Web Module 1:

How the Web Works

Introduction

In this first of a series of eight web development modules, we'll show you what makes the Internet and the World Wide Web tick.

Yes, yes: We know. You've been using the Internet and Web since you were five years old. You grew up on them, they're second nature to you. But, knowing how to use something correctly doesn't mean you understand how it works. For example, native English speakers are notoriously ignorant of the grammar and syntax rules that underlie their mother tongue.

So, stick around ... you just might learn something!

When you're done working through this module, you'll know:

- How the Internet works
- How the World Wide Web works
- How the client-server model works
- How to manage your RIT web account

The Internet

What is the Internet?

A global network of networks that connects billions of computing devices.

Who owns the Internet?

No one! No person, company, organization, government, etc.

Different companies around the world own some of the hardware that enables the Internet to work: cables, towers, etc. A few organizations preside over Internet standards. But none of these entities owns the Internet as a whole.

It's similar to the global phone system. Who owns it? Well, companies and countries own parts of it. But no one owns the global totality of it.

What are the two main components of the Internet?

Hardware

Internet hardware consists of two end points – clients and servers – and everything in-between: transmission cables, cell towers, satellites, routers, etc.

A client is a computer device that requests resources, web pages for example. A server is a computer device that stores and sends requested resources to clients. We'll talk more about the client-server model later in the module.

Protocols

All of this lovely Internet hardware would just sit there twiddlin' its thumbs without the second component of the Internet: protocols.

A protocol is a set of rules that computers and programs follow in order to perform specific tasks. Without universally accepted protocols, communication between Internet devices would be impossible.

Internet Protocols

The Internet uses many protocols to get the job done. Here are four of the biggies:

- FTP (File Transfer Protocol) – transfers large files across the Internet. These files are stored on FTP servers and are accessed by FTP programs (often called FTP clients), such as FileZilla. Most web browsers can also access files stored on FTP servers.
- Telnet – establishes a virtual terminal connection between a client and an Internet server, enabling users to send commands to the server.

  Telnet is what people in the early Internet days used to "speak" directly to servers and other remote computers. Some system admins still use Telnet to configure servers, though it's no longer the only or necessarily best choice.

- TCP/IP – manages the transfer of data across the Internet.

  TCP/IP wins the Key Internet Protocol title, hands down. It's actually a suite of 50+ protocols grouped into two main protocols: Transfer Control Protocol (TCP) and Internet Protocol (IP).

  The TCP side handles how data is sent across the Internet. It divides the data into a set of small packets, sends all of these packets to the specified destination, then reassembles the packets in the right order at the destination.

  The IP side uses addresses embedded in the data packets to ensure that the packets arrive at their specified destination.

A Guided Tour of an Internet Transmission

Let's say you fire up your browser and enter the URL (web address) of your favorite gaming site Playerz.com in the address bar. Here's a step-by-step account of what happens:

1. Your browser sends a request for the Playerz home page over your Internet connection to your Internet service provider (ISP). The ISP routes the request to a server further up the Internet chain. And so on, until the request eventually hits a domain name server (DNS).
2. The DNS server looks for a match for the playerz.com domain name you entered. If it finds a match, it redirects your request to the IP address of the playerz.com server. If not, it sends the request further up the chain to another domain name server with more information.
3. Assuming all goes well, your request eventually arrives at the playerz.com web server. This server responds by sending the requested home page file to your browser. Not as a single file, but broken up into a set of packets, each consisting of 1,000 to 1,500 bytes of data.

   The playerz.com web server gives each packet a header and a footer that report what's in the packet and how it fits with the other packets to create the entire file. Each packet travels back through the network to your browser.

   These packets don't necessarily take the same path. Instead, each takes "the path of least resistance" – whatever network route happens to have the least traffic at the moment the packet is sent.

   This is a very important (and cool!) feature of the the TCP/IP protocol. The fact that packets can take different paths home enables them to be routed around congested areas on the Internet. Even if big chunks of the Internet go down, packets can still travel from one place to another (though more slowly). This is exactly what the DOD had in mind when it created the Arpanet, grandpappy of the Internet: to enable communication over a network even if part of that network got taken down by hostiles.

4. When the Playerz packets arrive at your computer, the TCP/IP software it is running uses their header/footer information to splice them back together in the proper order, like putting together a jigsaw puzzle.

   The final result: The Playerz home page gets displayed in your browser.

The packet model is used for other kinds of data as well. When you send an e-mail, it gets broken into packets, journeys across the Internet, then gets reassembled at the destination. Phone calls over the Internet convert conversations into packets using VoIP: Voice-over-Internet Protocol. Netflix streaming movies? You guessed it: packets.

The World Wide Web

What is the World Wide Web?

A system of hypertext documents that are accessed via the Internet.

What is a hypertext document?

A hypertext document is a document that contains links to other documents. For example: a web page with links to other web pages.

Are the Internet and the World Wide Web the same thing?

No!

The Internet and the Web enjoy a whole-part relationship.

The Internet (whole) is a global network of networks.

The Web (part) is one of those networks.

Who owns the World Wide Web?

No one!

It's pretty much the same deal as with the Internet (see above). Individual files on the web (HTML docs, images, audio files, scripts, etc.) can be owned by some entity: person, company, organization. But no one owns the entirety of them, the whole Web.

What are the two main components of the Web?

HTML Documents

Commonly known as: web pages.

Protocols

Just like the Internet, the Web runs on a set of protocols.

Web Protocols

There are three Web protocols you should be familiar with:

- HTML (Hypertext Markup Language) – provides a logical structure for the elements of a web page (paragraphs, tables, links, etc.) by enclosing these elements in tags.

  There are plenty of supplemental languages that help determine the look and behavior of web pages: CSS, JavaScript, PHP, etc. But HTML is the foundation; without it, web pages would simply not exist. Hence its nickname: "the language of the Web."

  You'll learn gobs about HTML (and CSS + JavaScript) in the other seven Web modules.

- HTTP (Hypertext Transfer Protocol) – a set of rules that determines how Web data (text, images, audio, video, and other files) is transmitted over the Internet.

  For example: When you enter http://somesite.com/somepage.htm – or the short version, somesite.com/somepage.htm – in the address bar of your browser, it sends an HTTP command to the Web server (somesite.com) directing it to retrieve and transmit the requested Web page (somepage.htm).

- URL (Uniform Resource Locator) – enables HTML documents to point (link) to resources on the Web: other HTML docs, multimedia files (images, mp3s, movies), web apps, etc.

  You can think of a URL as the address of a Web resource.

The Client-Server Model

Back in the olden days of mainframes and minicomputers, processing power was at such a premium that users sat at terminals, sent requests for service to the computer (please perform this action, run this program), and then waited – sometimes for hours! – for a response back from the computer (here's your result, you're welcome).

This was the terminal-mainframe model. These days, processing power is so abundant that terminals and mainframes have become a rarity. But the essence of the terminal-mainframe approach is still around; we call it the client-server model.

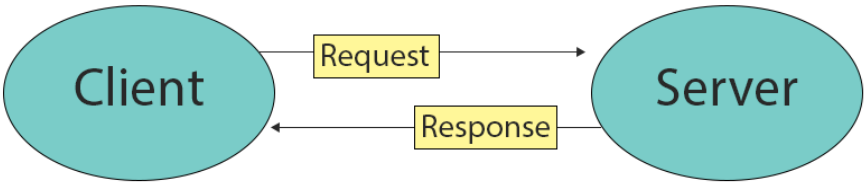
The client-server model describes a working relationship between two computing devices:

- Client – requests a service from a server (similar to the terminal)

  For example: May I please see a web page stored on you?

- Server – responds to a request from a client (similar to the mainframe)

  Sends client the web page. Or: sends client a message that the web page is gone. Etc.



Both the Internet and the World Wide Web use the client-server model to move data around the planet and get things done. For example:

> You wanna see what the weather has in store for you today. You fire up your Destler Weather Machine client app and click on the "Today's Weather" button. This tells your computing device (smart phone, tablet, etc.) to send a request for today's weather to the Destler Weather Machine server. The server responds by retrieving today's weather and then sending it back to your DWM client program, which displays it tidily on your screen.

The client-server model has become one of the mainstays of network computing in general, and the Internet and Web in particular. In fact, the main Internet protocol TCP/IP is built on the client-server model.

In the usual implementation of the client-server model, one server sits around waiting for requests from multiple clients. For example, RIT's Gibson web server awaits requests from web clients, which are typically browsers run by RIT students and faculty. When a request comes in, Gibson responds to it, then returns to its waiting state until the next request comes in.

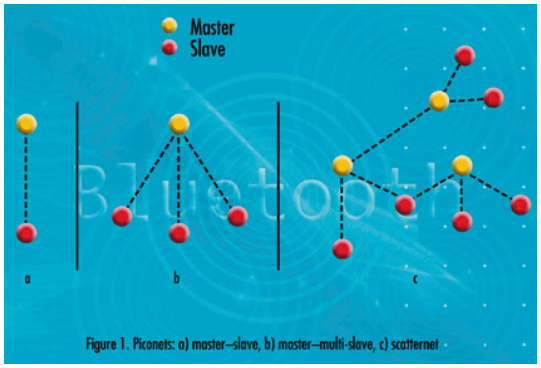
The client-server model is not the only model used for establishing network relationships. Here are two popular alternatives:

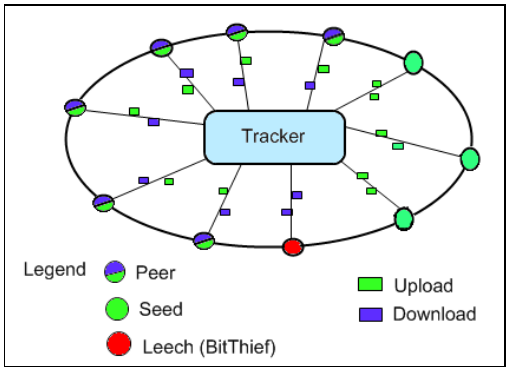
- Master-slave – the master program is in charge of all the slave programs

  Bluetooth is built on a master-slave architecture.

- Peer-to-peer – all peers are equally "in charge"

  Bittorrent is a peer-to-peer file-sharing protocol. (Heh heh heh.)





Setting Up your RIT Web Account

As an RIT student, you are entitled to have a computer account. Your account provides you with access to the Internet, e-mail, the myRIT portal, Student Information System (SIS), Wallace Library Online Databases, and other services.

If you don't have an RIT computer account – or are unsure whether you do or not – contact the ITS Help Desk: rit.edu/its/node/104

Within your RIT computer account you have a web account. In this account is a folder named www. All of your web files – HTML documents (web pages), images, audio files, etc. – must reside in this www folder or a subfolder within www.

If you upload an HTML file (web page) to your www folder, and set the file and folder's permissions correctly, anyone with a browser and an Internet connection should be able to view your page by entering a URL of this format:

> http://people.rit.edu/*username*/*pagename*
>
> *username* is your six-letter RIT user name and *pagename* is the name of the HTML file that you uploaded to the www folder. For example:
>
> http://people.rit.edu/rpsvks/mypage.html

Note that www does not appear in this URL, even though the www folder holds all your web files (pages, images, audio, etc.). Why? Because the RIT web server was configured that way. It's annoying to have to type in www every time you want to access someone's web page, so the network admins let you omit it.

Managing your RIT Web Account

To manage your RIT web account – upload files to it, create/rename/delete folders, set file and folder permissions – you need to use an FTP program like Fetch or Filezilla.

The procedures for doing these management tasks are presented in Getting Your Pages on the Web section of the HTML, Part II. But, to give you a feel for how these things work, we're ending this module with a Sneak Preview Practice task.

Gibson and UNIX

Gibson, the web server that hosts your RIT web account, is a UNIX server, which means that it runs under the UNIX operating system.

Why should you care? Because there's a good chance you'll hear hear people refer to Gibson as a UNIX server, and now you'll know what they're talking about.

Here are a few things you should know about the UNIX operating system:

- UNIX calls folders *directories*. But, to keep things simple, we'll refer to them as folders.
- UNIX treats everything in your web account as a file, including folders.
- A UNIX path specifies the succession of folders that leads to a target folder or file (shown here in red). For example:

  people.rit.edu/rpsvks/blog/2014 specifies the succession of folders that leads from the people.rit.edu account to the target directory 2014: rpsvks → blog → 2014

  people.rit.edu/rpsvks/blog/2014/january.html specifies the folders that lead from people.rit.edu to the target file january.html: rpsvks → blog → 2014 → january.html
- UNIX is case-sensitive, which means that your website visitors need to get the capitalization of your web folder and file names exactly right: /music/index.htm rather than /Music/index.htm, or /music/Index.html, etc. This, in turn, means that you should keep your folder and file names as simple as possible by using all lowercase letters.
- You can access a UNIX system, such as your Gibson web account, by entering text commands in a command-line interface. To do this, you need to Telnet into Gibson.

  Telnetting into Gibson is beyond the scope of this module. If you're interested, here's a good place to get started: rit.edu/its/services/desktop_support/mac/telnet.html

Practice: Using an FTP Program to Manage Your Web Files and Folders

1. Launch a text editor such as Notepad+ (Windows) or BBEdit (Mac). Don't use a word processor like Word or this procedure won't work.

   Create a new text file and type or copy/paste this HTML code into the file:

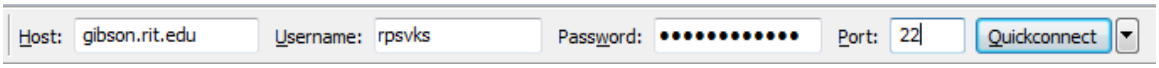
   ```
   <html>
   <head>
   <title>My Home Page</title>
   </head>
   <body>
   <h1>This is my home page!</h1>
   </body>
   </html>
   ```

   Save the file as a plain-text file with the name index.html – this is an HTML document (web page) that you'll soon upload to your RIT Gibson web account.

2. Launch your FTP program.

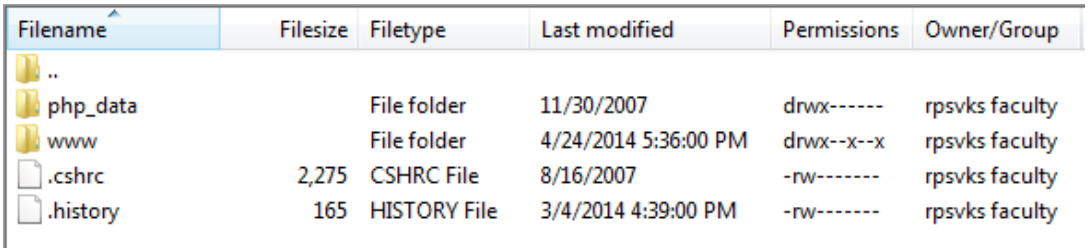
   We're using Filezilla. If you're using a different FTP program, your UI details will be a bit different than ours, but they should be similar enough so you can follow along.

3. Enter the following information:
   - Host – gibson.rit.edu (Gibson is the RIT server that hosts student/faculty web accounts)
   - Username – your six-letter RIT user name
   - Password – your RIT password
   - Port – 22 (this is the port number for **secure** FTP connections – technically SFTP)



4. Click on Quickconnect to establish an FTP connection with your Gibson web account.

   Once the connection is established, a listing of the files and folders in your Gibson web account is displayed in the right-hand pane of Filezilla.

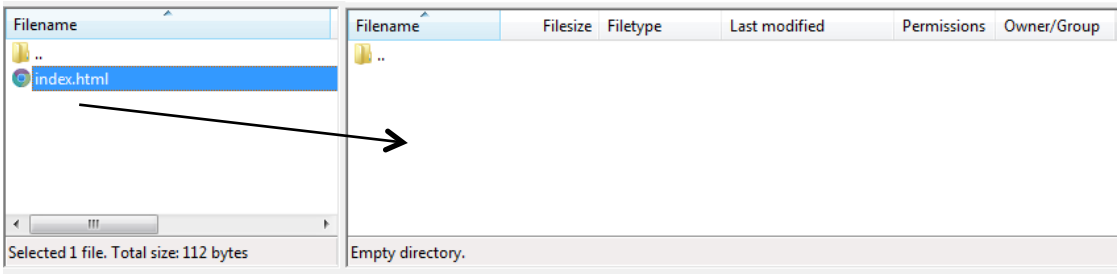


5. Find the www folder in your Gibson listing, and double-click it to open it.

   You are now in the area of your web account where all your web files need to be stored: HTML docs, images, etc. If you haven't yet uploaded any files to www, it will be empty.

6. Find the index.html file you created in step 1. It will appear In the left-hand pane of Filezilla, which lists the files and folders on your local computer.

   Drag index.html from your local computer to your Gibson www folder.

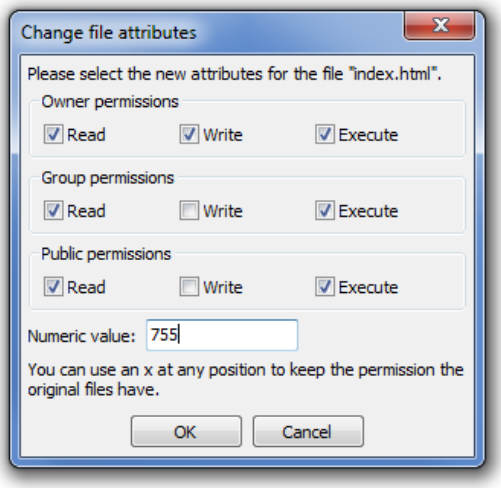


7. Congratulations! You've uploaded a web page to your Gibson web account! Try to view it in a browser by entering the URL (remember to replace *username* with your RIT user name):

   people.rit.edu/*username*/index.html

   It doesn't work. Why not? Because ... you didn't set the correct file and folder permissions to make your web page accessible to the world. To do this:

   Right-click on the index.html file in your Gibson account (right pane), and choose File Permissions. In the File Permissions dialog box, enter a numeric value of 755. (If your FTP program doesn't support numeric permission values, select the Owner, Group, and Public permissions shown below.)



8. That takes care of the permissions for your index.html file. But your www folder permissions also need setting.

   Navigate up one level so that your Gibson www folder icon is visible, and repeat the procedure you used to set the index.html permissions.

9. Now your index.html web page should be visible by entering this URL in a browser:

   people.rit.edu/*username*/index.html

10. Before you call it quits, you need to practice creating folder paths in your web account.

    A path, as you'll recall, is a succession of folders, each stored inside the previous one.

    Your mission is to create the following paths in your Gibson web account. The first folder in each path should reside in your www folder.

    /images (images folder resides in www)

    /work/miscellany (work folder resides in www, miscellany folder resides in work)

    /modules/web/notes (same deal)

    /myportfolio/media/sources/europe (same deal)

    When you're done, give each of the 10 folders in these paths the permissions necessary to make them accessible to web users. (See step 7.)

    This is how you create folder paths in which to logically store your web files.