
Web Development Overview

I. Learning Objectives:

By the end of this lesson, you will be able to:

1. Describe what the purpose of a web server and web client are
2. Breakdown a web URL and describe what the 3 parts of the URL mean
3. Differentiate between absolute links and relative links.
4. Explain the differences between web 1.0, web 2.0 and web 3.0
5. Explain how front-end web programming differs from back-end.
6. Describe what W3C is and how it affects web development?
7. Describe the concept behind progressive enhancement (explain where HTML, CSS and JavaScript fit in)
8. List and describe software tools used to create and publish web documents.

II. How Web Pages are Displayed

A. Servers, Clients and URLs

A web server is a computer that stores web pages and responds to users who request retrieval of a page the server stores.

A client is a computer that requests a page from the web server.

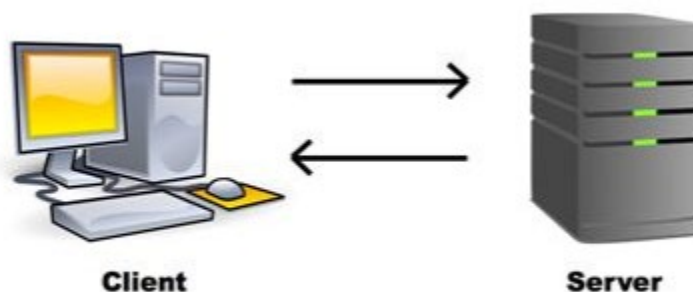


Figure 1-1: Clients and Servers

Retrieved from HTML 5 for Masterminds, August 23rd 2017

Web servers are organized just like your hard drive at home; they have folders that contain files and additional folders.

To view a page, you need to know the address or name of the computer storing the page, the folder(s) the page is in and the name of the page. The web address is called a URL (uniform resource locator)

A URL has 3 parts: protocol, domain and resource (or path)

Example:

https://www.example.com/contact.html

The diagram shows the URL **https://www.example.com/contact.html** with three horizontal lines below it. The first line is under 'https://', the second is under 'www.example.com', and the third is under '/contact.html'. Below each line is a label: 'Protocol', 'Domain', and 'Resource' respectively.

A) Protocol

A protocol is a set of rules that machines follow to complete tasks. In a web address, the protocol is displayed at the beginning of the address, and it indicates what type of page we will be viewing. That tells the web browser how to interpret and display the page content. The https and http protocol are used for web pages (http stands for **h**ypertext **t**ransfer **p**rotocol and the s on the end indicates it is secure).

B) Domain

The domain is a unique identifier for the computer storing web resources. Every web server has a unique domain name and a unique numeric address. The numeric address is in the form nnn.nnn.nnn.nnn where nnn is a number between 0 and 255. This address is called an IP (Internet Protocol) address.

How does the Internet know which text domain goes with a specific numeric IP address? There is a special server called a DNS (domain name server) that associates numeric IP's with their text equivalents. The DNS server manages a massive database that maps domain names to IP addresses. Your computer uses a DNS server to look up the numeric IP for the address you enter in the web browser.

Could you enter the numeric IP instead of the text domain name? The answer is yes; however, it is much easier to remember and enter the text domain than a very long string of numbers.

C) Resource

The resource includes folders you must open to get to the actual file you want to display. Folders are separated by / and the last part of the resource is the page you want to display. Web pages end in .htm or .html

Example #1:

<https://lisabalbach.com/index.html>

- The protocol is `https://`
- The domain is `lisabalbach.com`
- The resource is `index.html`

By default, servers will automatically include `https://` in the web address, so you do not need to type that portion into the address bar. If you omit the resource, most servers will automatically look for `index.html`. For Example #1, you could simply type: `lisabalbach.com` and the index page will display.

Example #2:

<https://lisabalbach.wordpress.com/digital-artwork-and-fine-art-photography/mixed-media/>

- The protocol is `https://`
- The domain is `lisabalbach.wordpress.com`
- The resource includes folders on the server and the name of the page displayed. Since the address doesn't include the name of the page, you can assume it is `index.html`. The address indicates there are 2 folders on the server: `digital-artwork-and-fine-art-photography` is one folder and inside that folder, there is another folder named `mixed-media`. In `index.html` page is located inside the `mixed-media` folder. NOTE: Folders are technically called Directories. If you look at the web server itself, what you will see are folders similar to the folders you use and create on your PC. Each folder on the server represents a directory.

Example #3:

https://www.w3schools.com/html/html_basic.asp

- The protocol is `https://`
- The domain is www.w3schools.com
- The name of the page is `html_basic.asp`. That page is located inside the `html` folder (i.e. directory)

You will notice the page in example 3 doesn't end in `htm` or `html`. That is because the page in example 3 was created using ASP.NET. Different web languages will place their own extension on the file.

B. Hypertext Links

In a web page, a link takes you to a different location within the page or to a completely different page.

Links can be created using the full URL or only the resource portion of the URL. Links that use the full URL are called **absolute links**. Links that only use the resource portion of the URL are called **relative links**.

Many students aren't sure of when absolute links should be used and when relative links should be used. We will be covering this in detail in a few weeks. The following analogy may help you understand this a little better. Let's say you want to navigate from your home to NMC. To get to NMC, you will follow the full set of directions beginning from your home. The directions are similar to an absolute link because you are getting the full set of directions (absolute links include the full URL).

Let's say that you go to Rite Aid and want to navigate from Rite Aid to NMC (you are no longer starting from home). You would want to use directions "relative" to your current location rather than directions that start from your home address. This is what relative links do, they look at the current location and navigate to the new location based on where you are currently at. These links do NOT use the full URL, they typically include the resource and omit the protocol and domain.

III. HTML and the World Wide Web

HTML stands for Hypertext Markup Language. HTML documents came about when the World Wide Web was developed in 1989-1990. They allow graphics, hyperlinks and multi-media to display at a web site. (Prior to the development of the World Wide Web, Internet documents didn't contain these features; they were text documents that were read from top to bottom).

The HTML language inserts codes around text to describe what the text does within a document. For example, if you wanted your first name boldfaced, you would insert `<strong>Your name here</strong>` into the document. The `<strong>` and `</strong>` are HTML elements that indicate the text within them should be boldfaced.

Each web browser has default settings for font, size, background color, links etc. Default settings are used:

1. If the author of a web page doesn't specify how a page should look
2. If the author of a web page specifies fonts, colors etc. that aren't available on the computer displaying the page

Different browsers will have different default settings (that is why documents displayed in Chrome or FireFox may look different than documents displayed on Microsoft Edge).

Default settings give anyone using a web browser the ability to display pages created in HTML. The use of defaults has given HTML two advantages over other languages:

1. **Portability** across computer platforms (it doesn't matter what type of machine you use to access the Internet, as long as you have a web browser installed, you can bring up a web page).
2. You don't have to give specific instructions to see a page. This results in faster **retrieval speed** (the more codes you insert, the longer the page takes to display).

The downside to using defaults is the lack of display consistency among browsers. Most web authors will check pages on more than one browser once it is loaded onto the Internet to ensure that it displays properly.

A. Evolution of the Web

Web 1.0 was characterized by individual publishing, one-way communication, passive involvement, read-only content and personal websites.

Web 2.0 has been called the collaborative or collective web because it allowed for collaboration among groups, group participation, two-way communication, active involvement, user-generated content and blogging. Web 2.0 encompasses everything we currently do to participate in websites (social media, wikis, blogs and that type of thing). It also includes websites that bring in information from multiple sources to generate a hybrid page or mashup (google maps is an example). Web 2.0 has given the viewer the power over the information presented.

The migration from web 1.0 to web 2.0 has been gradual. We are currently in a phase labeled web 3.0. **Web 3.0 is called the semantic web.** It makes searching the web smarter. For example, if you wanted to find a funny movie near a Mexican restaurant, you could ask a web browser a complicated question like: *Where is the closest movie theater playing a funny movie that is next to a Mexican restaurant?* The search engine will analyze your question and come up with the best option. In web 3.0, browsers tailor results based on your behavior. They also look at your online connections, content sharing, influencers, how you make connections etc. The result is the content you see is more relevant and searching the web is easier.

Web 4.0 is the next phase. It is being called the **symbiotic web** and looks to combine AI (artificial intelligence) and human interaction. Its focus is on mobility and mobile device usage which allow users to continuously add, share and distribute information into cyberspace. Technologies for this phase are currently being developed. It will be exciting to see where these new technologies lead us.

B. HTML Standards and HTML extensions

Standards ensure that web pages can be viewed in different web browsers. When web page development first started, there weren't any standards. Different web browsers allowed programmers to create and use HTML extensions to give them additional features not offered in regular HTML. The problem with extensions is they are browser specific, so if they are Internet Explorer extensions, the extensions would only work in the Internet Explorer.

To help solve this problem, a group of web authors, software companies, and users got together and formed a consortium known as W3C (or World Wide Web Consortium). The consortium developed standards for HTML and now develops standards for CSS, JavaScript and almost all web programming languages. For more information on W3C, visit <http://www.w3.org>

C. HTML & XML

Both HTML and XML structure pages using tags or elements.

The purpose of each type of page is very different:

- XML was designed to transport and store data. It is a data storage language.
- HTML was designed to provide structure for web pages and to display data.

XML is more flexible than HTML because the programmer makes up the elements or tags used in the pages (the elements aren't predefined like they are in HTML). Before you create and use XML, you should understand how HTML works.

After learning HTML, you will learn JavaScript which is used to create, update, and retrieve XML files. JavaScript can also add the content of XML files to existing HTML web pages.

D. HTML 5.0 and the Semantic Web

HTML 5.0 is the current standard and is paving the way for the semantic web (web 3.0 and web 4.0). With HTML 5.0, there are many new elements and features that version 4.0 didn't have. The 5.0 version has also eliminated older elements (these eliminated elements are being phased out).

When a feature is being phased out, the term used is **deprecated**. All deprecated means is that the feature is old and outdated and should no longer be used. The web browsers will still recognize it for a period of time, so pages that contain the code will still function; however, new pages should include the current standards and old pages should be converted over.

E. Front End vs Back End Web Programming

I'm sure you have heard the terms back-end web programming, front-end web programming, client-side web programming, server-side web programming etc. What they are referring to is the ability to make pages interactive. A back-end language runs on the server and does all its work before the page gets to you. Server-side web programming handles this. Once you can view the page on the web browser, any interactivity is done on the front-end which involves client-side web programming.

1) Front-End (client-side web programming)

The front-end is the part of the web that you can see and interact with. It refers to technology that is executed on the end user's computer by the web browser. In client-side programming, the browser understands the languages used and is able to interpret the code. HTML and CSS are part of the front-end.

HTML 5.0 and CSS 3 include some elements for interactivity; however, the interactivity is limited. To allow for a higher level of interactivity, you need to use a scripting language which is where JavaScript comes into play.

JavaScript is an object-oriented scripting language that runs inside a Web browser. It is the most popular and widely used client-side language. It is supported by all web browsers across the world. It manipulates elements on a Web page by means of a Document Object Model, or DOM. Learning how to work with the DOM is more complex, there are a lot of ways to interact with the DOM so the learning curve is a little steeper for JavaScript as opposed to other languages.

JQuery and AJAX are related to JavaScript. JQuery is a JavaScript library that takes common tasks requiring multiple lines of code and puts them into routines you can call with one line of code. So, JQuery makes a programmer's

life easier. AJAX stands for Asynchronous JavaScript and XML. The purpose of AJAX is to update parts of a page without reloading the page.

In addition to JavaScript and the JavaScript libraries, there are many popular frameworks such as React, Angular, Vue etc. At their most basic, JavaScript frameworks offer a more complete solution than a library because they provide a skeleton or framework that organizes the parts of your site where the framework is implemented. Libraries are used to solve a coding problem rather than implement a complete site.

Other client-side scripting languages include: Jscript and VBScript (Microsoft's front-end scripts that run on the ASP.NET framework), ActionScript (part of Adobe Flash), TypeScript, and any other scripting language that can be interpreted by a web browser.

FRONT-END DEVELOPMENT

Upwork™

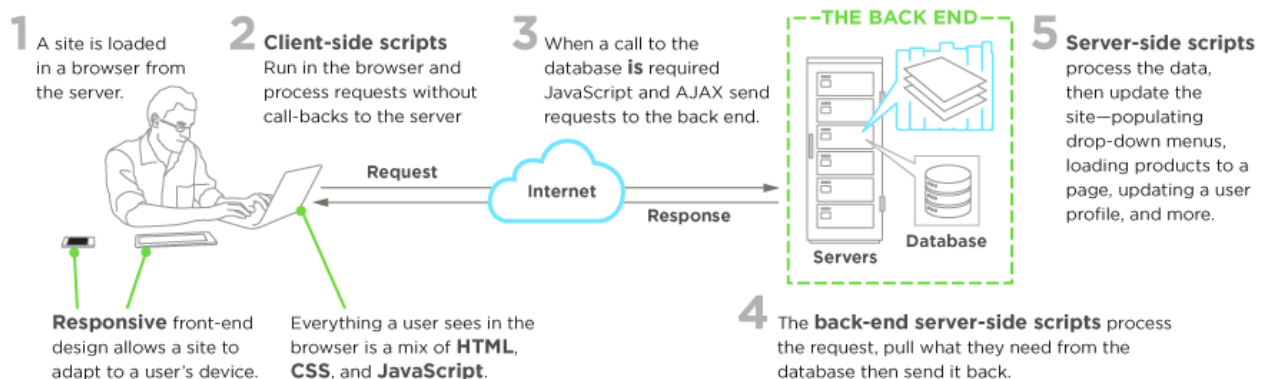


Figure 1 Front-end development

Retrieved from <https://www.upwork.com/hiring/development/how-scripting-languages-work/> on August 22, 2017

2) Back-End (server-side web programming)

Back-end or server-side programming are terms for programs that are run on the Internet server. They are used to process user input, display pages, structure web apps and store data.

The back-end usually consists of three parts: a server, an application, and a database or some type of data storage.

Back-end languages include PHP, Ruby, Python, Java, ASP.NET (Microsoft's Active Server Pages). When the scripts are processed, it is the server that processes them.

F. Progressive Enhancement (the big picture)

Progressive enhancement is the basis for front-end (or client-side) web programming. It is the separation of the content (HTML) from the presentation (CSS) and the behavior (JavaScript)

When developing content, you first focus on the HTML because it provides the structure for your page. Current standards advocate using semantic and meaningful tags to give the page structure. HTML pages can be displayed on the web without any additional coding (although they will not look very exciting).

To make the page look better after you have entered the HTML, you will add CSS to format the page and make it aesthetically pleasing. The CSS phase is all about page presentation. After you add the CSS, your web page should be fully functional AND it should look good too!

If you want to make the page interactive, you can add in JavaScript to control the behavior of elements within your page.

Ideally, each layer (structure, presentation, and behavior) is kept separate from the other layers. So, you would have an html page for the structure, an external CSS style sheet for the presentation and an external JavaScript file for the behavior.

The graphic below illustrates this concept:

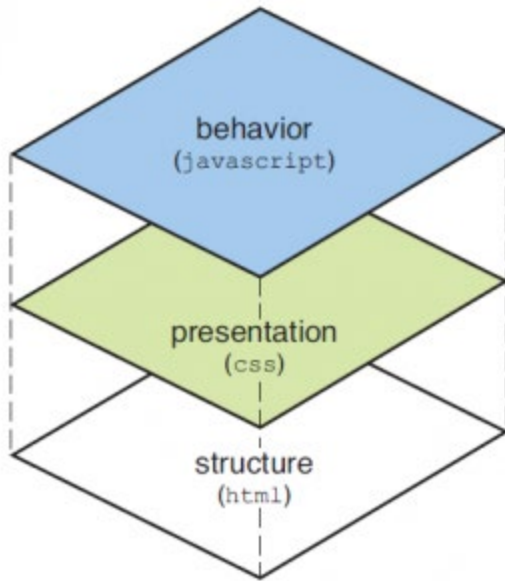


Figure 2 Progressive Enhancement

With web 3.0, there is also a fourth layer referred to as a resource description framework (RDF). RDF is a standard model for data interchange on the web. It has features that facilitate data merging and it extends the linking structure of the web to share data across different applications. It is supposed to add meaning to web content so the data can be identified, and the computer can make decisions based upon the users' previous preferences. An example of this technology is ads for products you have recently viewed appearing on Facebook or on other websites. The browser shows content based upon your preferences.

IV. Software for Creating HTML pages

To create a web page, you need to use some type of text editor or a web editing program. You also need access to a server and a program to transfer the files from your computer to the server.

1. Text Editors

Text editors don't insert extra code into your web pages. They only use what you type in. The text editor that comes with Windows is Notepad. The text editor that comes with Mac's is TextEdit. You can use either of these to create your web pages. DO NOT use word processing software to create web pages because they will insert a lot of extra codes into your html (these extra codes are confusing, and they are not necessary).

2. Web Editing Applications

There are a wide variety of excellent web editing applications on the market, and many are free.

At NMC, we have adopted VS Code (which is an industry standard in the Developer field). All video demonstrations and examples will be done using this application.

To see a list and description of some additional web editing programs, view:

- <https://www.creativebloq.com/advice/best-code-editors>
- <https://www.lifewire.com/best-free-html-editors-for-windows-3471313>
- <https://www.hostinger.com/tutorials/best-html-editors>

3. File Transfer Programs

A file transfer program copies your web content from your computer to the server you connect to. We will be using FileZilla in this course. The program is cross-platform and very easy to use. Best of all, it is FREE!

In upcoming weeks, you will be using VS Code to create pages and FileZilla to transfer them