

Chapter 13

Code

Writing proper, valid HTML, and any code for that matter, is vital to ensure that the pages, the site, and/or the applications being developed work successfully and will continue to work successfully over time. To make sure that your code is properly formed, several approaches can be used. The two most common approaches are validation services to check your Web site, or development tools to check your code for accuracy.

Several validation services have opened shop on the Internet to check Web sites for code quality issues. These validation services will check HTML for errors and provide a report. Some services are free, while others charge for the service. Alternatively, many popular HTML authoring tools now incorporate built-in features that allow coders to check their work. One such tool is Adobe's Dreamweaver. By using the "Check Target Browser" feature, designers can specify which browser they would like to target in their site design. This browser is then opened and HTML is displayed. Dreamweaver then checks to see if the HTML is appropriate for the browser specified. Valid code is a necessary node that must be considered when mapping the anatomy of sustainable Web ecosystem design.

Suggested Web Resource(s):

W3C HTML Validation Service: <http://validator.w3.org>

13.1 Stability Through Standards

Validating your code will ensure stability over time. To work within the proper and established Web standards helps to this end. The W3C or World Wide Web Consortium is the Web standards leader as an international community that develops open standards to ensure the long-term growth of the Web. The W3C standards "define an Open Web Platform for application development that has the unprecedented potential to enable developers to build rich interactive experiences,

powered by vast data stores, that are available on any device.” (W3.org) Incorporating this stability into your Web projects is a necessary node that must be considered when mapping the anatomy of sustainable Web ecosystem design.

Suggested Web Resource(s):

W3C Web Standards: <http://www.w3.org/standards>.

13.2 Lightweight

It is always important when creating Web content of all kinds to consider the “weight” of the material you are putting on your server. Another way of thinking about the weight is the term bandwidth. The total weight of your files, or the bandwidth, is the size of the files along with the types of files, which take a certain amount of time to transfer from the Web server to the users screen. For example, say you are sharing a weekend trip you took with some friends over the Web. If you have Web page with four jpgs, one mpg, and some text explaining your adventure, you would essentially have six elements that need to download from your Web server to each of your friends computers or devices when they went to view your material. You would have

1 html file + 4 images + 1 video = 6 total elements.

Incidentally, these six files would show up as six independent “hits” in your analytics software. To optimize this experience for your friends, you would want to make this site as lightweight as possible. In other words you would want to use the least amount of bandwidth that would be possible while maintaining the highest quality experience you can offer to your friends and anyone else who may see your site. There are many ways of going about this optimization process from streaming audio and video media instead of downloading to using media queries to server up the best content based on users bandwidth. Lightweight Web experiences for all of your users are a necessary node that must be considered when mapping the anatomy of sustainable Web ecosystem design.

Suggested Web Resource(s):

Google Developers Speed: <https://developers.google.com/speed>

Steve Sounders, Head Performance Engineer at Google: <http://www.stevesouders.com/blog/2012/02/10/the-performance-golden-rule>

Suggested Book(s):

High Performance Web Sites: Essential Knowledge for Front-End Engineers, by Steve Sounders, O’Reilly Media.

13.3 Secure

Security is always a top concern in anything you do on the Web. Taking smart, tested steps to ensure your code for your project is safe and secure is a necessary node that must be considered when mapping the anatomy of sustainable Web ecosystem design. The issue of Web security is a living issue, which means, like most of the Web, it is constantly evolving, changing, and being updated for any number of reasons. Technologies improve, risks increase, and, thankfully, methods of securing our sites also improve. There is no way to sum up the topic of Web code security in one simple missive. The best thing you can do to make your sites and your code safe and secure is to start learning now, and make it a habit of checking several of the top sites that offer updated methodologies on Web security.

Suggested Web Resource(s):

Mozilla Guidelines: https://wiki.mozilla.org/WebAppSec/Secure_Coding_Guidelines

Google Code: <http://code.google.com/edu/security/index.html>¹

Google Developers: <https://developers.google.com>

13.4 Open

The term open source describes practices in production and development that promote access to the end product's source materials. Some consider open source a philosophy, others consider it a pragmatic methodology. Before the term open source became widely adopted, developers and producers used a variety of phrases to describe the concept; open source gained hold with the rise of the Internet, and the attendant need for massive retooling of the computing source code. Opening the source code enabled a self-enhancing diversity of production models, communication paths, and interactive communities. Subsequently, the new phrase "open-source software" was born to describe the environment that the new copyright, licensing, domain, and consumer issues created. The open source model includes the concept of concurrent yet different agendas and differing approaches in production, in contrast with more centralized models of development such as those typically used in commercial software companies. A main principle and practice of open source software development is peer production by bartering and collaboration, with the end-product, source-material, "blueprints," and documentation available at no cost to the public.

¹ Google Code is currently in transition. Other useful resources here are <https://developers.google.com> and <https://developers.google.com/university>.

The philosophy and development model of open source is a method for creating software that “harnesses the power of distributed peer review and transparency of process” with the goal of “better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in.”

The Open Source Initiative, one leading open source advocate organizations is involved in creating a “standards body, maintaining the Open Source Definition for the good of the community.” Open source goes way beyond just making your code available for others to use, and it is a necessary node that must be considered when mapping the anatomy of sustainable Web ecosystem design.

Suggested Web Resource(s):

Open Source Initiative (OSI): <http://opensource.org>

Open Source Definition: <http://opensource.org/docs/osd>

Open Source.com: <http://opensource.com/resources/what-open-source>.

13.4.1 Semantic

The basic meaning of the word “semantic” is “meaning”. What we mean when we talk about the Semantic Web is that we give objects meaning so that not just people can understand the information that is the Web, but so that computers can also understand the information that is the Web. To get a historical perspective here we can look at the dominant characteristics of the Web over time (Table 13.1).

The semantic standard is largely encouraged by the W3C. The standard promotes common data formats on the World Wide Web. There are several widely used data formats for semantic code.

Microformats are simple ways to add information to a web page using mostly the class attribute (although sometimes the id, title, rel orrev attributes too). The class names are semantically rich and describe the data they encapsulate.

With RDFa (Resource Description Framework in attributes), a W3C recommendation, you can easily include extra “structure” in your (X)HTML to indicate a calendar event, contact information, a document license, etc. RDFa is about total publisher control: you choose which attributes to use, which to reuse from other sites, and how to evolve, over time, the meaning of these attributes.

The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

The Dublin Core Metadata Initiative, or “DCMI,” is an open organization engaged in the development of interoperable metadata standards that support a broad range of purposes and business models. DCMI’s activities include work on

Table 13.1 The dominant characteristics of the “versions” of the Web to date

Web “version”	Approximate years	Traits
Web 1.0	1994–1999	Code heavy Static Bottlenecks Webmasters only
Web 2.0	2000–2008	User generated content Tags Dynamic, data driven
Web 3.0	2009–present	Semantic Usability/user psychology Software as service Cloud computing Mobility/ubiquitous Augmented reality Micro-payments Open source CMS

architecture and modeling, discussions and collaborative work in DCMI Communities and DCMI Task Groups, annual conferences and workshops, standards liaison, and educational efforts to promote widespread acceptance of metadata standards and practices.

Coding for semantic understanding is a necessary node that must be considered when mapping the anatomy of sustainable Web ecosystem design.

Suggested Web Resource(s):

Microformats: <http://microformats.org>

RDFa: <http://rdfa.info>

W3C Semantic Web Ontology Language OWL: <http://www.w3.org/TR/owl-features>

Dublin Core Metadata Initiative: <http://dublincore.org>.

13.5 Interoperable

For our purposes here, the quality of interoperability is considered to be a type of compatibility. “Interoperability is a property of a product or system, whose interfaces are completely understood, to work with other products or systems, present or future, without any restricted access or implementation” (Wikipedia). The IEEE defines interoperable as “the ability of two or more systems or components to exchange information and to use the information that has been

exchanged.” This efficiency of code and all other Web elements is a necessary node that must be considered when mapping the anatomy of sustainable Web ecosystem design.

13.6 Energy Utilization

The W3C hosts a Web Protocols and Energy Utilization Community Group. The purpose of the group is to create a space for brainstorming and collaborating on the problem of lowering the necessary energy consumption of Web protocols, that is, the data transferring mechanisms that make the Web possible. At present there is not much activity within the forums, however there are many great initiatives, ideas, and blog posts on the Web that are starting to address and come up with solutions for Internet energy consumption. This consideration of energy efficiency whether it is for the good of the environment or the good of the user experience is a necessary node that must be considered when mapping the anatomy of sustainable Web ecosystem design.

Suggested Web Resource(s):

W3C Web Protocols and Energy Utilization: <http://www.w3.org/community/webprotoenergy>

W3C WPEUG Willy Svenningsson Partial Serving of XHTML demo: <http://www.deciweb.se/SuggestionsW3C>

Net Magazine article by Pete Markiewicz: <http://www.netmagazine.com/features/save-planet-through-sustainable-web-design>

Pete Markiewicz, Green Resources, Energy Use Optimization: <http://sustainablevirtualdesign.wordpress.com>.