

Computing Everywhere:  
The Structure of the Web  
Matthew Heston & Jim Maddock

### **Overview**

Many communications students study or use web based ICTs, yet the Internet largely remains a complex, unapproachable, “black boxed” technology. We hope to demystify “the web” for the non-technical communications audience, enabling students to build a rudimentary theoretical understanding of web based computing into their work. We also teach students a basic web programming in order to both reinforce theoretical concepts and to provide a starting point if they wish to further develop their technical skills. This workshop will also allow students interested in pursuing a career in software project management or user experience to become more conversant with developers and designers.

### **Why is this Important?**

Students interested in digital media are asked to think critically about the design of web based ICTs and social media. However, the architecture and experience of such technologies is shaped by much more than superficial design decisions made during development, as design is often constrained by the structure of the Internet. For example, early websites could not support interactive or dynamics elements due to lack of browser support for Javascript and Flash. By providing a broad understanding of the structure of the Internet and the interaction between key web technologies, we enable students to understand and critique the integrated human and technical elements of the systems they study.

### **Improvements from Last Year**

Based on feedback from last year’s workshop, we included more interactive exercises. We created an exercise to better explain Javascript, and added a final activity in which students write code based on concepts they’ve learned throughout the workshop. We chose to focus exclusively on client side technologies because we found that server side computing was far too big a topic to explain in enough detail.

### **Learning Objectives**

After taking this workshop participants will be able to:

- Describe the basic structure of the internet
  - Describe the client-server model of computing
  - The role of the browser

- The role of different client side browser technologies (HTML, CSS, Javascript)
- Complete basic web programming tasks
  - Edit existing HTML and create basic static websites
  - Edit CSS in order to change the visual appearance of a web page

### **Teaching Experience**

**Matthew** has been a teaching assistant in COMM 227 here at Northwestern as well as delivered guest lectures in classes such as “Language and Technology” as a Master’s student at Illinois Tech. He also worked professionally as a software developer for 2 years where he developed primarily web applications.

**Jim** has been an undergraduate teaching assistant at the University of Washington for relevant classes such as *Information Visualization* and *Social Media and Data Mining*. He has also designed and developed full stack web applications for a number of research studies.

**Both instructors** taught this workshop together during the 2015-16 academic year and received positive feedback. All students felt that it should be taught again in the future.

### **Full Course Plan**

#### **Introductions (5 min)**

Introduce the instructors, talk about research interests and experience, and discuss learning goals for the course.

#### **Broad Theme 1: Structure of the Web (45 min)**

This module will create a basic understanding of the structure of the web in order to contextualize client side technologies within the broader framework of the web.

Activity 1.1 (10 minutes): Have students get in small groups (three to a group) and create a list of things they do on their computers everyday, and how much of this activity uses the web. We anticipate this will be primarily web applications. We will use this activity to get students to think about how many computing applications use the web.

Activity 1.2 (5 minutes) Ask students to think about different representations of one web application (probably facebook). Show students that facebook is a website, but also code, servers, etc. Using the previous groups, brainstorm the the architecture of Facebook, and explain the role of servers and clients in this context.

## **Broad Theme 2: Client Side Computing and the Browser (60 min)**

This module will unpack browser specific technologies (e.g. HTML, CSS, and Javascript) in order to show students that even complex web programming contains recognizable, straightforward elements that they can understand.

Lecture 2.1 (5 minutes): Introduction to HTML and the Browser. We will tell students that we're going to focus on the browser, and explain it in terms of our physical model of computing. We'll explain that there are three main technologies: html, css, and javascript.

Activity 2.1: (5 minutes) We will point students to a basic web page created by the instructors and ask them to view source ([http://jmaddock.net/computing\\_everywhere](http://jmaddock.net/computing_everywhere)). We will ask students to form small groups and discuss the relationship between this HTML source and what they see in the rendered webpage. The instructors will point out that much of the text in the source code is the same as the text on the website.

Activity 2.2: (20 minutes) When a student brings up HTML tags we will explain their basic functionality (these are the "code" that your browser knows how to "read"). We'll show students the developer console, and show them that they can mouse over elements in the code that will be highlighted on the web page. We'll then ask students to find certain HTML tags and identify their function.

Lecture 2.2 (5 minutes): We'll move on to CSS and explain that HTML is for creating structure, while CSS is for styling a web page. We'll show students the same "basic" web page without CSS to demonstrate its function. We'll also show how the developer console can be used to modify CSS.

Activity 2.3: (10 minutes) Students will return to the basic web page and be given a list of css related task to accomplish in the developer console. This list will have basic tasks challenges for more experienced students.

Activity 2.4 (10 minutes): We will assign each student group a more "complicated" (more HTML, CSS) website, and ask them to repeat exercises they performed on the basic web page, e.g., finding tags, changing styles. Doing so will show students that although some web pages can be much more complicated than our example, all web pages still consist of the same basic elements.

Lecture 2.3: (5 minutes) We will briefly explain that javascript web pages to be interactive. We'll show students drop down menus and update indicators as examples.

Activity 2.5: (10 minutes) We will provide a basic javascript/jquery function that selects a DOM and manipulates it. Students will describe the different components of the function (selecting, updating).

**Broad Theme 3: Bringing it Together: Build Your Own Webpage (20 min)**

This module will bring all three components together, and show students how they can use existing tools (bootstrap, jquery) to build their own webpages.

Activity 3.1 (20 minutes): Provide students with some boilerplate html/css to build their own web page using bootstrap. Provide students with a general design goal (header, sidebar, dropdowns), and ask students to create the design for the remainder of the class. Encourage students to experiment with elements they haven't seen before.