Web Architecture Review Sheet

Erik Wilde (School of Information, UC Berkeley)

INFO 190-02 (CCN 42509) — Spring 2009
May 11, 2009

Available at `http://dret.net/lectures/web-spring09/`

Contents

1   Introduction

In principle, the final exam covers all the material covered in the lectures (regardless of whether it is written down in the lecture notes, which most of it is). This means everything mentioned on `http://dret.net/lectures/web-spring09/` (lecture notes, required readings) can be part of the final exam. However, to make exam preparation a bit easier, the following list of topics highlights the important parts of the course, and also specifically excludes some of the topics we only touched briefly.

1.1   Setup

- How does the Web space setup for this course work, what are the different mechanisms involved for transferring data and making it available on the Web, and what is the difference between loading an HTML file from a local disk in the browser, to loading it via the Web with an HTTP URI?

1.2   HTML

- HTML is a language for describing Web documents. Why is it important or at least helpful to validate them? What kind of tools and methods did we discuss for HTML validation?

- There is *no* need to learn all HTML elements. However, it is important to know the overall structure of an HTML page (the head and the body) as well as to have a general idea of the available elements. We looked at text-level elements, bigger structures (lists, tables), images, and links.

- Why is the HTML document head important and where can you see document head information in a browser

- Why is it important and helpful to try to retain content structures in HTML? There are special elements just for this, and they can be used with CSS to specifically identify and then style content of a certain type.

- We only briefly looked at the box model of HTML/CSS, i.e. the way in which a browser formats an HTML page into a visual structure for display. There is no need to look at the specifics of the box model.

- Frames come in two flavors: Frames on the page level, and IFrames which are embedded Web pages within Web pages. What's the difference between these frames, and what are the general problems with frame-based pages and sites?

- Image maps will not be part of the final exam.

1.3   CSS

- Why does CSS exist, and what development of the Web in general and HTML in particular led to CSS?

- CSS is about selecting things in a Web page which will then be styled. What functionality is still missing; in other words, which problem can be solved with using HTML/CSS, and which problems cannot be solved?

- What's the general way of how CSS works? Looking at the CSS Zen garden nicely illustrates the power and mechanics of CSS.

- There is no need to learn all CSS properties, but it is important to have some basic understanding of what they do. They mostly apply to the HTML/CSS box model and set certain properties (spacing, colors, fonts, and many other things) of how content is turned into a formatted page.

- What are the three ways how CSS properties can be applied to an element?

- What is the best way to create HTML content so CSS code is reusable across HTML pages (e.g., for the layout of a complete site)?

- There is no need to learn all individual CSS selectors, but it is important to understand what they do in general, i.e. what they select, and how they support selections. They allow CSS authors to select those parts of a documents to which certain properties should be applied. We may ask for examples.

1.4   Browsers

- What is a browser? What kind of functionality is usually provided by a browser? Why is there more than one browser, and what are the differences between some of them?

- How does a browser do its core job: displaying a Web page? What Internet/Web technologies are required to make that work?

- What is the relationship between a browser, the Internet's DNS, the Internet's TCP, HTTP, HTML, and CSS?

- How are URI schemes supported and processed by a browser? What URI schemes are most popular and most widely supported?

- What is caching in a browser?

- What is scripting in a browser?

- How are different content types retrieved from URIs handled by a browser? What are the advantages, disadvantages, and side-effects of these methods?

1.5   HTML Forms

- What is the difference between a regular HTML page and an HTML form? What are the special actions a browser need to support to handle forms?

- What is the HTML page returned by a form submission? Why does it make sense to have data submission procedures broken up into several HTML forms?

- There is no need to learn all HTML elements for forms, but it is important to understand the general idea of how forms are represented in HTML. The accessibility and usability aspects are important, too.

1.6   Internet

- What is the difference between the Internet and the Web?

- What are the core protocols of the Internet? What Internet technologies are required in every Web client/browser?

- When looking at a Web page in a browser on a mobile phone, how does the telephone transmit the data? We did not look at any mobile phone protocols in particular, but it is important to understand that the mobile phone carrier acts as the Internet service provider in such a scenario.

- There is no need to look at how TCP works, the only important thing to understand is that it is the reliable end-to-end communications protocol that is used for transmitting HTTP traffic.

- What is DNS? Why is the DNS critically important for (almost) all Web interactions? How is the DNS name space organized?

1.7   Security & Privacy

- What are the three important concepts, and what do they mean? Identification, authentication, authorization.

- How do browsers support security for a scenario such as online banking? What are the risks that need to be addressed, and how are they addressed?

- How do browsers support privacy for a scenario such as sites tracking users in their surfing behavior? What are the risks that need to be addressed, and how are they addressed?

- There is no need to go through the different cryptographic methods, we only looked at these briefly.

- What is the difference between HTTP and HTTPS? When should users make sure they access sites via HTTPS?

1.8   URI & HTTP

- What is the most important functionality of URIs? They allow a well-defined way of identifying (almost) anything.

- What are URI schemes? What are some other schemes besides HTTP?

- What is HTTP's role on the Web and in a browser? What is the fundamental way of how Internet technologies (such as TCP and DNS) and HTTP play together to make the Web work?

- What other traffic besides HTTP is transmitted over the Internet?

- How does HTTP traffic look like? It is a text-based protocol, so HTTP messages are text messages. What is the overall structure of these messages? What are the two major message types, and how are they used in transmitting Web pages?

- There is no need to learn HTTP header fields or status codes. But it is important to understand what these things do, and how they are used in the context of a Web browser requesting Web pages.

- How does HTTP pipelining speed up Web surfing? What are the resources a browser needs to fetch to display a typical Web page?

- There is no need to look into HTTP authentication.

1.9   Site Navigation

- Why is site navigation important, and what is the problem of implementing site navigation for a large site?

- What is required on the server-side to implement site navigation efficiently? Some mechanism to combine the content of a specific page with the general navigation info for the site.

1.10   Cookies

- What are cookies good for, and how do they relate to HTTP? How are cookies created and exchanged between the browser and the site which created them?

- Why are sessions important from the user perspective? HTTP is a stateless protocol (there are only request/response pairs of message exchanges and no larger context); how do cookies work to allow sessions?

- What are 3rd party cookies? What are the privacy implications of 3rd party cookies? What are the commercial reasons why 3rd party cookies are so widely used?

- There is no need to look into the more advanced questions of how cookies relate to stateless HTTP interactions.

1.11   Multimedia Content

- What is the difference between images and graphics? Which content type is best for what kind of picture?

- What are the most popular image formats on the Web? What are the important differences between these formats? Based on certain use cases, what are the distinguishing features of these formats?

- The Web does not support audio or video per se. How do the popular video Web sites work? What is the critically important feature of most browsers that supports this infrastructure?

- What is the fundamental difference between streaming and downloads? For which scenarios/content do these two types of content handling work best?

- What is a Content Delivery Network? Why are these networks important for large-scale content providers?

- There is no need to look into any technical details of how picture, video, or audio encodings work.

1.12   Media Types

- Why is it important to have a way to talk about media types? Where is that information important on the Web, and how do browsers use that information?

- How do computers handle media types (regardless of the Web)? How does a computer know that a certain file has to opened with a certain program?

- What is the general structure of Web media types? It is important to understand the type/subtype structure, but it is not required to learn the individual types or subtypes. However, it is important to know at least three examples of popular media types on the Web.

- There is no need to look into the details of media type registration of fragment identification.

1.13 Internationalization

- What is the general problem of ASCII? It is convenient because every character is one byte, but it limits the character repertoire.

- What is the difference between a character and a glyph?

- Unicode is the current method for encoding virtually all characters that exist. How does Unicode solve the problem of encoding over 100'000 characters? There are various encoding optimized for various user communities; UTF-8 works well for western countries, UTF-16 works well for eastern countries.

- There is no need to look into the details of how UTF encodings work.

- What are the differences between internationalization and localization?

- What are the important things that need to be addressed for internationalization and localization? There is no need to memorize the list of issues, but it is important to remember that internationalization and localization is more than just characters or written text on Web pages.

1.14 Scripting

- What is the difference between HTML and DHTML? Is DHTML a language?

- The most important thing about scripting is the DOM, which is the "glue" between a Web page's HTML and the scripting code. The browser builds the DOM when it renders a Web page. Remember how Firebug allowed us to see how a script actively inserted HTML code because of user interactions (such as hovering the mouse pointer over an element).

- What is the difference between DHTML and Ajax? Ajax uses requests to the server to implement Web pages; the most popular example is Google Maps, where new map tiles are requested from the server because the user drags or zooms the map.

- JavaScript frameworks are important because they hide (some of) the complexity of DOM and scripting from users. There is no need to look into the details of specific frameworks, but it is important to understand why they exist, and why they are useful and popular.

1.15 Syndication

- What was the original motivation why syndication was invented and what were the first applications of it? What are the features and limitations of syndication?

- Syndication is "pull" as opposed to "push" formats. Why is the "pull" method useful and how do Web technologies play together to allow massive scalability of syndication feeds?

- What are the most important differences between RSS and Atom? RSS comes in a variety of sometimes conflicting variants, whereas Atom is one well-defined and extensible format.

- What is the general structure of feeds? What are some of the important fields that can be used for feeds or entries? There is no need to know all fields, but some properties (such as titles and time stamps) are very important.

- Podcasts are just one variant of feeds. What do they add and why? Can you use a podcast feed in a regular feed reader (such as Google Reader)?

- What is the difference between HTML and feeds? HTML is content of one page to be displayed in a browser, whereas a feed is mostly an "index" into a collection of information items. These items can be embedded and/or linked in the feed.

- Chrome (Google's browser) does not support feeds at all. Does it mean it's broken? Depends on the user needs, but some minimal feed formatting (such as in Firefox) or maybe even more sophisticated display (such as in Safari) may be useful for many users.

- What are possible ways to read a feed? It can be read directly in the browser (if the browser handles feeds well), in an Ajax application such as Google Reader, or in standalone feed readers. Some feed readers are even specialized; iTunes, for example, is a feed reader, but only works with podcasts.

- There is no need to look into the more advanced topics of syndication aggregation.

1.16   Location and Geocoding

- How does GeoRSS work and what kind of information is useful?

- How does KML work and what kind of information is useful?

- What are the different ways of how a mobile device can be located?

- What is the difference between EXIF location tags and GeoRSS? EXIF is embedded in the JPEG format, GeoRSS is published in the feed (such as the flickr feed) through which an image may be published.

1.17   Google Maps Mashups

- What is the difference between embedding a Google Maps IFrame and using the Google Maps API?

- Why does Google issue an API key and what is it used for? It is used for access control and accounting.

1.18   Web Semantics

- What are the two main approaches for representing semantics for Web resources? Microformats and Semantic Web.

- Microformats are simple to use and express simple things, what are popular examples, and what can they do?

- The Semantic Web is a more sophisticated approach than microformats, what is the main difference? The main difference is that the Semantic Web tries to solve the problem of semantics in general, whereas microformats are ad-hoc solutions for isolated problems.

- There is no need to look into the more advanced topics of RDF, RDFS, and OWL, the main languages of the Semantic Web.