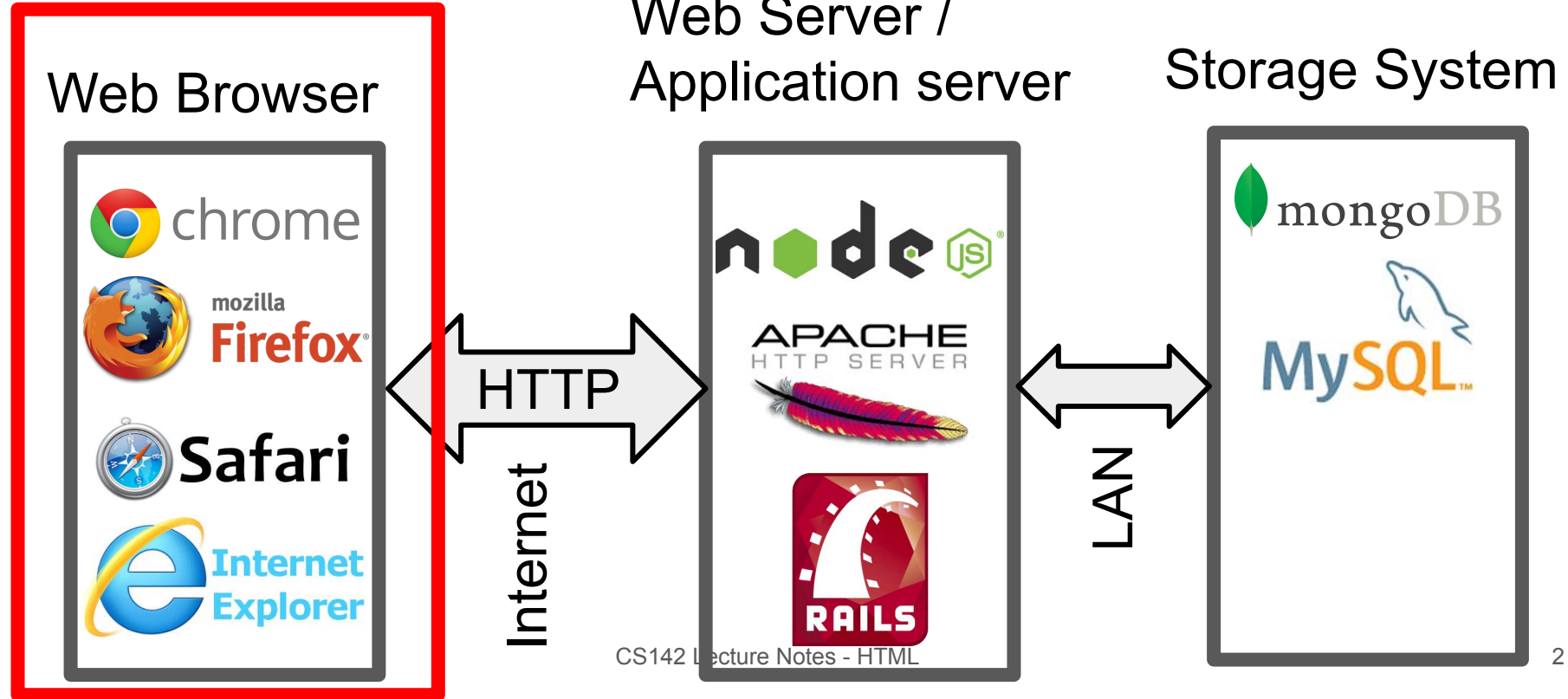


# HyperText Markup Language (HTML)

Mendel Rosenblum

# Web Application Architecture



# Browser environment is different

Traditional app: GUIs based on pixels

Since 1970s: software accessed mapped framebuffers (R/G/B)

Toolkits build higher level GUI widgets (buttons, tables, etc.)

Web browsers display **Documents** described in **HTML**

Until HTML5's canvas region, you couldn't write pixels

Make applications out of documents

Early web apps: Multiple documents (pages) with 'form' tag for input

Current: Use JavaScript to dynamically generate and update documents

# HTML: HyperText Markup Language

Concept: **Markup Language** - Include directives with content

Directives can dictate presentation or describe content

Idea from the 1960s: RUNOFF

Examples: `<i>italics word</i>`, `<title>Title words</title>`

Example of a **declarative language**

Approach

1. Start with content to be displayed
2. Annotate it with **tags**

HTML uses `< >` to denote tags

# HTML tags

Tags can provide:

Meaning of text:

- `<h1>` means top-level heading
- `<p>` means paragraph
- `<ul><li>` for unordered (bulleted) list

Formatting information (`<i>` for italic)

Additional information to display (e.g., `<img>`)

Tags can have tags inside (nesting supported) - Document forms a tree

# Example of HTML - Start with raw content text

## Introduction

There are several good reasons for taking

CS142: Web Applications:

You will learn a variety of interesting concepts.

It may inspire you to change the way software is developed.

It will give you the tools to become fabulously wealthy.

# Example of HTML - Annotate with tags

`<h2>Introduction</h2>`

`<p>`

There are several good reasons for taking

`<i>CS142: Web Applications</i>:`

`</p>`

`<ul>`

`<li>`

You will learn a variety of interesting concepts.

`</li>`

`<li>`

It may inspire you to change the way software is developed.

`</li>`

`<li>`

It will give you the tools to become fabulously wealthy.

`</li>`

`</ul>`

# Browser doesn't care but programmers do

`<h2>Introduction</h2>`

`<p>`

There are several good reasons for taking

`<i>CS142: Web Applications</i>`:

`</p>`

`<ul>`

`<li>`

You will learn a variety of interesting concepts.

`</li>`

`<li>`

It may inspire you to change the way software is developed.

`</li>`

`<li>`

It will give you the tools to become fabulously wealthy.

`</li>`

`</ul>`



# Example HTML - Browser output

## Introduction

There are several good reasons for taking *CS142: Web Applications*:

- You will learn a variety of interesting concepts.
- It may inspire you to change the way software is developed.
- It will give you the tools to become fabulously wealthy.

# HTML Evolution

Influenced by browser implementation quirks

What to do if you see “**<p>**Some text” (missing closing **</p>**)?

1. Complain bitterly about malformed HTML.
2. Figure out there was a missing **</p>**, add it, and continue processing.

Forked into HTML and XHTML (XML-based HTML)

**XHTML is more strict about adhering to proper syntax**

For the HTML class projects (1, 2, and 3) we will use XHTML

Users came to depend on browser quirks, so browsers couldn't change

# Example XHTML document

```
<?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" xml:lang="en" lang="en">
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

# Basic Syntax rules for XHTML

**Document:** hierarchical collection of **elements**, starting with `<html>`

Element: start tag, contents, end tag

Elements may be nested

Every element must have an explicit start and end

Can use `<foo />` as shorthand for `<foo></foo>`

Start tags can contain **attributes**:

```

```

```
<input type="text" value="94301" name="zip">
```

```
<div class="header">
```

# Need to handle markup characters in content

To display a literal < or > in a document, use entities:

&lt;        Displays <

&gt;        Displays >

&amp;        Displays &

&quot;      Displays "

&nbsp;        Nonbreaking space (won't insert a line break at this space)

Many other entities are defined for special characters.

Whitespace is not significant except in a few cases (e.g. textarea, pre tags)

# Example XHTML document structure

```
<?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" xml:lang="en" lang="en">
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

# XHTML document structure

```
<?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">
```

Indicate that this is an XHTML document, conforming to version 1.0 of the standard; use these lines verbatim in all the web pages you create for this class.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

Outermost element containing the document

<head>: Contains miscellaneous things such as page title, CSS stylesheets, etc.

<body>: the main body of the document

# Common usage XHTML tags

<code>&lt;p&gt;</code>	New paragraph
<code>&lt;br&gt;</code>	Force a line break within the same paragraph
<code>&lt;h1&gt;</code> , <code>&lt;h2&gt;</code> , ...	Headings
<code>&lt;b&gt;</code> , <code>&lt;i&gt;</code>	Boldface and italic
<code>&lt;pre&gt;</code>	Typically used for code: indented with a fixed-width font, spaces are significant (e.g., newlines are preserved)
<code>&lt;img&gt;</code>	Images
<code>&lt;a href="..."&gt;</code>	Hyperlink to another Web page
<code>&lt;!-- comments --&gt;</code>	Comment tags



# Common used XHTML tags - continued

<code>&lt;table&gt;</code> , <code>&lt;tr&gt;</code> , <code>&lt;td&gt;</code>	Tables
<code>&lt;ul&gt;</code> , <code>&lt;li&gt;</code>	Unordered list (with bullets)
<code>&lt;ol&gt;</code> , <code>&lt;li&gt;</code>	Ordered list (numbered)
<code>&lt;div&gt;</code>	Used for grouping related elements, where the group occupies entire lines (forces a line break before and after)
<code>&lt;span&gt;</code>	Used for grouping related elements, where the group is within a single line (no forced line breaks)
<code>&lt;form&gt;</code> , <code>&lt;input&gt;</code> , <code>&lt;textarea&gt;</code> , <code>&lt;select&gt;</code> , ...	Used to create forms where users can input data

# Commonly used tags: <head> section

- <title> Specify a title for the page, which will appear in the title bar for the browser window.
- <link> Include CSS stylesheets
- <script> Used to add Javascript to a page (can be used in body as well)

# HTML differences from XHTML

HTML supports the same tags, same features, but allows quirkier syntax:

- Can skip some end tags, such as `</br>`, `</p>`

- Not all attributes have to have values: `<select multiple>`

- Elements can overlap: `<p><b>first</p><p>second</b> third</p>`

Early browsers tried to "do the right thing" even in the face of incorrect HTML:

- Ignore unknown tags

- Carry on even with obvious syntax errors such as missing `<body>` or `</html>`

- Infer the position of missing close tags

- Guess that some `<` characters are literal, as in "What if `x < 0`?"

- Not obvious how to interpret some documents (and browsers differed)

# Newer HTML - HTML5

- Additions tags to allow content definition
  - `<article>`, `<section>`, `<header>`, `<footer>`, `<summary>`, `<aside>`, `<details>`
  - `<mark>`, `<figcaption>`, `<figure>`
  - `<nav>`, `<menuitem>`
- Drawing
  - `<svg>` - Scalable Vector Graphics - Draw shapes
  - `<canvas>` - Draw from JavaScript - 3D with WebGL
- Timed media playback: `<video>` and `<audio>`

# Another markup language: Markdown

# Heading

## Sub-heading

### Another deeper heading

Paragraphs are separated  
by a blank line.

Two spaces at the end of a line leave a  
line break.

Text attributes `_italic_`, *\*italic\**, `__bold__`, **\*\*bold\*\***,  
``monospace``.

Horizontal rule:

---

Bullet list:

- \* apples
- \* oranges
- \* pears

Numbered list:

1. apples
2. oranges
3. pears

A [link](http://example.com).

# Markdown language example output

## Heading

### Sub-heading

#### Another deeper heading

Paragraphs are separated by a blank line.

Two spaces at the end of a line leave a  
line break.

Text attributes *italic*, *italic*, **bold**, **bold**, monospace.

Horizontal rule:

---

Bullet list:

- apples
- oranges
- pears

Numbered list:

1. apples
2. oranges
3. pears

A [link](#).