., images)

## Element Markup (14)

---

- Elements contain content for HTML structures
  - the text of a paragraph
  - the items of a list
  - the content of a list item
  - emphasized text with a list item's text
- Elements are marked up using *tags*
  - start tags mark the start of an element: `<ul`
  - end tags mark the end of an element: `</ul`
  - start and end tags must always match
  - elements can never overlap, everything must be properly nested

# Element Structures

---

## Element Nesting (16)

---

- Elements can be nested to represent nested content
  - everything in HTML is nested (contained in body)
  - advanced Web pages often may not look like simple nested structures
  - representing a pages *logical structure* is important

```
<ul>
  <li>Unordered lists contain one or more items</li>
  <li>Each item <em>may contain <strong>markup</strong>
</em></li>
  <li>items can also contain further block-level content ...
  <ol>
    <li>... such as ordered lists</li>
  </ol>
</li>
</ul>
```

## Element Names (17)

---

- Elements may use lowercase or uppercase
  - *HTML* treats names as case-insensitive
  - *XHTML* treats names as case-sensitive
  - in XHTML, everything has to be lowercase
  - sticking to lowercase as a general rule is a good idea

```
<ul>
  <li>Unordered lists contain one or more items</li>
  <li>Each item <em>may contain <strong>markup</strong>
</em></li>
  <li>items can also contain further block-level content ...
  <ol>
    <li>... such as ordered lists</li>
  </ol>
</li>
</ul>
```

## End Tags (18)

- Elements can be closed explicitly or implicitly (if allowed)

```

<ul>
  <li>Unordered lists contain one or more items
  <li>Each item <em>may contain <strong>markup</strong>
</em>
  <li>items can also contain further block-level content ...
    <ol>
      <li>... such as ordered lists
    </ol>
  </li>
</ul>
<hr>

<!ELEMENT LI - O (%flow;)*          -- list item -->
<!ATTLIST LI
  %attrs;                            -- %coreattrs,
%i18n, %events --
>

<!ELEMENT HR - O EMPTY -- horizontal rule -->
<!ATTLIST HR
  %attrs;                            -- %coreattrs,
%i18n, %events --
>

```

## Overlapping Elements (19)

- HTML Elements can never overlap
  - any overlapping markup has to be interpreted somehow
  - depending on the browser, different things may happen

```

<h2>Overlapping elements</h2>
<ul>
  <li>Unordered lists contain one or more items</li>
  <li>Each item <em>may contain <strong>markup</em>
</strong></li>
  <li>items can also contain further block-level content ...
    <ol>
      <li>... such as ordered lists</li>
    </ol>
  </li>
</ul>

```



## Element Markup Errors (20)

- HTML Markup allows only certain variations
  - whitespace *after* the element name is allowed
  - whitespace *before* the element name is not allowed
  - the markup characters ( < > / ) always must be used properly

```
<ul>
  < li>Unordered lists contain one or more items</ li>
  <li >Each item <em>may contain <strong>markup</strong>
</em></li >
  <li>items can also contain further block-level content ...
  <ol>
    <li>... such as ordered lists</li>
  <ol>
</li>
</ul>
```

# Attributes

## Element Metadata (22)

- Elements may need additional information
  - usually not considered content, but data about data (thus, *metadata*)
- Elements and attributes represent HTML's full capabilities
- Some attributes can be used on many different elements
  - id can be used to assign a unique ID to an element
  - class and style are used to associate styling with elements
  - title assigns a human-readable title to an element
  - lang [<http://www.w3.org/TR/html4/struct/dirlang.html#h-8.1>] specifies the language of an element's content
  - on\* associate various events with an element
- Some attribute are specific to a certain element
  - href [<http://www.w3.org/TR/html4/struct/links.html#edef-href>] specifies the target of an a hyperlink
  - src specifies the source URI for an img
  - type [<http://www.w3.org/TR/html4/interact/forms.html#edef-type-INPUT>] specifies the type a form input

## Attribute Syntax (23)

- Attributes are specified in an element's start tag
  - they have a name and can occur at most once per element
  - they follow a simple name=value syntax
  - values must be quoted (with some exceptions), using single or double quotes

```
<h2 id="attributes">Attributes specify element
metadata</h2>
<ul type="square">
  <li>Unordered lists contain one or more items
  <li>Each item <em>may contain <strong title="'quotes' in
value">markup</strong></em>
  <li>items can also contain further block-level content ...
  <ol start='42'>
    <li>... such as ordered lists
  </ol>
</li>
</ul>
<hr>
<p><a href="http://validator.w3.org/check
/referer">validate page</a> with the <a
href="http://validator.w3.org/">w3C validator</a></p>
```

## Attribute Markup Errors (24)

- Attributes have to respect a number of limitations
  - they cannot be repeated on the same element
  - some attributes have restricted value spaces
  - quotes within values must be quoted with the other type of quotes
  - quotes around values must be properly paired

```
<h2 id="attributes" id="error">Attributes markup syntax
errors</h2>
<ul type="diamond">
  <li>Unordered lists contain one or more items
  <li>Each item <em>may contain <strong title="'quotes' in
value'>markup</strong></em>
  <li>items can also contain further block-level content ...
  <ol start='42">
    <li>... such as ordered lists
  </ol>
</li>
</ul>
```

## Summary **(25)**

---

- Web Servers are not required to serve HTML pages
  - local copies can be handled like regular documents
  - running a local Web server makes Web development easier
- HTML pages use markup to represent document structures
- Markup should be validated before it is published on the Web
- Markup on the Web can be inspected as source and within the browser