

Fast, Easy, Complicated, and Powerful Web

[ITP, Spring 2011, H79.2885.1](#)
[Dan Phiffer, dan@phiffer.org](#)

Description

The Web has long been celebrated for its revolutionary potential, giving individuals access to a global audience once reserved for wealthy media barons. We now have an unprecedented expressive capacity, using fast and easy online tools, available for little or no cost. This course makes an assessment of how these tools have evolved in response to actual users. It is a story of cobbled together software, most of which shipped flawed, but has been incrementally improved and iterated upon. We will investigate the biases and quirks that shape these readily available tools of expression. How have user interface design, web standards, source control, and bug tracking played a role in this process? Who are the people writing the code, how do they respond to the needs of users? The course will ask students to take critical look at the tools that underpin our everyday experience of the Web, while also exploring how to get the most out of them. Students will learn how to write themes and plugins for WordPress to build sites that "don't look like a blog," but also examine how colloquial design patterns like blogs are received by site visitors. Throughout, we will study and hack and misuse casual content management, asking how we might enhance the revolutionary competence of its users.

The course is composed of five sessions of lecture, demos, and discussion, with a 6th class meeting devoted to presenting and critiquing projects. We will have weekly programming assignments and short readings (no books are required). By the end of the course you will have a firm grounding in how to create a website from scratch, how to install and customize WordPress, and engage in a critical dialogue about web technology and how it is developed.

Schedule

Two sections of students will meet each Monday, for half the semester, from 6:00 p.m. to 8:55 p.m.

Section 1 schedule

1. January 24

2. January 31
3. February 7
4. February 14
5. February 28
6. March 7

Section 2 schedule

1. March 21
2. March 28
3. April 4
4. April 11
5. April 18
6. April 25

Week 1

When investigating why a piece of software is misbehaving it is often useful to build an extremely pared down test case that demonstrates the problem. This approach may also be helpful to understand the rough dimensions of what the course is about. We start the course by considering a very simple example of a dynamic content-managed website in PHP.

1. Introductions
2. Lecture: The web context
3. Group activity: Solving the maze
4. Demo: Diving into PHP
5. Reading: [Write Code Like You Just Learned How To Program](#)
6. Assignment: Starting by starting

Week 2

Before we get overly tempted into perfecting our homebrewed content management software, we instead train our focus on an established tool: WordPress. How does it work? What are its fundamental assumptions? We continue in the spirit of the first week and start with the minimum elements required to create a custom theme.

1. Informal project presentations
2. Lecture: The easy web
3. Discussion: What would you like to build?
4. Assignment: The first refactoring
5. Reading: [Wikileaks Exposes Internet's Dissent Tax, not Nerd Supremacy](#)

Week 3

While "the cloud" offers many useful free resources, we should also be capable of hosting our own websites independently. This week we look at how to install WordPress from scratch, how to back up, and how to *restore* from backup.

1. Discussion: Navigating the quasi-public
2. Lecture: The complicated web
3. Demo: How to be your own sysadmin
4. Assignment: Migrating to a new host
5. Reading: [Open Source is not a Democracy](#)

Week 4

Free Software is primarily defined by the explicit freedoms reserved in a product's license. It's important to also consider some social factors in the software development process itself. We also explore how our projects might be understood by distinct groups: our users, clients and potential collaborators.

1. Lecture: Outlaws, bike sheds, and meritocracies
2. Discussion
3. Project progress reports
4. Open lab time
5. Reading: [Lean Usability](#)
6. Assignment: Testing with actual users

Week 5

We round out our analysis of WordPress by looking at its extensibility features. We consider the basic components involved in writing a plugin. We also consider when you should use a custom post type and how to keep your site running smoothly under high traffic loads.

1. Lecture: Powerful tools from small parts
2. Demo: Callbacks, filters, and actions
3. Discussion: Testing feedback
4. Open lab time

Week 6

Let's see what you built!

1. Project presentations, critique