



Lars Gunther is a web developer, computer science teacher and a pastor, who lives in Trollhättan, Sweden. He is the lead editor of several courses for WaSP Interact and invited expert to develop Web technology related courses for Skolverket, the national agency for education in Sweden. When he likes to relax from theology or teaching, he usually reads about history. He is married to Penilla and bonus dad to two young adults, Alexander and Oliver.

<http://kenyx.se/>

<http://twitter.com/itpastorn/>

<http://itpastorn.blogspot.com/>

# CHAPTER 4

# Internet Basics

by **Lars Gunther**

When you develop for the Web it is imperative to know a bit about the Web, such as where it came from and how it works. Too many designers still think that web design is exactly like print design or that the Web only is about people browsing on a desktop or laptop computer. This chapter will provide some historical context and an educated guess about the near future of the Web and the technology it is built on. (What is this thing called the Internet anyway?)

It will also look at the terminology about web servers, web services, and the devices we use to participate. But the Web is more than technology. It is shaping our society, culture, and daily lives. Indeed, it is already interwoven with our lives in more ways that we might realize. We must take a good look at the sociological and ethical challenges we face as developers, as well as the possibilities.

Protocols on the Internet can be grouped according to their purpose in a layered protocol stack (**Table 4.1**).

Layer	Protocols
Application layer protocols	HTTP, FTP, POP, IMAP, SSH, Telnet, IRC
Transport layer protocols	TCP, UDP
Internet layer protocols	IPv4, IPv6
(Data) Link layer	(Lots of stuff outside the scope of this book)
Physical layer	(Lots of stuff outside the scope of this book)

**Table 4.1:** *Protocols and Layers.*

There are few points to consider about Internet access for a web developer:

- While there are some people who are starting to enjoy the net through fibre, ultra-high-speed cable modems, and 4G telephony (being rolled out in Scandinavia as this is being written), there are still lots of people who do not have broadband; they access the net through dial-up modems or GSM telephony. Many countries, like the United States, still only have spotty 3G coverage. Not all ISPs offer their customers a fixed IP-address, and those that do may not allow you to set up your own web server (see next section).
- Most access technologies are highly asymmetrical; they have much higher download speeds than upload speeds.

For the last two reasons you probably would prefer to use the services of a web host, and not set up a server of your own.



**Tip:** Latency has historically not been an issue for web developers, but now that we are moving into web applications, even 3D games like Quake or WoW, every millisecond of latency will be significant. Gamers hate when there is “lag”!

Such files are stored in the web root directory, often `/var/www/html` on a Linux machine or `C:\Inetpub\wwwroot` on a Windows machine. Starting from the web root there usually is a 1:1 mapping of the URL-path and the file-path, when serving static content.

The most common HTTP-servers are:

- **Apache.** An open source solution that usually runs on Linux, but is available on most operating systems. There are many modules available to extend

Apache's basic capabilities. Roughly half of all websites run Apache.

- **Internet Information Server (IIS)** from Microsoft. Runs on Windows only and integrates nicely with other services on a Windows Server. IIS is still trailing Apache in available modules, but not that far behind any more. Roughly 30 percent of all websites run IIS.

## Common Gateway Interface (CGI) and Server Side Scripting

Some user actions, such as form submissions, have always required the server to run a small program (a script) before it can deliver an answer. During the early years of the web those scripts were often written in PERL.

During the later half of the 1990s many technologies evolved that made dynamic content an integral part of the web server. Today we usually call this **Server-side Scripting**.

Developing scripts and databases to run on servers is called **back-end development**—in contrast to **front-end development**, which focuses on HTML, CSS, the Document Object Model (DOM) and ECMAScript (JavaScript). But modern front-end web development often means producing templates for a **content management system (CMS)**, which means that although the developer perhaps does not need to know server-side scripting in depth, he or she needs to be aware of the concept and recognize the meta-languages that exist for templates. Some popular server-side scripting options, besides PERL, are covered in this section.

### PHP

PHP is the most popular scripting language for web servers. It is open source and runs on all common operating systems. Traditionally it is used on Linux, with Apache and MySQL as a database server, hence the acronym LAMP. PHP is used on many popular sites, including Facebook and Wikipedia. There are many popular content management systems built with PHP, including WordPress, Drupal, and Joomla.

### ASP.NET

ASP is an abbreviation of Active Server pages. .NET is a framework for

software development, made by Microsoft. ASP.NET is the usage of the .NET framework for web development. Developers can use C#, Visual basic or J# as a programming language.

## JavaServer Pages (JSP) and Java Servlets

This is the use of Java on web servers. Although there are few applications for entry-level content management, Java has a strong user base at the enterprise level.

## Python

Python is a language that is strong on Linux systems, that also can be used for web development. It is open source and can be both compiled and interpreted. Zope and Django are perhaps the most well known web applications written in Python.

## ColdFusion

Originally just a platform that allowed the mapping of database records to HTML pages, ColdFusion has grown to be an all-purpose platform and a full-fledged server side scripting language—ColdFusion Markup Language (CFML)—that resembles HTML in syntax. Server-side ActionScript can also be used, as well as CFScript. Both of these resemble JavaScript.

## Ruby on Rails (RoR)

Ruby is a programming language, very rarely used on the Web without the Rails framework. RoR is driving well-known sites like Twitter and Basecamp.

## JavaScript on the server

As JavaScript has matured and is no longer considered a “toy.” The use of it on the server as well as on the client is becoming more popular. In fact, it has been available for server-side usage since 1996, but it is only in recent years that JavaScript has been seen as a major player on the server.



































