Web Design

Prof. Anselm Spoerri
aspoerri@rutgers.edu

Course Overview

- Course Goals
- Approach
- Gameplan Schedule | Grading | Software Tools

Your Guide

Web Basics

- URL Basics
- HTML Elements & Tags Block-level vs. Inline Elements
- XHTML
- Naming and Hierarchical Structure of Elements
- Simple Page Layouts

whereRU

Using innovative image and map technologies to

virtually experience Rutgers

Created by **ITI InfoVis class**

http://whereru.rutgers.edu

Course Goal

Create Website Prototypes with

Dynamic and **Responsive** Layouts

Course Goals

1. **Understand Key Web Design Principles**

   Web Design Matrix

2. **Competitive Design Analysis**

   whereRU = who to emulate ? learn from ?

3. **Web Design using XHTML/HTML5 and CSS**

   Strategic Skills | Basic Understanding of Key Web Technologies

4. **Group Project = Dynamic, Responsive Site**

   Access to whereRU assets | Create alternative site prototypes

5. **Contribute to whereRU Web Design Effort**

   Design Analytics | Prototype & Responsive Design

Course Goals

Web Design course will **Give You**

- **Hands-on Experience**
- **Practical Knowledge**
- **Marketable Skills**
- **Contribute to Unique & Special Project**

Approach

Conceptual, **Analytical** and **Technical Skills**

need to create **well designed** and **dynamic Web site** using

XHTML/HTML5, **CSS**, **client- and server-side scripting**

1 Learn MECHANICS

- Dreamweaver, Fireworks

2 Create MEANING

- Based on understanding: Visual Perception, Graphic Design.
- **Contribute to whereRU web design effort**
  - Provide Site and Competitive Analysis ☐ who to emulate

Learn together and from each other

- Students with different skill levels … **all will create a dynamic web site**

Approach (cont.)

Regular Semester

- **Hybrid Course**
  - **In Person**: Review Key Concepts | Demos and Help
  - **Online Lectures & Demos**

Summer Semester

- **Online Course**

Online Content

- Relevant **Lectures** and video demos
- Relevant **LyndaCampus** content

Expectation

- You will view Online Lectures and Videos **Before In Person Class**

Gameplan – Schedule

Lec 1 – **Introductions / Web Basics**

Course Overview

Web Basics: URLs, (X)HTML

Lec 2 – **Site Development Process**

Planning & Site Development Process ☐ Group Projects

Dreamweaver: Understanding Web Site Design | Create Web Page

Lec 3 – **Web Design Principles**

Web Design - Layout & Grid System

Dreamweaver: Add Navigation & Pages | Test & Upload

Lec 4 – **Cascading Style Sheets Basics**

Cascading Style Sheets (CSS)

Dreamweaver: CSS and Stylizing Content

Gameplan – Schedule (cont.)

Lec 5 – **Design Principles & CSS**

Web Design Principles Summary

Dreamweaver: Creating Flexible Layout

Lec 6 – **Layout Design & Advanced CSS**

CSS: Positioning Elements

Dreamweaver: Positioning Elements and Layout Design

Lec 7 – **Interaction Design**

JavaScript & Client-Side Scripting

Dreamweaver: Rollovers and Image Maps

Lec 8 – **Navigation Design**

Dreamweaver: Navigation Design

Fireworks: Images for Navigation Button States

Gameplan – Schedule (cont.)

Lec 9 – **Dynamic Web**

Dynamic Web

Programming Concepts

Server Side Scripting: PHP

Databases: Introduction to MySQL

Lec 10 – **Databases & Server Side Scripting**

How to use server side scripting to get data from a database

How to display database data in HTML page

Lec 11 – **Databases & Server Side Scripting**

Continue to work on server-side scripting and MySQL

Lec 12 – **Databases & Server Side Scripting**

Recap of Key Concepts in MySQL and PHP

Gameplan – Schedule (cont.)

Lec 13 – **Work on Group Projects | Responsive Layout**

Open lab session to work on group projects

Lec 14 – **Course Review | Criteria**

Open lab session to work on group projects

Course Review

Project Evaluation Criteria

Lec 15 – **Group Projects**

Present & Evaluate Group Projects

Grading

Individual Exercises – 52.5%

- **Quizzes** (10%) – open book, no redo
- **Short Assignments** (10%) – no redo
  - Practice the techniques and technical content covered in class.
- Ex1: **Create Website** (15%) – redo
  - Meaning: Evaluate a site of your choice
  - Mechanics: External CSS controls layout and interactive navigation structure; create at least five pages.
- Ex2: **Create Advanced Website** (15%) – redo
- **360 Evaluation** (2.5%)

Group Projects – 47.5%

- **Competitive Website Analysis** (15%) – redo
- **Group Website** (25%)
- **Post-Mortem Paper** (7.5%)

Gameplan (cont.)

Course Website

http://comminfo.rutgers.edu/~aspoerri/Teaching/WebDesign/Home.html

http://comminfo.rutgers.edu/~aspoerri/Teaching/WebDesignSummer/Home.html

- **Online Lectures & Video Demos**
- LyndaCampus content

Sakai

- Syllabus | Submit Assignments | Quizzes | Discussions | Resources

Software

- **Free Trials**
  - Adobe Dreamweaver (30 days – get toward end of semester)
  - Adobe Fireworks (30 days – get toward end of semester)
- **Adobe** in the **119 PC Lab**
  - Adobe Dreamweaver | Adobe Fireworks
- **Adobe** via **SoftwareAnywhere** Web Service

Gameplan (cont.)

SoftwareAnywhere

1. Login with your Rutgers ID:
http://account.comminfo.rutgers.edu

2. Login with your Rutgers ID:
http://sa.comminfo.rutgers.edu

3. Make Sure to give SA **access** to your computer and check that files **saved** on your computer.

LyndaCampus

https://lynda.comminfo.rutgers.edu/Login

Recap – Course Goals

You will **contribute** to a **special project**

whereRU – experience Rutgers virtually

Help develop alternate and responsive designs

Goal: **Rich Visual Experience**

Rutgers community

Prospective students

Alumni

Recap – Course Goals

Web Design course will **Give You**

- **Hands-on Experience**
- **Practical Knowledge**
- **Marketable Skills**
- **Contribute to Unique & Special Project**

Your Guide

Anselm Spoerri

- Computer Vision
- Filmmaker – IMAGO
- Information Visualization – InfoCrystal ☐ searchCrystal
- Media Sharing – Souvenir
- Rutgers Website

Web Design Feedback

- "Professor Spoerri would not just lecture to you all period, and would actually **really force students to learn** what he was teaching."
- "I like that Spoerri is **so willing to help you** out."
- "The professor is **very determined to help us learn** the material."
- "I felt the instructor was **one of the most helpful**, and **definitely the most prepared** of all the instructors I have had in 5 years at Rutgers."
- "Instructor was very good … the amount of material he posted on **the course website is amazing** ... very useful

URL - uniform resource locator

- "http://www.abc.com/aaa/bbb/ccc.html"
  - "**http://**" - hypertext transfer protocol - **scheme**
  - "**www.abc.com/**" - **server name** – host, domain name, top-level domain
  - "**/aaa/bbb/ccc.html**" - **path** through folder hierarchy

URL Basics

- Absolute URL
  - "**http://www.abc.com/aaa/bbb/ccc.html**"
  - ***"Complete street address"***
  - Info located on external server
- Relative URL
  - "**../../../xxx/yyy.htm**"
    - "../" = up 1 level => up 3 levels, then subdir "xxx" to get to "yyy.htm"
  - ***"Direction to neighbor's house"***
  - Anchor (same page), Internal (local)

Default "Home" Page = index.html

- Keeps out prying eyes out of directories (also instructor :).

Mechanics – Web Basics: HTML Elements & Tags

HTML is made up of **elements**

- Elements are denoted in HTML by using **tags**
- For the most part, you will **enclose content** you are marking up in **between tags**
- Tags look like this: **<tag>Content</tag>** (read as: open tag, content, close tag)
- Three major elements needed for an HTML page

  <html> - container for **all of our HTML code**

  <head> - **put data** for browser and other machines

  <body> - **put content** to show to the user

HTML – Example

```
<html>
    <head>
        Machine readable code (metadata) goes here
    </head>
    <body>
        User readable content goes here
    </body>

</html>
```

Some text elements

**<p>, <h1>, <h2>, <h3>, <ul>, **

paragraph, heading 1, heading 2, heading 3, unordered list, list item

Other elements

**<img>, <a>, , **

image, anchor, strong, emphasis

HTML – Inline- vs. Block-level Elements

Block-level elements (<p>, <h1>, <h2>, etc.)

- take up their own space **vertically**.
- force elements after them to jump to next line.

Inline-level elements (<a>, <img>, , , etc.)

- do not take up their own vertical space
- can be placed **inside** of other elements.

Note: **cannot place a block-level element inside of inline-level element**
document will not validate properly if you do

Mechanics – Web Basics: XHTML

XML = Language for creating other languages

- **Custom markup language** that contains tags for describe the data that they contain.
- If a **tag identifies** the **data**, then the data becomes available for other tasks.
- **Not as lenient** as HTML.

XHTML = HTML rewritten in XML

XHTML: Keep code **Consistent** & **Well Structured**

Use "Transitional" XHTML in Dreamweaver

- Allows for the use of deprecated tags

XHTML – Examples

- XHTML elements must be in **correct order**

  Content

  Content

- XHTML elements **must close**

  Content___

  Content

- XHTML elements **must be lowercase**

  <P>

- XHTML documents must have **one root**
  (Only one <html> element per document; no frames)

XHTML – Examples

- Content
  Content
- Content
  Content
-
-

Validation

- HTML = very forgiving markup language

  Java not as forgiving

- If content **validates** to a standard
  ☐ better optimized for search engines

- DOCTYPE = XHTML Transitional standard

- http://validator.w3.org/

- More info: http://www.w3schools.com/Xhtml/xhtml_validate.asp

Mechanics – Web Basics: XHTML (cont.)

Body of (X)HTML document encloses **Content** of Web page.

Required in XHTML:

The ***head*** and ***body*** tags and Closing **</*p*>** tag.

☐ **Dreamweaver** includes **required tags** & **declarations.**

Naming Elements
id="name" or **class="name"**

Useful with **div** (content blocks) and **span** (inline text) **elements**

Breaking up a Page into **Divisions (DIV)**

Creating a Line Break: **
**

Hierarchical Structure of Web pages
Elements contained inside another element (latter = parent, former = child)

Two methods for creating **Web Page Layout**

Tables : easy to create, modify and format in **DW**.

CSS: create, modify and maintain in **DW**.

Structure Your Pages

Divide **logical sections** of document into **div elements**

- ☐ Produces "linear / natural flow" of divs

Use **header elements** (h1, h2 …)

Use **comments** /* hello world */

Ordered & **Unordered Lists** easy to create in **DW**.