



CGT 353 Lecture I

Introduction to Rich Internet Applications,
Background, and Possibilities

Introduction

- The Web is and has been changing - away from static, boring pages...
- With **Rich Internet Applications (RIAs)**, developers now have an option...
- All the interfaces in the Adobe line are nearly identical... We will use Adobe Flash as a tool to create some RIAs
- We need to see how Flash fits in with other technologies...
- Flash should not be utilized by itself (use with Javascript, HTML, etc...)

Rich Internet Applications (RIAs)

- Rich Internet Applications (RIAs) are web applications that have most of the characteristics of desktop applications, typically delivered either by way of a standards based web browser, via a browser plug-in, or independently via sandboxes or virtual machines.
- Examples of RIA frameworks include Ajax, Curl, GWT, Adobe Flash/Adobe Flex/AIR, Java/JavaFX, Mozilla's XUL, OpenLaszlo and Microsoft Silverlight.



Rich Internet Applications (RIAs)

- These are web application that have characteristics of desktop applications.
- Rich Internet applications (RIAs) offer a rich, engaging experience that improves user satisfaction and increases productivity.
- Using the broad reach of the Internet, RIAs can be deployed across browsers and desktops.



Difference between RIAs & other Web Apps

- The key difference between RIAs and other Internet applications is the amount of interaction in the interface. In a traditional page-based Internet application, interaction is limited to a small set of standard controls such as checkboxes, radio buttons, form fields and buttons.
- This severely limits our ability to create usable and engaging applications, and most Internet applications have been clumsier and more difficult to use than their desktop counterparts.



Designing for your Audience

- For Whom Are You Developing?
 - Always develop for the lowest common denominator (within reason)
 - Good rule of thumb to design for the upper 84% of the population

HTML5

- The newest iteration of HTML
- Many of the existing HTML tags you know still work
- Some of the HTML tags you know have been deprecated or removed
- W3C Working Draft:
<http://www.w3.org/TR/html5/>

What is HTML5?

- It's a collection of new components
- It's not one entity
 - Before, you could detect for certain things, like browsers or plugins
 - Similarly, you can detect for HTML5 components before you attempt to use them, such as video or canvas
 - You **cannot** simply detect HTML5 though

What is HTML5?

- You can still detect browsers
 - Based on the browser, you can research whether the HTML5 you want to use will work on that browser.

What is HTML5?

- Your older HTML can be upgraded to become HTML5... or you can leave your old HTML the way it is, as long as it calls the correct doctype
- New Doctype:
 - `<!DOCTYPE HTML>`

Page Setup Requirements

- The first line of every XHTML5 document should always be the following XML declaration

<?xml version="1.0" encoding="UTF-8" ?>

- The second line of each document should use the XHTML5 DTD for the <!doctype> tag.

<!DOCTYPE html>

Page Setup Requirements

- The next line of code should be the opening `<html>` tag. The `<html>` tag in every document should always be **`<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">`**
 - `xmlns` specifies the XML namespace for a document.
 - `xml:lang` specifies the contents and attributes as English and is a required accessibility standard.
 - `lang` specifies the content of a tag as English and is a required accessibility standard.

Page Setup Requirements

- At the top of every XHTML5 file, between the <head> and </head> tags, place a <meta> tag as shown below:
<meta charset="utf-8" />
- The encoding type (UTF-8) should match the encoding in the XML declaration above.

Validating

- The Total Validator tool is pretty handy for validating your HTML5 documents.

<http://www.totalvalidator.com/tools/index.html>

- There are a few validator add-ons to the browsers, some do HTML5 well, others do not.

Parts of HTML5 doc

- `<section>`
 - `<section>` is a set of content that you could store as an individual record in a database
 - General format:
`<section>`
 `<h1>`Any level of heading`</h1>`
 The rest of the content
`</section>`
 - It is still advisable to use a `<div>` to perform styling

Parts of HTML5 doc

- `<header>`
 - Do not confuse with the `<head>` element
 - `<header>` is a group of introductory or navigational aids
 - It is the head of a new section
 - Use the `<header>` at the start of your page or inside an `<article>`, as examples.

Parts of HTML5 doc

- `<footer>`
 - As you might expect, represents the footer for the section it applies to.
 - Where you have a header, you may often find a footer that goes with it, again for the main page, or in a section or article, etc.

`<section>`

Section content appears here.

`<footer>`

Footer information for section.

`</footer>`

`</section>`

Parts of HTML5 doc

- `<article>`
 - Use `<article>` when you have a page that is laid out like syndication. For example, an RSS feed, course announcements, forum post, magazine article, etc.

`<article>`

`<h1>Title</h1>`

`<p>paragraph information about the
topic</p>`

`</article>`

Table layout

```
<table style="width:500px; padding: 1px; margin: 1px; border:0px" title="Listing of Server Variables">
  <thead>
    <tr>
      <th style="border: 1px solid #000000;">Key</th>
      <th style="border: 1px solid #000000;">Value</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td><em>a listing of server variables</em></td>
    </tr>
  </tfoot>
```

Table layout continued

```
<tbody>
  <?php
  foreach($_SERVER as $key=>$value)
  {
    ?>
    <tr>
      <td style="border:1px solid #000000;"><?php echo $key; ?></td>
      <td style="border:1px solid #000000;"><?php echo $value; ?></td>
    </tr>
    <?php
  }
  ?>
</tbody>
</table>
```



Assignment