Girl Develop It Albuquerque

Intro to Web Concepts

Supplemental Material v1.1.0
November 2015

Hello!

Welcome to Girl Develop It Albuquerque's Introduction to Web Concepts class. This packet contains material and notes that supplement the lecture slides.

Your Instructor

Rochelle Lewis
Front End Web Developer and Web Development Educator
at the CNM STEMulus Deep Dive Coding Bootcamp

GitHub: rlewis2892
Twitter: @RochelleUXUI
Web: www.rochellelewis.com
Email: hello@rochellelewis.com

What We'll Cover

In-Class Lecture Slides and Materials:

http://gdi.rochellelewis.com/intro-web-concepts

Also available at:
http://girldevelopit.github.io/gdi-intro-web-concepts

Terms & Technologies

Acronyms and Technologies, the World Wide Web, Languages Libraries & Frameworks

Building Websites

Web Standards, UX, UI, IxD, & IA, Tools of the Trade.

Web Dev Professionals

Web Teams, Hiring a Pro.

Terms & Technologies

HTML - Hyper Text Markup Language

Not really a “programming language”, but a *markup language* that structures the content on a web page. The term “markup language” is a throwback to when documents were literally “marked up” by editors.

CSS - Cascading Style Sheets

CSS is called the “style language” of the web. CSS tells the web browser how HTML elements should look: colors, backgrounds, borders, placement and positioning, and even simple animation.

WYSIWYG - What You See Is What You Get

WYSISYG (pronounced “whizzy-wig”) editors enable end-users to use an input interface to update web pages without having to write any code. Think of the text editor in Wordpress.

IDE - Integrated Development Environment

When you begin writing code, you’ll probably use a simple text editing program to create the files for your first projects. As your projects become more complex, using an IDE is generally preferred.

An IDE is software that provides a comprehensive toolset for software development. Generally, an IDE consists of a code editor, build automation tools, and a debugger. IDEs offer much more power and functionality than simple text editors can. Oftentimes, IDEs will be geared towards specific programming languages. For example: PhpStorm for PHP, IntelliJ for Java/JavaEE, Visual Studio for Microsoft languages like C# and .NET, etc.

Popular text editors: Notepad, Atom, Notepad++, Sublime Text.
Popular IDEs: PhpStorm, Eclipse, Xcode, NetBeans, Microsoft Visual Studio, IntelliJ.

CMS - Content Management System

A CMS is software that allows the publishing, editing, and modifying of website content through a central user interface. CMS software must be installed and set up on the web server. Each CMS will have their own unique advantages, disadvantages, and technology requirements.

Popular CMSs: Wordpress, Drupal, Joomla, and many more.

Open Source

Open Source software are programs in which the source code is available to the general public for use and/or modification. This means that Open Source software is free to use, and you can contribute to the code base.

Popular Open Source Software: Wordpress, Linux, Apache, MySQL, Firefox, Git, GIMP, InkScape.

NOT Open Source: OSX, Windows, iOS, Adobe Creative Suite... These are all examples of *proprietary software* that are copyright protected. Proprietary software usually requires a paid license to use.

API - Application Programming Interface

Web APIs allow you connect with other programs and use their data and assets. Have you ever “Signed in using Facebook”, or seen an embedded Twitter widget, Flickr album, or Google map? These are examples of functionality that can be enabled with the use of web APIs.

Popular Web APIs: Google Maps, Google Places, Facebook, YouTube, Twitter, Flickr, Photobucket, LinkedIn, and many more.

IP (Internet Protocol) Addresses

Like your home address, every computer on a network must have a unique address. This enables network traffic to be routed between two uniquely identifiable computers so they can exchange information. No two computers on a network can have the same IP address (this will cause an error).

Networks have IP addresses too. For example, on your home internet connection your router is assigned a public IP address by your ISP (Internet Service Provider - Comcast, CenturyLink, etc.), and then each device on your home network (each laptop, phone, tablet, printer, etc.) is then assigned a local IP address by your router.

An IP (IPv4) address is formatted as a series of four numbers separated by periods: 192.168.0.1, etc. IPv6 addresses look different, and will ultimately replace IPv4. IPv6 will be widely adopted soon because the world is rapidly running out of IPv4 numbers!

Further Reading:
IP Addresses for Beginners: https://goo.gl/VkEMz1

DNS - Domain Name Service

A directory that associates domain names (URLs) with IP addresses. DNS allows users to connect to web sites via URLs... In other words, we don't have to remember to enter 216.58.217.14 in our browser's address bar when we want to go to google.com.

> From Google.com (I like this succinct definition!):
> *"Domain Name Servers (DNS) are the Internet's equivalent of a phone book. They maintain a directory of domain names (URLs) and translate them to Internet Protocol (IP) addresses. This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses."*

SEO - Search Engine Optimization

The process of increasing your web site's perceived value to search engine web crawlers, raising its rank in search results.
Beware of anyone who promises to know how to "trick" the googlebot into giving your site top rank on Google. Nobody outside of Google knows the PageRank algorithm. SEO can be a murky field full of snake oil and shaky promises.

Further Reading:
Google SEO Starter Guide: http://goo.gl/RMbCFm

The World Wide Web

Overview

A typical web site is "hosted" on a web server. Web servers are often large computers connected to a network.

Steps involved when you visit a web site:

1. Type a web site address (URL) into the address bar of your browser.
2. DNS connects your browser to the hosting server for the website you want to visit.
3. Your browser requests files and data from the server to display the web page.
4. Files and data are sent to your computer for display.
5. Sometimes code must be compiled before being sent back to you.

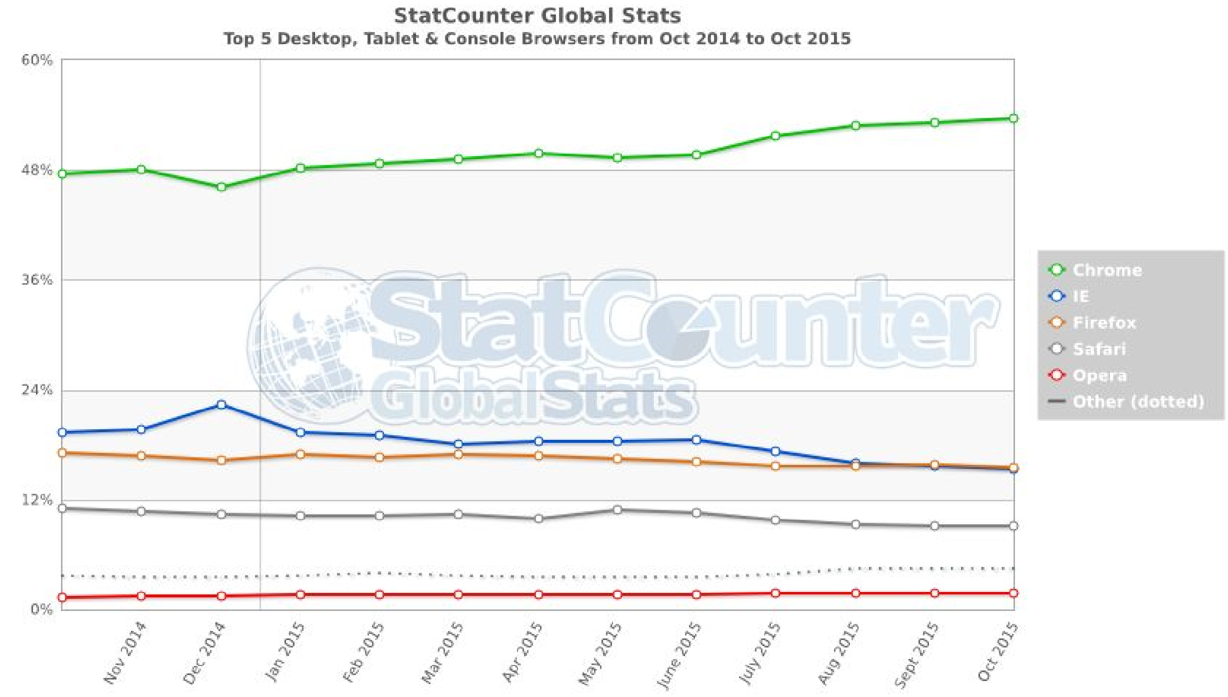
Clients

Technically speaking when we refer to a ***client***, we are talking about accessing a web site through a browser. Usually the browser is the "client". Clients request data from servers. Clients can also be web crawlers, command-line interfaces, and other applications too.

Servers

Servers are generally large, powerful computers that are connected to the internet for the sole purpose of “serving” up websites and content. Servers also store data in databases, and contain code that performs operations on stored data.

Updated Browser Usage Statistics: Oct. 2015



Languages & Frameworks

About Adobe Flash

You may have heard of Flash for web animation and video, but it is currently considered a dying technology. Flash is being widely replaced by HTML5, CSS3, and JavaScript animation.

Apple withdrew Flash support on the iOS platform in 2012 (citation needed). Due to this, Flash is not considered a good option for mobile-friendly development. Flash also requires a plugin to run, and has dwindling support.

Technologies That Are Not Languages: Libraries & Frameworks

Libraries are a set of programming shortcuts for a language. jQuery, MooTools and Prototype are examples of *JavaScript libraries*. They are written in JavaScript, and are NOT languages of their own. They are prewritten JavaScript tools that make development easier. Libraries are WIDELY used in web and software development.

Frameworks are a collection of libraries and other tools that help you rapidly build applications using a particular language, usually from scratch.

Front End Frameworks:

- Bootstrap (HTML, CSS, JavaScript)
- AngularJS (JavaScript)

Back End Frameworks:

- Ruby on Rails (Ruby)
- Django (Python)
- Cake, Laravel (PHP)

Specifications are the technical guidelines and rulebook governing the use of a programming language. These are very detailed, formal, and written for a technical audience.

Java !== JavaScript

Java and JavaScript are two COMPLETELY DIFFERENT programming languages. Believe it or not, they really have nothing to do with each other.

Java:

- Server-side
- Object-oriented
- Requires special parser (JVM)
- By Sun Microsystems

JavaScript:

- Primarily client-side, with server-side JavaScript on the rise.
- Primarily procedural, with Object Oriented JavaScript on the rise.
- Runs in the browser
- By Netscape

Web Standards

- Online code validators can help check your HTML and CSS for errors.
  - W3C Markup Validation Service: https://validator.w3.org/
  - W3C CSS Validation Service: https://jigsaw.w3.org/css-validator/

- **Accessibility Standards** ensure that people with disabilities can access the web. This is very important and REQUIRED for public and education projects.
  - Web Accessibility for Beginners: https://goo.gl/3mjbI8

- **Semantic guidelines** are akin to “best practices”. See language specs and documentation. The Mozilla Developer Network is a recommended resource.
    - MDN: https://developer.mozilla.org

- **Character encoding** is important for internationalization. UTF-8 is standard for the web. It ensures all the characters will be displayed properly on your pages internationally.

- **Metadata** is “data about data”. This is part of “best practices” and good coding style. This includes setting the right character encoding in your HTML.
    - MDN - HTML Metadata: https://goo.gl/PvpCNM

Example of metadata in HTML code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8"/>
    <meta http-equiv="X-UA-Compatible" content="IE=edge"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
```

Building a Web Site

Consider a “User-Centered” Approach. Goal Driven Design - what are your user’s goals?

User Experience (UX)

Refers to all aspects of an end-user’s interaction with your product.

> *“User Experience (UX) involves a person's behaviors, attitudes, and emotions about using a particular product, system or service.”* (“User Experience.” Wikipedia, The Free Encyclopedia. Wikimedia Foundation, Inc. 3 Jan 2015. Web. 27 Jan 2015. <http://en.wikipedia.org/wiki/User_experience>)

User Interface (UI)

Refers to the graphical elements of your website or application that they see and interact with on the screen. These are the visual elements such as buttons, form fields, etc.

Back End/server-side code DOES have a direct influence on the UX, even though it may not directly involve the UI elements. UX is everyone’s responsibility. Everything that everyone contributes must come together into a final product.

Interaction Design (IxD)

Adapted from UX Booth's Complete Beginner's Guide to Interaction Design: http://goo.gl/7jeg5j

IxD is a subset of UX. UX is ALL aspects of a product, whereas IxD focuses on human-computer interactions specifically. The Interaction Design Association explains:

> *"Interaction designers strive to create useful and usable products and services. Following the fundamental tenets of user-centered design, the practice of interaction design is grounded in an understanding of real users - their goals, tasks, experiences, needs, and wants. Approaching design from a user-centered perspective, while endeavoring to balance users' needs with business goals and technological capabilities, interaction designers provide solutions to complex design challenges, and define new and evolving interactive products and services.*

Information Architecture (IA)

This is the backbone of your site. How are your files, pages, directories (folders), and assets organized on the server? Is the structure SCALABLE? Is it user friendly for you, your client, and the users? (Keep in mind, the server side of a web site DOES affect the front end). How will your URLs look? How will your navigation be organized?

Going Deeper: Information Architecture for DATABASES is referred to as Data Design and Database Schemas. This is specifically how a database is organized and constructed, and is an area of specialty!

Sitemaps, Decision Trees, Use Cases, Wireframes

Persona

A hypothetical model for a user, based upon your intended target audience or a segment thereof. A project can have many personas - but limit it. I read one recommendation for one persona for each major segment of your audience. 3-5.

Use Case

How does your persona interact with your site? How, when, and where? For what purpose? What are the user/persona's goals? What steps must they take to complete this goal? Map this out in a diagram.

Decision Tree

A graphical illustration of possible decisions and outcomes. This helps to map out strategies and paths to goal completion.

Sitemap

A sitemap is a schematic representing the structure of your website or app. Hierarchical structure/taxonomy of the content. How are you going to group your content?

Additional Reading:

- Usability.gov - User Experience Basics: http://goo.gl/vSNNDV
- UX Booth's Complete Beginner's Guide to Interaction Design: http://goo.gl/7jeg5j

Tools of the Trade

What Languages Should I Use?

- Count on HTML and CSS
- JavaScript is ubiquitous, and is being used increasingly for back-end functionality (Node.js and the MEAN stack). If you'll be sending/retrieving data via AJAX, count on JavaScript.
- Your choice of Back End languages can be affected by a CMS and other technologies. For example, Wordpress sites are built using PHP and MySQL databases.

Tools

Minimum:

- A text editor at a MINIMUM. (Atom, SublimeText, Notepad++)
- An FTP client to transfer files to/from your web host. (Filezilla, etc.)
- A stack of current web browsers: Chrome, Firefox, Safari, IE, iOS, Android
- Chrome and/or Firefox Developer Tools (included with the browsers)
  - Chrome DevTools Documentation: https://developer.chrome.com/devtools
- Optional, but nice: Graphics Editor (GIMP, InkScape, Photoshop, Illustrator, etc.)
- Web Hosting and Domain Name(s)
  - GoDaddy, A Small Orange, BlueHost, HostGator, etc.

Going Further:

- An IDE for more development power. (PhpStorm, Eclipse,)
- A local server configuration on your computer. (Such as WAMP or XAMPP)
- Git and GitHub for version control (We have an upcoming class on that!)
- ...The sky's the limit!

A non-comprehensive list of tools I personally enjoy (front-end focus):
https://github.com/rlewis2892/favorite-frontend-tools