

Overview

Web 3 is a workshop-style class, meaning that students take greater responsibility for the scope and goals of their projects. That said, the exercises and projects will revolve around the projects/topics below.

Projects

Students must complete two primary projects and submit files to the instructor on the due date. Both projects will be judged by the following criteria, weighted equally: *quality of concept* (interpretation of the problem into a design idea, originality of the approach, clarity of idea to the end user), *quality of design* (aesthetic harmony of visual elements, attention to typography, degree to which the design supports the content), and *quality of code* (validity of HTML and CSS, efficiency of coding methods, management of edge cases).

Wordpress

The first emphasis will be on learning the fundamentals of Wordpress. While optimized for blogging, Wordpress can be used as a general-purpose Content Management System (CMS). Because it is fundamentally a collection of PHP files, you can mix in your own PHP alongside Wordpress's built-in functions, variables, and looping structures. And, of course, Wordpress enables us to build a site that can be managed and updated by a client who has no knowledge of HTML.

We'll use Wordpress to explore the contemporary notion of designing a *system*, rather than a static, one-time product. The site you build will require you to design and program templates, re-usable modules of code, and conditional rules that control the way elements (text, images, video) interrelate and combine. The rules may be simple or complex, but the outcomes they produce in the form of pages viewed at any given moment can be diverse.

Dynamic Art Piece

The second emphasis (and project) will be in building a more experimental web site that visualizes a data set or creates artwork in a dynamic way.

For inspiration we will be looking at work by the new generation of artist/programmers and studios like Jonathan Harris, Ben Fry, Aaron Koblin, Stewart Smith, and Lust.nl. We will be exploring technologies like MYSQL, RSS, and XML, and the ways they can be manipulated with PHP. For this project, invalid code and HTML hacks are allowed/encouraged.

Requirements & Grading

Students must complete the two projects, participate in class discussions and critiques, and post to the class blog as assigned. Periodic quizzes will cover technical material reviewed in-class. Weighting is as follows:

20%	In-class and blog participation
80%	Two web projects (40% each)

Absences will be subject to Watkins policy: see pages 44-50 of the student handbook. Work must be submitted on time so that all students can benefit from the group critiques. *Projects turned in after the due date will be subject to a one letter-grade penalty (e.g. an A- becomes a B-).*

Students With Disabilities

Per the Watkins 2011 handbook/course schedule, students with needs for academic or other accommodations are encouraged to contact the Director of Student Life as soon as possible to receive assistance in arranging appropriate accommodations. Further information may be obtained from the Director of Student Life at (615) 383-4848.

Supplies

Files

Students are required to bring their project and exercise files to each class and to maintain backups. A USB flash drive is the simplest way to accomplish this, though Watkins server space or Dropbox account may be another option.

Server Space

Students will continue to use their server space on <http://www.watkinswebdev.com> and are expected to post projects there. MYSQL databases will be set up as needed.

Software

Textwrangler (free, <http://www.barebones.com/products/textwrangler/>)

Dreamweaver CS5 (available in-class on lab computers)

Weeks 1–3

Review: HTML and CSS

Lecture & Demo Topics

Review of core principles of contemporary web design: separation of style and content, accessibility, usability, and progressive enhancement. Review of HTML5 tags, CSS selectors, and layout techniques using floats and positioning.

Sample Exercises (most will be collaborative)

- Fix a broken layout in a sample page.
- Reduce the CSS being used on a sample page through more efficient selectors.
- Review a design comp and propose the ideal markup for the content. Be prepared to discuss your reasons.

Resources & Reference

- <http://maxdesign.com.au/articles/css-layouts/>
- <http://alistapart.com>
- <http://positioniseverything.net>

Weeks 4–9

Wordpress

Lecture & Demo Topics

PHP review: variables, arrays, loops, and functions. Building a website with Templates, Variables, and conditional code. Working with Wordpress features like thumbnail images, custom fields, search, etc.

Sample Exercises (most will be collaborative)

- Modify the provided Wordpress theme (“Starter Kit”) to show or hide Author, Tag, and Category information.
- Create a home page that styles a “featured post” differently from the rest of the blog posts.

Resources & Reference

- <http://learn.wordpress.com>
- <http://codex.wordpress.org>
- <http://wpdesigner.com/2007/02/19/so-you-want-to-create-wordpress-themes-huh/>

Weeks 10–15

Designing with Code

Lecture & Demo Topics

Common web data structures: MYSQL XML, RSS. Using PHP to retrieve data from external sources. Manipulating datasets with PHP and outputting them to HTML. Using PHP and inline CSS to generate programmatic layouts.

Sample Exercises (most will be collaborative)

- Retrieve and output a portion of an external web page using PHP.
- Take a provided RSS feed and style the output it using HTML and CSS to reflect sameness and difference in the source material.

Resources & Reference

- http://www.mezzoblue.com/archives/2004/05/19/what_is_rssx/
- <http://fuelyourcoding.com/reading-xml-with-php/>

Project 1: Wordpress Site

Create a blog, entirely controlled by Wordpress, that serves as an online notebook of and about your design process.

You should design and structure the blog to accept multiple kinds of content: samples of your work in progress, others' design/artwork that inspires you, questions about design or code, web bookmarks, etc.

The success of this project depends *both* on the design of your blog and the content you add to it. While blogs are by their nature templated, you should not treat yours as a generic container: its design should express an idea or emphasis that is important to your work and methods. Take advantage of native Wordpress features like categories, tag, and pages to highlight difference and variation in your posts.

Requirements

- Your site must contain, at minimum, a permalink page for each blog post, a chronological archive of all posts, and an about page.
- While you are encouraged to use the class's "Starter Kit" theme as a foundation, your final blog should be customized — both visually and structurally — to reflect the interests, biases, and ideas in your own work. You do not need to follow the traditional "main column and sidebar" blog format when it comes to layout.
- You are expected to add posts to your blog before, during, and after the completion of the design/templates. No backdating!

Due Date

March 21, 2012 (Preliminary Grade: structure *must* be complete)

May 2, 2012 (Final Grade)

Project 2: Dynamic Artwork

Create an online visualization of an external dataset or source of visual material.

First choose an external source of text, data, or imagery to work with. Then build a site that generates a visual expression from that source. Your final pages may be primarily typographic or image-based, informative or beautiful, organized or haphazard, but they should enhance or comment on the source material, and they should change over time or in response to user input. It is crucial that you *sketch multiple ideas* — both in Photoshop and in code — before arriving at your solution. Do not feel beholden to traditional representations of your chosen source material; in fact, it may help to avoid them (What would the news report look like as a series of colors? How could a Flickr stream be expressed through typography?). Some possible data sources you might consider:

- The news, via Google or Yahoo: stories published in real-time, quantity of coverage that two opposing views receive, stories about a particular person, etc.
- Weather conditions, via Google or Yahoo: display a full report, the forecast for a given zip code, or focus just on a location's temperature, wind speed, etc.
- Twitter: timelines on a per-user basis, a hashtag over time, or a search term of the user's choosing.
- Wikipedia: articles on a given topic, edits to a particularly controversial entry.
- Flickr: images tagged with a particular phrase, popular images, etc.

Requirements

- Your site must use a source of text or images that you did not create, and must be approved by the instructor.
- Your site must not be static: it should change over time, in response to user input, depending on the user's location, or all of the above.
- Your site does *not* need to adhere to the traditional rules of accessible, efficient web design. You are free to make use of inline styles, Javascript dependencies, image-heavy layouts, browser bugs, and other hacks.

Due Date

May 2, 2012