

# Internet Addresses

The [Internet](#) is a vast network of many different computers that are able to talk to each other in spite of the fact that they may be separated by large distances, may be very different computers built by different manufacturers, and may be running many different kinds of operating systems. This is similar to people in (say) Israel, Brazil, Japan, the United States, and Norway, each normally speaking a different language, being able to converse almost instantaneously with each other on very broad ranges of topics!

The computers of the Internet exchange packets of information using addresses that have much in common with ordinary mail addresses. These addresses are called Internet Protocol (IP) addresses.

# TCP/IP Protocols

How is such efficient communication possible? Well, there are two big advantages that the computers on the Internet have over our hypothetical natives of different cultures and countries. The first is that the computers of the Internet are in reasonably constant contact with each other over a set of telephone lines, transoceanic cables, satellite links, and so on that allow rapid transfer of electronic information. The second is that the computers of the Internet have agreed to a common set of understandings that allows them to exchange vast amounts of information even though internally they may speak very different computer languages.

These communications are possible because of a set of *protocols*, which is a fancy word for mutually agreed-upon rules about exactly how information will be transferred between the computers of the Internet. The two most important protocols that allow networks to cooperate with one another and exchange information are called TCP (*Transmission Control Protocol*) and IP (*Internet Protocol*). Collectively, these two protocols are often referred to as TCP/IP.

Local networks can be based on a variety of technologies, but as long as the software layers on top of the network hardware implement the TCP/IP protocol operations, a network can exchange intelligible information with the rest of the Internet.

## Packets of Information

How does information get transferred between computers on the Internet? A pretty good analogy is that of letters and the usual mail system (the one that computer types call "snail mail", because it is so slow compared with electronic mail!). The information is transferred in what are called electronic *packets*, which are something like letters containing information. If the packets are like letters, they must have an *address*. Just as normal letters must have an address on the front to make delivery likely, TCP/IP communication depends on addresses being included in each packet. Not surprisingly, these addresses are commonly termed "IP addresses". So, if someone asks what your IP address is, she is asking the electronic equivalent of what your post office address is.

As these packets of information move through the network, electronic devices called *routers* use these IP addresses to decide whether to keep a packet in a local network or to forward it to a different network. This is a complex task, because there are many networks that either comprise the Internet, or that are attached to it through some *gateway*.

Thus, routers and gateways in the network serve much the same function that different levels of post offices serve in the usual mail system, with each level of the postal system deciding whether the letter in question should be delivered locally, or forwarded to another part of the postal system.

## Addresses

The address on a normal letter typically has several lines, each of which supplies different levels of information about the exact location corresponding to the address. In a very similar way, addresses for Internet information packets have four fields that contain numbers and are separated by periods. For example,

160.36.28.37

is an IP address. However, people generally can remember names better than numbers, so it is convenient to associate a name with such an IP address. In this case, the corresponding name is

csep10.phys.utk.edu.

The translation between the numbers used by the network, and the name more commonly used by people is done by a computer called a *nameserver*. The purpose of a nameserver is to look up addresses, so its function may be likened to directory assistance on the telephone system.

If I specify `csep10.phys.utk.edu` as an IP address to a network, the first thing that it does is call its "directory assistance" - that is, a nameserver - to get the number associated with that name. (Actually, most computers have a small list of address-name mappings stored locally that they first consult, and only go to an external nameserver if they don't find the address there.) The network uses only the number, but by this method the humans and the computers are both kept happy: the computers in the network just have to deal with the numbers, which is what they do best, and the humans using the computers need only remember more symbolic names like

mycomputer.myplace.mynet.mycountry

which is what humans do better. Of course, the humans can also supply the IP address directly to the network as a number. Then, the nameserver isn't invoked. In fact, sometimes the nameservers, since they are machines, are down for some reason and one **MUST** supply the number to the network. However, that doesn't happen too often these days on modern and well maintained networks because there usually is more than one nameserver available, and if one is down another can often be found by the network when it needs a name translated.

[NEXT ►](#)

## Internet Addresses | The Naming System

# The Naming System

Let us consider as a typical example the internet address

`www.techcorps.org`

which is actually the name of the IP address `160.36.28.57`. How did the people who run the Internet come up with that name, and how did we know that `www.techcorps.org` is really a nickname for `160.36.28.57`?

The symbolic name of a site is determined by a naming standard known as the *Domain Name System* or DNS. Each field in a symbolic address corresponds to a single domain. The first field is a host name, which identifies a single computer. The last field is a top level domain. In between are things like department names, organization names, and so on in order of increasing generality.

For example, `www.techcorps.org` is the address for the main server of *Tech Corps*, which explains the ".techcorps" in the name, and this is an organization, which explains the ".org" part of the name. When we want to be precise we will term a sequence like `160.36.28.37` an *IP address*, while we will call the corresponding name `csep10.phys.utk.edu` the *DNS name* for the computer with IP address `160.36.28.37`. However, in many instances we will use the term "IP address" to refer loosely to either the number or the name.

DOMAIN	MEMBERSHIP	EXAMPLE
edu	→ Educational institutions	→ mit.edu
gov	→ Government organizations	→ nsf.gov
com	→ Commercial organizations, corporations	→ hp.com
mil	→ Military organizations	→ navy.mil
net	→ Network administration	→ nices.net
org	→ Other types of organizations	→ npr.org

COMMON COUNTRY CODES	
ca → CANADA	jp → JAPAN
au → AUSTRALIA	uk → UNITED KINGDOM
se → SWEDEN	de → GERMANY

Some top level domains in the US and some sample addresses are listed in the preceding figure. Addresses of sites outside the US end with a two-letter country code; some common country codes are also listed in this figure.

NEXT ▶

## Finding Addresses

If you wish, you may use this [exercise](#) to add links to your homepage that will aid in using [electronic mail](#) by providing ways to find addresses, and by providing on-line assistance for topics related to email. In the process, the exercise introduces the important technique of creating "lists" in HTML. In particular, you will see how to create *numbered lists*, where items in the list are numbered sequentially, and *un-numbered lists*, where items are not numbered, but each is preceded by a "bullet" symbol. You will also see how to "nest" one list as a sublist of another.

## Going Further

Here are some links that give further information about networks, the domain naming system, and email.

- [Zen and the Art of the Internet](#)
- [Email Guide](#)

For more information about using these things, see the later section on [finding things on the internet](#).

# Navigation with Browsers

Let us emphasize at the start that Web Browsers are for exploring. If you are not sure what a button does, try it and see what happens! As we will discuss in more detail while we go along, you are unlikely to damage anything by pushing buttons on the browser to see what happens, and there are simple ways to recover if you don't like what happens when you push a button.

The key to navigation of the World Wide Web with browsers is the Uniform Resource Locator (URL), which gives the address of a document, not only with respect to a local computer, but with respect to the entire Internet.

# URL Addresses

A Uniform Resource Locator or URL address is a wedding of the information in the IP address for a machine and the information in its local file structure. Thus a URL address gives the location of a file, *not with respect to a single computer, but with respect to the entire Internet!*

## What Are URLs: An Analogy

Imagine that you live in a large building and that your address in the building corresponds to a room number. We might call that a local address: anyone in the local building can locate you by using your room number. That is analogous to the name of a file on a single computer: anyone logged into that computer can locate a file on the computer if they know its name, and what folders or directories it resides in (and have permission to look in those directories).

Now imagine that someone from another country wants to locate you. The local address within the building is no longer sufficient because it doesn't specify how to find your building. At the very least, it is necessary to specify additional information giving the country, city, street, and so on of the building in which you reside.

This is now analogous to the information that a URL address provides: a URL address gives a unique address for a file with respect to anywhere on the Internet, just as your complete residential address gives a unique way to locate you from anywhere in the world. Thus, URL addresses allow the computers of the Internet to behave at a certain level as if they were a single computer.

## What do URL Addresses Look Like?

Here is an example of a URL address:

<http://csep10.phys.utk.edu/webcourse/browser/textfile.html>

This is a functioning URL address, and it is also a hypertext link (notice the color and the underline, and that if you hold the mouse over the link the pointer turns into a pointing hand, all of which indicate that this is a link). Therefore, you can go to it by clicking on it. Try it (but then come back here, by using the Back button on the browser).

## URL's Can Address More Interesting Things

The preceding example shows the use of a URL to specify a file containing text. However, URL's can be used to address much more general things. For example, try the following links corresponding to

An image at

<http://www.techcorps.org.org/webcourse/browser/usa2.gif>

A sound file at

[http://www.techcorps.org/webcourse/browser/hasta\\_la\\_vista.au](http://www.techcorps.org/webcourse/browser/hasta_la_vista.au)

A movie file at

<http://www.techcorps.org/webcourse/browser/goldgate.mpg>

Therefore, we see that a URL address is a very powerful thing, allowing us to address many different kinds of files.

## Comments: Case Sensitivity

This is as good a place as any to issue a warning about a common pitfall in accessing directories and files on the Internet. Some computer systems (for example, Unix) use case-sensitive

names for files and directories; others, (for example, Windows and Macintosh) ignore the case in such names. Thus, on a Windows computer *file1* and *File1* refer to the same files, but on a Unix system these would generally be two distinct files.

You can come to grief over this in the following way: Suppose you have a GIF file named *myfile.GIF* on your Windows computer and you access it locally from your browser using a Web link of the form

```
<a href="myfile.gif">
```

This will generally work on your Windows computer because it views *myfile.gif* and *myfile.GIF* as the same files.

Confident that everything is working as it should, you now transfer the file containing this link and the GIF file to a Unix Web server and try to access this link over the Web. To your dismay, you (and anyone else on the Web trying to access your file) will now get an error message that the file *myfile.gif* is not found on this Web server. Why? Unix is case sensitive, therefore (unlike the Windows computer that you used to develop the files), the server views *myfile.GIF* as being a different file from *myfile.gif* and croaks. The only cure in this case is to change either the filename in the link or the name of the GIF file so that the names are case compatible.

## Comments: Blank Spaces

A second pitfall in Web addressing is associated with the fact that different systems deal with blank spaces in file or directory names in different ways. For example, Windows file names can have blank spaces, but Unix systems generally use blank spaces as separators between names. Thus, a filename *July Budget*

*Reports* is a perfectly acceptable filename for Windows, but is a bad choice for a Unix system because it will interpret this filename as *July* because of the trailing blank space unless special actions are taken. (For example, always enclosing the entire filename in double quotes would cause a Unix system to construe the blank spaces as part of the file name.)

The common ways that Unix systems name such files without employing blank spaces is to use upper case letters to start words (with no blanks between words), or to use underlines or dashes to indicate where blank spaces would be. For example,

- JulyBudgetReports
- July\_Budget\_Reports
- July-Budget-Reports

would all be acceptable Unix filenames.

## Words to the Wise

Now if you are a Windows user developing your own material for the Web, you probably would like to ignore these "peculiarities" with Unix systems concerning case sensitivity and blank spaces. Unfortunately, you cannot because most of the servers on the Internet use Unix operating systems, and indeed the Internet itself largely developed in a Unix environment. This has two consequences:

1. Because of the Web's Unix heritage, URL addresses generally cannot contain blank spaces (if blank spaces are required they must be inserted with special character sequences).
2. If your Web material is served from a Unix server, the case of filenames and directories will matter.

Therefore, if you are going to be developing material for the Web,

we suggest strongly that you immediately get into the habit of (1) assuming case sensitivity in all your filenames, and (2) not using blank spaces in any filenames.



## URL Details

Since URL addresses are so important, let's look at their structure in a little more detail. We shall use as an example the address

<http://www.whitehouse.gov/kids/index2.html>

URL addresses have 2 basic parts: (1) a portion specifying the *method* of file access, and (2) a portion specifying the Internet location of the file to be accessed.

### The Method of File Access

The first part of a URL, ending with a `://`, tells the type of file access that is permitted from the Internet. In the preceding example *http* says that the "server" that is going to give the file to you is of the type "http", which stands for "HyperText Transfer Protocol". This part of a URL address has a function similar to that of specifying the method by which a normal mail package will be transported (the U. S. Postal Service, Federal Express, UPS, ...).

The http transfer protocol is the most common one on the Web, but it is not the only one. For example, you may also see URLs addressing servers using "File Transfer Protocol", or ftp:

<ftp://ftp.funet.fi/pub/>

Now, if a browser can be used to get a file (almost) anywhere on the Internet, you might think that surely it can be used to access a file on the same computer as the browser? Indeed it can, and if you do that (by choosing "Open" from under the "File" menu) the URL will have the typical form:

`file:///usr/people/guidry/webcourse/frame1.html`

The URL address beginning with "file:" is the signal that the address is not off somewhere else on the Internet, but is in fact on the same computer as the browser. Actually, the above would be the form of the URL for a file on a Macintosh or a Unix machine. For a file on a PC, you would see a URL of the form

`file://c:/usr/people/guidry/webcourse/frame1.html`

since a PC designates a disk drive, or other file storage device, with a letter and colon, rather than with a name. As a final example, you may often encounter the "mailto:" type URL address; for example:

`mailto:user@domain.org`

In this case, "mailto:" indicates that this URL can be used to send an email message to the address specified after the colon. There are other transport protocols, but these examples illustrate some of the most important.

## The Internet Location of the File

The second part of the URL address specifies the Internet address of the machine housing the file and the location of the file at that internet address. In the first example cited above, *www.whitehouse.gov* specifies the IP address of the http server that houses the file. (See the earlier section on the [Domain Naming System](#) for the Internet.) The rest of the URL address then gives the location of the file in question on the machine specified by the preceding IP address. For example, in the URL address

<http://www.whitehouse.gov/kids/index2.html>

the file being accessed is called *index2.html*, and it resides in a directory (folder) called *kids* on a machine with IP address *www.whitehouse.gov* that is supplying the file with a server

running the http protocol.



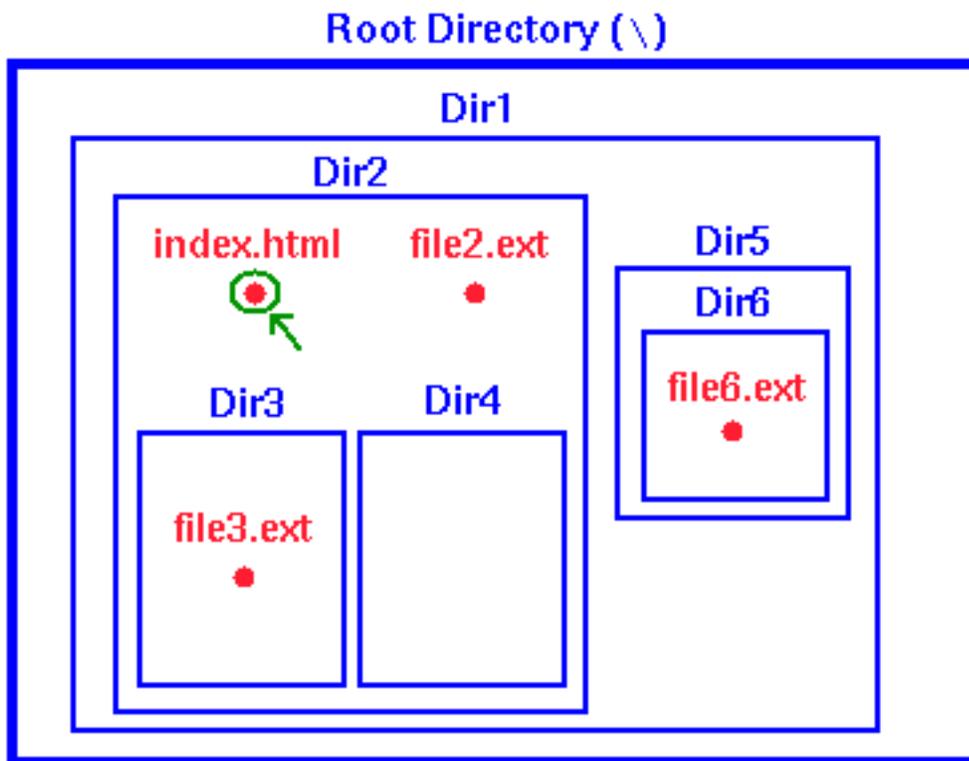
# Relative Addresses

The term "path" is sometimes confusing for new users of the Web. However, it is really a simple concept: A *path* is the sequence of directories that must be traversed to get from one directory to another. (In Windows directories are often called "folders"; we will use the two terms as synonymns.) In order to construct all but the simplest of Web documents, it is useful to know the basics of how to specify paths between directories and files.

The subsequent material in this section is somewhat more technical than the preceding discussion. It is not essential to casual Web use, but is important if you plan to construct your own Web site.

## Directory Hierarchies

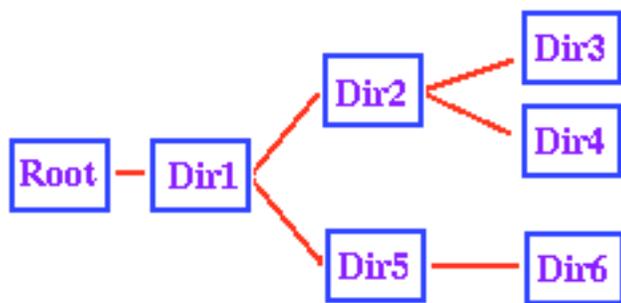
Consider the directory structure shown in the figure below.



This typical directory structure has a master directory (usually called the *root directory* and designated on a PC by the symbol "\") that contains within it all other directories as subdirectories, and each directory so contained may itself contain subdirectories. In this example the root directory contains the directory *Dir1*, and this directory in turn contains the directories *Dir2* and *Dir5*. Finally, *Dir2* contains subdirectories *Dir3* and *Dir4*, while *Dir5* contains a subdirectory *Dir6*.

The files on the system are contained in these directories. In the figure some representative files are illustrated in red. For example, the file *index.html* (see the green arrow) is contained in the directory *Dir2*, which is a subdirectory of *Dir1*, which is finally a subdirectory of the root directory.

## Directory Trees



Another useful way to look at this directory structure is in terms of a *Directory Tree*. The adjacent figure shows a tree diagram corresponding to the directory structure

illustrated in the first figure. If we view this as a tree (fallen on its side), the branches of the tree correspond to the subdirectory structure. If you stand the directory tree upright, this analogy also makes clear why the master directory containing all other directories is termed the "root" directory.

## Paths

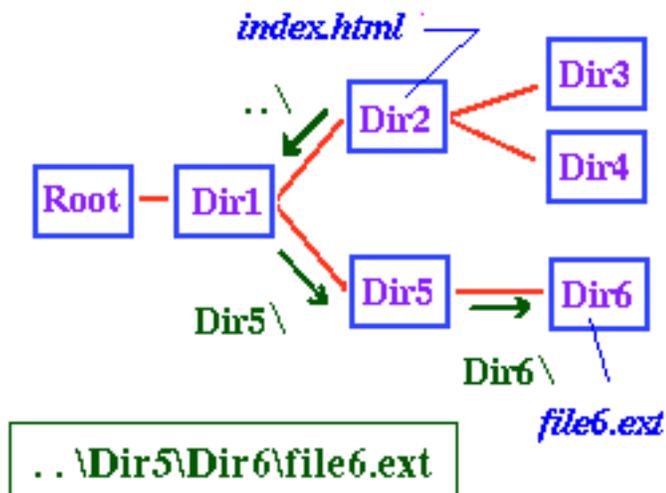
The directory tree diagram illustrates graphically the meaning of a path: it literally *is* a path: the (shortest) one that we would follow on the directory tree to go from one directory to another. For example, consider the files illustrated in the first diagram.

Suppose we ask what the "path" is from the file *index.html* to the file *file3.ext*. The file *index.html* is in the directory *Dir2* and *file3.ext* is in the directory *Dir3*. From the tree diagram, to get from *index.html* to *file3.ext* we must first go to the directory *Dir3*, which is contained in *Dir2*. We say that (relative to the file *index.html*) the file *file3.ext* is at the location *Dir3\file3.ext*. Likewise, relative to the directory *Dir1*, the file *file3.ext* is at *Dir2\Dir3\file3.ext*, and so on. Examples such as these are termed "specifying the relative path to a file" from some starting point.

## Specification of Paths

Thus, the complete path to a file that lies in a subdirectory of your present directory is specified by listing, in order, the directories (omit the one that you are in) that you must pass through to get to the file, with backslashes after each directory name, and followed

finally by the file name. Pretty simple, once you get the hang of it!



But what if the file is not in a subdirectory of your present directory? For example, where is the file *file6.ext* (which is in *Dir6*) relative to the file *index.html*? The same rule applies as before, but the adjacent figure illustrates that to get to the directory *Dir6* from the directory *Dir2*

we must first go *BACK* in the directory hierarchy until we reach a directory that contains *Dir6* (and, of course, *Dir2*) in its subdirectory structure.

Thus, to reach *file6.ext* from *index.html* we go back to *Dir1*, then to its subdirectory *Dir5*, and then to its subdirectory *Dir6*. The notation that is used to go back one level in a directory hierarchy is "..\" (two dots, followed by a backslash), and we may say that the location of *file6.ext* relative to *index.html* is *..\Dir5\Dir6\file6.html*, as illustrated in the preceding figure.

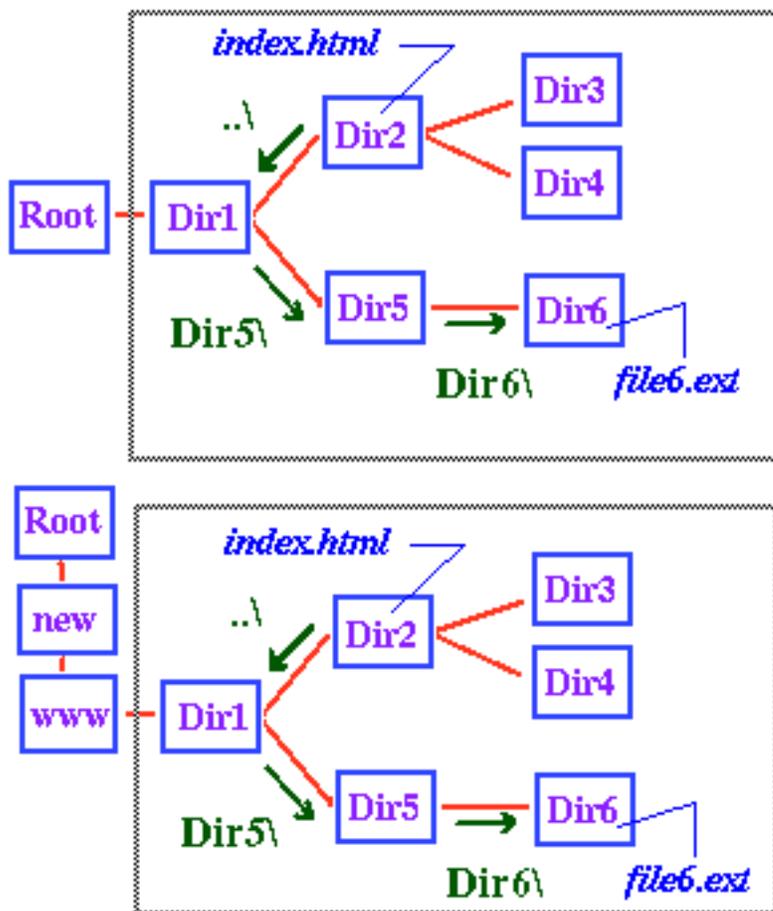
## Absolute and Relative Addressing

The examples of addresses we have just been considering are termed "Relative Addressing", because the location is specified *relative to our present location*. However, notice that we could also specify an "Absolute Address" for files by going back to the root directory and writing the complete path to the file from there. In the previous example, the absolute address of *file6.ext* is *\Dir1\Dir5\Dir6\file6.ext*, where the initial "\" stands for the root directory.

Notice the difference between absolute and relative addresses: if I move the file *index.html* to a new directory, the address of *file6.ext* relative to *index.html* changes, but its absolute address

remains the same.

On the other hand, suppose I move the entire directory structure to the right of "Root" in the directory tree intact to a new location on the same computer or a different computer, as illustrated in the adjacent diagram. Then generally the absolute address of *file6.ext* would change, but its address relative to *index.html* (indeed, relative to any file in the directories that I moved) would stay the same.



Thus, in the preceding diagram the absolute address of *file6.ext* is *Dir1\Dir5\Dir6\file6.html* in the upper example and *new\www\Dir1\Dir5\Dir6\file6.html* in the lower example, but since the part inside the gray boxes is the same in the two cases, the relative address as referenced from *index.html* is unchanged.

## The Connection Between Paths and URLs

So why should you worry about paths, whether absolute or relative, if you are creating web documents? Because the path is the part of a web document's URL that gives the document's location on the server on which it resides. Just as the location of a file can be given relative to the location of another file, or can be given as the absolute path to the file (which is really the file's location relative to the computer's root directory), so the path portion of a document's URL can also be specified relative to the location of another document or in terms of an absolute location. In the latter case, one refers to an absolute URL, and it is given, not in terms of the location relative to the computer's root, but in terms of the base directory for all the html documents. This directory is often called the *html document root*.

Thus to determine the appropriate path portion of a document's absolute URL, one needs to know its directory path on the server, relative to the html document root. If the server is a Windows machine, it has a path of the form that we discussed above. To get the path portion of the document's absolute URL, simply:

1. write the directory path of the file relative to the html document root directory
2. replace every backslash in the path with a (forward) slash

To obtain the complete absolute URL, combine this path part with the host name and the string (`http://`, `ftp://`, etc.) that gives the method of file access.

To determine the URL of document A relative to document B, just write the directory path of A relative to B and replace each backslash with a (forward) slash.

So why, you may well ask, do I have to worry about both slashes and backslashes? Why couldn't we just use one or the other for both directory paths and URLs? The answer lies in the fact that URLs and web browsers were first developed on Unix machines. There the forward slash is used in directory paths and they were naturally carried over into URLs. Unfortunately, however, the

backslash already had a long history of serving as the delimiter in directory paths in the PC world, so it was retained in this role even after Windows machines began to be used as Web servers. Consequently, in the Windows environment one has to use two different symbols, while you can get away with only one in the Unix world.

## Use Relative Addressing Whenever Possible

Absolute and relative addressing clearly each have advantages and disadvantages, but in producing Web pages it is advisable to make addresses *to files on the same machine* relative addresses unless there are circumstances dictating otherwise. Then, if the directories containing your Web pages are moved intact to another machine the links among these files should still work properly. If instead the files are specified by their absolute addresses, the links to local files in your directories will often "break" when you move the files, because the absolute addresses will generally change. Repairing these broken links can be a time-consuming task if your site involves many files.

## Homepage Exercise: Relative and Absolute Addresses

If you wish, you may now use your homepage folder and files to illustrate some of these ideas with this [exercise](#).

The elementary examples in this exercise illustrate the difference between absolute and relative addressing, and how to implement relative addressing for the two simple (but common) cases where the file being addressed is either in the same directory as the HTML file (the *next.gif* example), or in an immediate subdirectory of the HTML file (the *gifs\previous.gif* example). As we have noted above, it is generally advisable to use relative addressing in your Web pages to increase portability.

NEXT ►

## URL Summary

So do you need to know all these details about URL's to access files with your browser? Not at all! You don't usually need to know why the URL address is what it is any more than you need to know why your normal mail address is what it is in order to use it. You just need to give the browser the correct address, and let it take care of the details, just like you put the correct address on a letter and then let the Postal Service worry about the details of delivering it.

That is why Web Browsers are just about the greatest invention ever: You don't need to know very much to be able to use them quite effectively! They hide an enormous amount of technical sophistication behind a simple and intuitive graphical interface.

To review, a URL address specifies uniquely a file on the Internet and has two basic parts, the server method, and a specification telling exactly where the file is on the Internet:

<http://www.whitehouse.gov/WH/Welcome.html>

The Server Method. In this case, http means "HyperText Transport Protocol"

The location of the file. In this case, the server is at the address [www.whitehouse.gov](http://www.whitehouse.gov) and [Welcome.html](http://www.whitehouse.gov/WH/Welcome.html) is a file in directory WH.

Note: the URL in the above figure is shown for illustrative purposes only. While once valid, the URL was rendered inoperative when the White House site was reorganized after the eventual results of the 2000 presidential election.

A more complete discussion of URL addresses may be found at

<http://www.december.com/html/>

but what you have learned above should be sufficient for most uses of the Web.



## Methods of Navigation

So now we know WHAT to navigate to (URL addresses). What are the *methods* that we can use to navigate with a Web Browser? There are several ways, some of which you have already used:

(1) You can click on links and buttons, as you have already done in several examples. If you have difficulty, visit the [Web Browser Primer](#) to review how browsers work.

(2) You can enter the URL address directly into the browser. In Explorer, you can enter it into the "Address" window on the browser:



(3) You can enter the URL address into the window that appears if you select "Open" from the pulldown ["File" menu](#):



(4) You can use the "[Favorites](#)" pulldown menu to select URL addresses previously visited.

There are a few other ways to navigate too, but these are sufficient for most purposes.

NEXT ►

## Going Further

The material that we have discussed in this section should be more than sufficient for most basic Web navigation. A more extensive discussion of URLs may be found in [A Beginner's Guide to URLs](#). A broader discussion of Web navigation may be found in the topics of the [HTML Station](#) reference guide. Finally, the Web browsers have help documents that describe all aspects of using browsers.

[NEXT](#)

# Finding Things

Now you know how to find a page once you've learned its URL, but how can you find information on the Web about some person, place, or thing if you don't know the URL at which this information resides? The answer to this question is clearly crucial to the utility of the Web for educational (or other) purposes. As we shall now see, there are some very powerful and efficient ways to find things on the Web.

## [Finding Things](#) | Compilations of Resources

# Compilations of Resources

*Resource compilations* are lists of links to pages with information on particular subjects. These lists may be constructed by a program, but they are often created by people who are interested in particular topics and who have already searched the Web for sites with information on these subjects. For example, suppose you are interested in the sport of soccer. Then you might like to check out these sites:

- [Women's Soccer World WWW Page](#)
- [U. S. Soccer](#)
- [Soccer on US TV Information Pages](#)

This is an example of a very simple Web resource compilation for a particular topic. Here is a large list of [resources specifically for K-12 education](#). (Remember to return to this page with the browser's "Back" button or the "Back in Frame" command if you check these compilations.)

To make your own compilation (or to understand how they are made), start at a page that has information about the topic you are interested in. Look for other links to relevant information and visit those pages on different web sites. Keep track of the URLs by either using the Add to Favorites option of the browser's Favorites menu or by cutting and pasting the URLs to a new Notepad file that will become the source for a new web page with the listings.

If you're still looking for additional sites on your topic and you are using Internet Explorer 5.x, you can take advantage of the "Show Related Links" feature which is in the "Tools" pulldown menu.

## Caveats

When you visit a new site, notice the source of the information. Is it a reputable institution or a publisher that has expertise in this area? Or does the organization behind the source seem to have an "agenda?" Keep in mind that the information on a web page is a reflection of the organization or individual who created it. Also take the time to examine the information on the site -- to the best of your knowledge is it accurate? If you notice some inconsistencies, you may want to reconsider before adding the site to your compilation.

If you're looking at someone else's list of links, you may find dead links or links to outdated information. Developing a collection of links is one thing; taking the time to maintain the list is another.

[NEXT ►](#)

## Search Engines

A *search engine* is a program that can search the Web on a specific topic for you. By typing in a word or phrase (known as a keyword), the search engine will produce pages of links on that topic. Supposedly, the more relevant links are at the top of the list, but that is not always true. To understand more specifically how search engines operate visit [How Search Engines Work](#).

### Example: the *Google* Search Engine

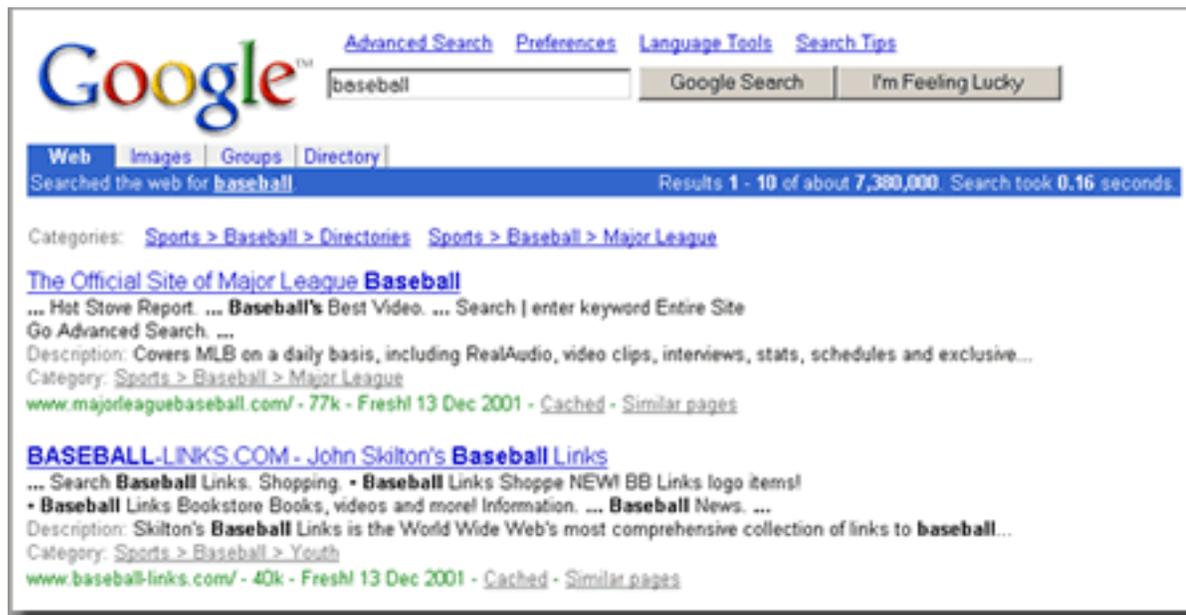
An example of a search engine is the popular *Google* Search Engine. The homepage of this web site includes a small window, which you can type a word or set of words on a topic you are interested in.

For instance, suppose you wanted to locate information about baseball on the Web. You could go to the *Google* site and enter your query in the entry field:



You then press on "Google Search" (other search engines will have slightly different forms). The search engine will respond by providing you with a list of all the documents in its database that

contain the word "baseball". Your browser will display a page with your keyword(s) and a list of documents in which the engine found the word(s). For a Google search on "baseball", the results might look like:



In this example (click on [image](#) for larger example), we displayed only the first few matches. Note that each match (or "hit") returned by Google has a link to a document that contains the keyword(s). The link appears at the beginning of the hit telling you the title. You can go directly to the document by simply clicking on the link. (But in this example don't try to go to the document by clicking the underlined words above. Remember this is just a "photo" of the page the engine returned, not the actual page itself.)

## Sharpening Your Search

If the keyword you specify for the search is a rather common term, you may be presented with an enormous number of Web documents that contain that term. (Note that Google found over 5 million documents/links to the word "baseball") In this case many of the hits may contain the term but in only one or two instances. Rather than wasting a lot of your time looking through the long list and visiting links that may turn out to have little or nothing to do with the topic you are really interested in, you can let the search engine

do the sifting by repeating your search with additional and more specific keywords.

Here are a few general tips to help your search:

- Be as specific as possible. For example, maybe you are really only interested in baseball information on the Chicago Cubs, then you should use the term "Chicago Cubs" instead of the general term of "baseball."
- If you're truly interested in the "Chicago Cubs," wrap the term in quotes. This means you want both words together. Otherwise you will pull travel web sites about the city of Chicago or unnecessary documents with information about baby bears into your result lists.
- On most search engines adding a plus sign (+) next to any term means to be sure and include the term in the results. If you want very specific information about the Chicago Cubs' 2002 season (forget their past), put a plus sign (+) next to both terms. For example +"Chicago Cubs" and +2002 means that you want only links that have both information about the Cubs and their activities during the year 2002.

These general tips work for most search engines, however each search engine may differ in how they want the user to search. Most search engines do have help files and information on advanced searching if you want to hone your searching skills further.

## Other Search Engines

There are many search engines besides the Google service available on the Web. Here are the links for some other popular ones:

[AltaVista \(www.altavista.com\)](http://www.altavista.com)

[Northern Light \(www.northernlight.com\)](http://www.northernlight.com)

[Excite \(www.excite.com\)](http://www.excite.com)

Although there may be minor differences in the details of their use,

most of these engines behave in a similar way, accepting a list of keywords from you and returning a list of documents that contain those keywords. Keep in mind that if you don't find the results you're looking for in one search engine, try another because each search engine analyzes the Web a different way.

## Internet Safety

Search engines can access a broad range of material on the Web, depending on search strings used. Thus, unsupervised use of search engines by students can lead to access of inappropriate material.

It is advisable to consider this if you intend to use search engines directly in your classes (as opposed to using them to find material that you intend to use later in class). Fortunately, some search engines try to prevent this. [Searchenginewatch.com](http://Searchenginewatch.com) has a list of the search engines on their web site that attempt to screen inappropriate material.

Also, you may find it useful to warn your students that it is possible to go back and find out from the computer which files and addresses have been accessed by the Web browser on that computer (by looking at what are called the cache files). You should also consult the discussion of [netiquette](#) and the section on [Internet Safety](#).

NEXT ►

## Directories

Web directories, such as [Looksmart](#) and [Yahoo](#), are similar to search engines. At first glance they appear to be the same, but people compile directories while web crawlers and similar computer programs compile the databases used by search engines. In other words, a directory is built by someone looking at a web site to determine if it's worthy of being added to their database and if so, under what category it should fall. Regular search engines are not so discerning. As you might suspect, directories don't cover as much of the Web as a search engine, and there is the potential element of human bias in categorization.

Many directories have search tools to help you find specific items within each category.

### **Caveat about Search Engines-Directories**

Sometimes, a search engine company or directory decides to accept what are termed "paid listings". This means that companies can pay to be displayed prominently in the search engine results. For example, if someone types in a search for "antique car," someone could pay to have "sellmywreck.com" appear at the top of the results listings.

Because paid listings tamper with the credibility of search engines, most engines avoid the practice altogether or at least mark the results list in a way to let the user know that someone paid to be listed first---that the first link is not necessarily the best place to look for the requested information.

## Other Tools

We have just seen that search engines provide powerful tools for locating specific Web documents relating to almost any given subject. Can you use search engines to obtain information about people and organizations on the Web? Sometimes you can. For example, try the [Google](#) search engine on your own name and see what it turns up; you may be surprised.

There are also many specific compilations designed to help you locate people and places. We'll look at two examples.

### PH Directories & Phonebooks

Many organizations maintain online PH directories and "phonebooks" that you can use to obtain information about people in those organizations. Here is a site with a good [compilation of links](#) to PH directories and phonebooks for universities and other organizations all over the world. Use these links to see if you can find some interesting or useful information. Try this exercise to get started: is there anyone with the last name of "Mann" at the United States Department of Education? Do you find someone named C. Dundee at the Queensland Institute of Medical Research in Australia?

### Domain Names -Whois

There are many Internet companies that register domain names (just type "domain registration" into your favorite Search Engine and you'll find plenty). Some of these companies, such as [Networks Solutions](#), have a "Whois" search tool, which allows you to find who owns a particular Internet domain name and other

relevant information about a specific IP address.



## Who Owns It?

Now that you have become adept at finding things on the Web, you should be aware of the important question of ownership and copyright: what things on the Web may be freely downloaded and what things may not be, at least not without restrictions? For Web users in the K-12 environment this is of double importance: not only should you be aware of personal legal liabilities, you also are generally influencing the attitudes of students toward these issues, which may have much larger implications than your personal legal exposure.

## Copyright Law

The question of what you may legitimately download can be a very sticky one. In some cases it is simple: clearly trademarked commercial logos (e.g., the Golden Arches) are likely to be protected by trademark law and an army of corporate lawyers. At the other extreme are tutorials like the one you are using now, for which the authors have given explicit permission for you to use it as you would like (subject to some broad restrictions).

Of course, even in the above "clear-cut" situations things may still be murky. For example, someone may give you permission to copy something from their page when they do not have the authority to do so. If they insert someone's corporate logo on their page and then give you permission to copy it, they may have violated copyright or trademark law in placing the image there to begin with. This example illustrates two important principles:

1. Just because something appears on the Internet does not mean that it is legal for you to copy it.
2. The law generally holds that it is YOUR responsibility to determine the legality of copying something, irrespective of whether the source from which you copy it has broken the

law or not.

A related point to bear in mind is that when, as in the case of the Tutorial, someone gives you permission to use their material, this permission does not extend to material that may be accessed by a link from the original material. The question of how material accessed in an external link may be used clearly must be put to ITS owner.

Notice carefully the distinction: you may usually insert a *link* to someone else's Web material in your pages without restriction; the issue of ownership becomes important only when you attempt to download material (text, images, movies, . . .) from other Web sites and incorporate it directly into your Web material.

## Fair Use

Although copyright law protects those who create the work, even a protected work can be "used" under certain circumstances. This circumstance is known as "fair use" and according to the Copyright Law, copying work "for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research, is not an infringement of copyright."

There are restrictions for fair use, but the purpose of this section isn't to dispense legal advice. We merely want to make you aware that there are some important legal considerations when using material from the Web.

## More Information

Here are some links to more extensive discussions of copyright, fair use and trademark law:

- [A Brief Intro to Copyright](#)

- [U.S. Copyright Office](#)
- [10 Big Myths about Copyright Explained](#)
- [United States Patent and Trademark Office](#)
- [University of Tennessee Library pages on Fair Use](#)
- [A Short Course on Copyright for Educators](#)

You are well advised to look at this material if you intend to download and use significant amounts of material from the Web.

NEXT ►

## Shareware & Freeware

There is software available on the Web that can either be downloaded for free, or downloaded for a small fee that you are expected to pay on an honors system. The first of these categories is commonly called "Freeware"; the second is called "Shareware". Shareware is often offered with an explicit grace period in which you can try the software for free; if you continue to use the software after that, you are expected to pay the shareware fee. Typical shareware fees are in the \$10-\$50 range.

### Licensing Agreements

Shareware and freeware both often come with licensing agreements: the software is only free or available at the quoted shareware price if particular restrictions are observed. For example, a typical restriction is that the software cannot be incorporated into a commercial product. Another common licensing agreement may restrict the software to individual or educational users. Notice that software sometimes comes in several versions. In the most extreme examples there may be freeware versions, shareware versions with extended capabilities, and a fully commercial version with additional features (or, perhaps, pseudofeatures---caveat emptor).

### Pay the Fees!

Paying for shareware is an honors system, but the writers of good shareware are more than deserving of the small fees that they request. Try the shareware out for free, but if you continue to use it register and pay the fees (downloaded shareware typically comes with README files that explain what to do; often the software will print a registration form for you). As an added bonus,

when you pay the shareware fee you often receive further useful material such as a user's manual or upgrade information.

Note that in addition to prices for individual copies, there may be a flat price for a "site license" that allows any number of copies of the software to be used at a single site such as a school. Also, since shareware writers as a group are clearly generous souls, you may be able to negotiate a reduced price for an educational application if you ask.

## Sources of Freeware and Shareware

There are many archives that can be accessed by Web browser or by anonymous `FTP` that specialize in freeware and shareware for microcomputers. A particularly useful link for Mac users is the [Software Hyperarchive](#), which searches these archives based on keywords that you supply and brings up links to allow automatic downloading of software to your computer.

For Macintosh software, the files will often be archived in `Stuffit` format and encoded in `BinHex`, so you will need the freeware or shareware version of `Stuffit Expander` installed on your computer to decode and expand the files into usable form. Properly configured Browsers will normally launch such helper applications automatically when encoded and compressed files are downloaded.

NEXT ►

## Homepage Exercise

We've now learned a variety of ways to find resources on the Web. If you wish, you may now use some of these Web resources and this [exercise](#) to modify your homepage into something that looks a bit snazzier.

## Going Further

You have some extremely powerful resources at your disposal in finding things on the Web. A representative list includes

### ● Search Engines

- [Google](#)
- [Altavista](#)
- [Northern Light](#)
- [Excite](#)

### ● Directories (which are more or less compilations)

- [Yahoo](#)
- [LookSmart](#)
- [Open Directory](#)

### ● Other More Specialized Tools

- [Network Solutions \(Whois Directory\)](#)
- [PH and Phonebook Directories](#)

One important thing that you can find on the Web is inexpensive software available for free, or available for a trial before you purchase it. Here are three places to start looking for freeware and shareware on the Web.

- [Association of Shareware Professionals](#)
- [SharewareStockpile.com](#)

- [FreewareWeb.com](#)

An introduction to copyright issues may be found at

- [A Brief Intro to Copyright](#)
- [A short Course on Copyright for Educators](#)
- [U.S. Copyright Basics](#)

Making *links* to things that you find on the Web usually is permitted without concern about copyright issues. However, if you *download and save* things that you find on the Web - particularly if you incorporate those things into your own material - you need to be concerned about issues of ownership and copyright.

NEXT ►

# Glossary of Terms

For specific definitions on Web terms check [A Beginner's Web Glossary](#) and [Webopedia](#), an online dictionary for computers and the Internet.