Programming web-based educational media

Matthew X. CuringaAntonios Saravanos

**EDT 603 *Programming web-based educational media*, Spring 2013**

<!DOCTYPE html><html> <body> <blockquote> Anyone who has lost track of time when using a computer knows the propensity to dream, the urge to make dreams come true and the tendency to miss lunch.<br> <strong>Tim Berners-Lee</strong> <em>, inventor of the world wide web</em> </blockquote> </body></html>

**Description:** In this course students learn techniques of web programming to develop interactive, educational media. Using the Python programming language and web development technologies (HTML5, Javascript, CSS), students gain practice in the object oriented programming and design of interactive software. For their final project, students will create their own educational website.

**Key words:** computer science, web development, python, interaction design, django, html, html5, css, javascript, OOP

# Office Hours

Matt Curinga, Post Annex, Room 1

[mcuringa@adelphi.edu](mailto:mcuringa@adelphi.edu)

Tuesday 3pm - 5pm

Monday 4pm - 6pm

*and by appointment*

# Lab Hours

Hannah Groves, Harvey 104

[hannahgroves@mail.adelphi.edu](mailto:hannahgroves@mail.adelphi.edu)

Thursday 6:30pm - 8:20pm

# Goals and Objectives

This course builds on CSC 602 to move beyond basic programming concepts; students will gain expertise in building more complex computer programs, over many iterations. At the end of the course, students will be able to design educationally sound web-based learning media, solve moderately complex problems using OOP, collaborate on programming projects, and identify methods for teaching programming and web development.

Specific teaching and learning goals include:

* designing web-based interactions and multimedia to support learning
* coding effective user interfaces for learning
* implementing Universal Design goals for accessible web sites
* identifying effective methods for teaching more advanced programming concepts and web design skills

Specific software development goals include:

* modeling real world problems with software
* planning iterations of a project
* testing and debugging
* Object oriented programming concepts:
* Abstraction
* Encapsulation
* Objects & Classes
* Composition
* Inheritance
* Polymorphism

# Required Books

Duckett, J. T. (2011). [*Html & css: design and build websites (1st ed.)*](http://htmlandcssbook.com/). Indianapolis, IN: Wiley Pubishing, Inc.



# Optional Books

Anderson, T. (Ed.). (2008). [*Theory and practice of online learning*](http://www.aupress.ca/index.php/books/120146). Edmonton: AU Press.

Greenfeld, D., & Roy, A. (2013). [*Two scoops of django: best practices for Django 1.5.*](http://twoscoopspress.org/products/two-scoops-of-django-1-5-e-b%20ook-bundle)

# Bibliography

Castro, Elizabeth. 2010. *Html 5 Visual Quickstart Guide.* Peachpit Press. ISBN 9780321719614.

Hayward, J. (2011). *Django JavaScript Integration: AJAX and jQuery*. Packet Publishing. ISBN 1849510342

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. M. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software* (1st ed.). Reading Mass.: Addison-Wesley Professional. ISBN 0201633612.

Knuth, Donald. 1997. *The art of computer programming.* Addison-Wesley Pub. Co. Reading Mass. ISBN 9780201896831.

Moreno, R., & Mayer, R. (2007). Interactive multimodal learning environments. *Educational Psychology Review*, *19*(3), 309–326.

Moreno, R. (2006). Learning in High-Tech and Multimedia Environments. *Current Directions in Psychological Science*, *15*(2), 63 -67.

Shneiderman, B. (2000). Universal usability. *Communications of the ACM*, *43*(5), 84–91.

Shneiderman, B. (2002). Promoting universal usability with multi-layer interface design. *ACM SIGCAPH Computers and the Physically Handicapped*, (73-74), 8.

Teague, Jason. 2011. *CSS3: Visual QuickStart Guide.* Peachpit Press. Berkeley CA. ISBN 9780321719638.

Zelle, John. 2004. *Python programming: an introduction to computer science.*Franklin Beedle. Wilsonville, OR. ISBN 9781887902991.

# Schedule

|  |  |  |
| --- | --- | --- |
| Week | Date | Topic |
| 1 | 28-Jan | How the web works |
| 2 | 4-Feb | Document Structure, Text, and Lists |
| 3 | 11-Feb | Links and Style |
| 4 | 18-Feb | Media and Usability |
| 5 | 25-Feb | Layout and Responsive design |
| 6 | 4-Mar | Dynamic website and Django, **Single Page Website Due** |
| 7 | 11-Mar | URL Dispatching and Templates |
| - | 18-Mar | *Spring break* |
| 8 | 25-Mar | Defining Objects |
| 9 | 1-Apr | Saving Objects |
| 10 | 8-Apr | Customizing and Validating Forms |
| 11 | 15-Apr | Object composition and inheritance |
| 12 | 22-Apr | Searching Django models |
| 13 | 29-Apr | Users and Authentication |
| 14 | 6-May | Advanced Topics, Project Lab |
| 15 | 13-May | Wrap-up and Reflection |

# Class Sessions

## How the web works

**Read (optional):**

* [CERN: How the web works](http://home.web.cern.ch/topics/birth-web)

## Document Structure, Text, and Lists

**Read/Do:**

* Duckett, chapters 1, 2, & 3
* Bradley, S. (2012, November 5). Exploration Of [Single-Page Websites](http://www.smashingmagazine.com/2012/11/05/navigation-patterns-in-single-page-websites/). *Smashing Magazine*.
* [Web Basics Tutorials Introduction to HTML & Body elements](http://www.codecademy.com/courses/web-beginner-en-HZA3b/0/1). *Code Academy*.

**Example:**

* [Teaching Labor, basic](http://home.adelphi.edu/~mxc/html/labor1.html)

**Milestone:**

* choose a topic for your single page website
* bring an html document that has initial content for your project, organized using <h1>–>;<h6>, <ul> or <ol>, and <p> tags

## Links and Style

**Read:**

* Duckett, chapters 4, 10, 11, & 12

**Milestones:**

* update your site to use stylesheets
* add links to your site
* improve content

## Media and Usability

**Read:**

* [Universal Design](http://en.wikipedia.org/wiki/Universal_Design), a general overview
* [NNGroup: Usability 101](http://www.hh.se/download/18.5173bcf712de11663378000958/1341268042153/diskussionsuppgift_F5_nielsen.pdf)
* Read Marcotte, E. (2010, May 25). [Responsive Web Design](http://alistapart.com/article/responsive-web-design). *A List Apart*.

## Layout and Positioning

**Read/Do:**

* Ducket, chapters 8, 13 & 15
* *Code Academy* [HTML Basics III](http://www.codecademy.com/courses/web-beginner-en-f8mcL?curriculum_id=50579fb998b470000202dc8b) part 3 (*3. Div and span*)
* *Code Academy* [CSS Element Positioning](http://www.codecademy.com/courses/web-beginner-en-6merh?curriculum_id=50579fb998b470000202dc8b)

**Milestones:** - DIV tag - add positioning, margins, padding, float, etc.

## **Dynamic website and Django**

* **Single Page Website Due**

## URL Dispatching and Templates

## Defining Objects

## Saving Objects

## Customizing and Validating Forms

## Object composition and inheritance

## Searching Django models

## Users and Authentication

## Advanced Topics, Project Lab

## **Wrap-up and Reflection**

* **web application due**

# Assignments & Grading

There are two major projects in this course, the *single page website* and the *dynamic web application*. Other grades come from particiaption and improvement.

## Improvement (10%)

The two coding projects will be based on mastery of the skills of web development. Because students come to the course with different skill sets, part of their final assessment will be based on their improvement. In the first week of the course, students will post a short, narrative self-assessment of their web programming skills: HTML, CSS, Javascript, (Python or other) Programming, and general understanding of how the Interent and Web work.

At the end of the semester, each student will email the instructors with an updated assessment and the grade they think they deserve for their improvement.

## Single Page Website (40%)

Students will build a single-page, informational website on a topic of their choice. Their website will be composed of HTML, CSS, and (optionally) Javascript. It will display their fluency with the HTML elements, styling with CSS, and enhancing user experience with Javascript. The content of the website is entirely up to the student. They are encouraged to create a website on something they are already familiar with and interested in. This is an individual project.

The project will be graded on the following criteria:

1. **Technical design (15 pts):**
   * is html code well formatted and error free?
   * are html elements used appropriately to give semantic structure to the document?
   * are elements combined in ways to make complex layouts?
   * is the document elegant? does it use tags minimally, but in a way that makes it clear for developers to read?
2. **Graphic design (10 pts):**
   * aesthetics
   * usability
   * media (images, video, audio)
   * typography
3. **Information architecture (10 pts)::**
   * logical structure of information
   * organization of internal links on page
4. **Impact (5 pts):**
   * does the site deliver an important message?
   * do the design and content (copy) work together?

## Dynamic web application (40%)

Student teams will propose, design, and execute a project. The project can be completed individually, or in a 2-person team. The requirements for the project are that it demonstrates the students understanding of server-side and client-side web development. The instructor will work with teams to develop suitable projects.

# Resources

## HTML & CSS

* [W3C](http://www.w3.org/), read the official specifications
* [Treehouse](http://teamtreehouse.com/), free trial of web, design, and other tutorials
* [Mozilla Developer Network](https://developer.mozilla.org/en-US/learn), html, css, & javascript
* [W3 Schools](http://www.w3schools.com/), html reference and tutorials
* [CodeAcademy](http://codeacademy.com)

## Design & Usability

* web typography <http://webtypography.net/>
* [A list apart](http://alistapart.com/) design and layout
* [Nielsen/Norman Group](http://www.nngroup.com/articles/)
* [United States Section 508](http://en.wikipedia.org/wiki/Section_508_Amendment_to_the_Rehabilitation_Act_of_1973)
* <https://www.section508.gov/>
* <http://webaim.org/standards/508/checklist>
* [Usability.gov](http://www.usability.gov/index.html)
* [Research-Based Web Design & Usability Guidelines](http://www.usability.gov/guidelines/guidelines_book.pdf)