(World Wide) Web

- **a way to connect computers that provide information (servers) with computers that ask for it (clients like you and me)**
  - uses the Internet, but it's not the same as the Internet
- **URL (uniform resource locator, e.g., http://www.amazon.com)**
  - a way to specify what information to find, and where
- **HTTP (hypertext transfer protocol)**
  - a way to request specific information from a server and get it back
- **HTML (hyptertext markup language)**
  - a language for describing information for display
- **browser (Firefox, Safari, Internet Explorer, Opera, Chrome, …)**
  - a program for making requests, and displaying results
- **embellishments**
  - pictures, sounds, movies, ...
  - loadable software
- **the set of everything this provides**

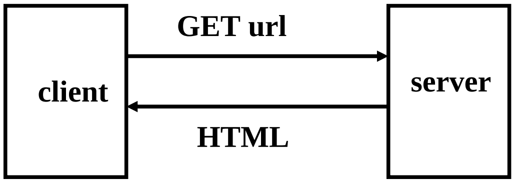
Web history

- **1989: Tim Berners-Lee at CERN**
  - a way to make physics literature and research results accessible on the Internet
- **1991: first software distributions**
- **Feb 1993: Mosaic browser**
  - Marc Andreessen at NCSA (Univ of Illinois)
- **Mar 1994: Netscape**
  - first commercial browser
- **technical evolution managed by World Wide Web Consortium**
  - non-profit organization at MIT, Berners-Lee is director
  - official definition of HTML and other web specifications
  - see `www.w3.org`

HTTP: Hypertext transfer protocol

- **What happens when you click on a URL?**
- **client opens TCP/IP connection to host, sends request**

  `GET  `*`/filename`*`  HTTP/1.0`



- **server returns**
  - **header info**
  - **HTML**
- **since server returns the text, it can be created as needed**
  - can contain encoded material of many different types (MIME)
- **URL format**

  service://hostname/filename?other_stuff
- ***filename?other_stuff* part can encode**
  - data values from client (forms)
  - request to run a program on server (cgi-bin)
  - anything else

Embellishments

- **original design of HTTP just returns text to be displayed**
- **now includes pictures, sound, video, ...**
  - need helpers or plug-ins to display non-text content

    e.g., GIF, JPEG graphics; sound; movies
- **forms filled in by user**
  - need a program on the server to interpret the information (cgi-bin)
- **HTTP is stateless**
  - server doesn't remember anything from one request to next
  - need a way to remember information on the client: cookies
- **active content: download code to run on the client**
  - Javascript and other interpreters
  - Java applets
  - plug-ins
  - ActiveX

Forms and CGI programs

- **"common gateway interface"**
  - standard way to request the server to run a program
  - using information provided by the client via a form

- **if the target file on server is an executable program**
  - e.g., in /cgi-bin directory
- **or if it has the right kind of name**
  - e.g., something.cgi
- **run it on the server to produce HTML to send back to client**
  - using the contents of the form as input
  - output depends on client request: created on the fly, not just a file

- **CGI programs can be written in any programming language**
  - often Perl, PHP, Java

Example form in HTML

```
<html>
<body>
<form METHOD=POST enctype="multipart/form-data"
  ACTION="echo.cgi">

Background color:
<input type="text" name="Background" size="40">
<p>
<input type="radio" name=Color value="Red" checked> Red <br>
<input type="radio" name=Color value="Blue"> Blue <br>
<input type="radio" name=Color value="Green"> Green <br>
<input type="radio" name=Color value="Yellow"> Yellow <br>
<p>
<input type="submit" value="Send">

</form>
</body>
</html>
```

Example CGI program in Perl (echo.cgi)

```
#!/usr/princeton/bin/perl -Tw
use CGI;
$query = new CGI;

$c = $query->param('Color');
$bg = $query->param('Background');
if ($bg eq '') { $bg = 'ffffff'; }

print $query->header;
print $query->start_html(-title=>'test', -bgcolor=>$bg);
print "<h1><font color = $c>  this is a test...</font>\n";

print "<P> bg = $bg\n";
foreach $name ($query->param) {
        $value = $query->param($name);
        print "<P> $name is $value\n";
}
print $query->end_html();
```

Cookies

- **HTTP is <u>stateless</u>: doesn't remember from one request to next**
- **cookies intended to deal with stateless nature of HTTP**
  - remember preferences, manage "shopping cart", etc.
- **cookie: one line of text sent by server to be stored on client**
  - stored in browser while it is running (transient)
  - stored in client file system when browser terminates (persistent)
- **when client reconnects to same domain,**
  browser sends the cookie back to the server
  - sent back verbatim; nothing added
  - sent back only to the same domain that sent it originally
  - contains no information that didn't originate with the server

- **in principle, pretty benign**
- **but heavily used to monitor browsing habits, for commercial purposes**

Cookie crumbs

- **get a page from xyz.com**
  - it contains <img src=http://doubleclick.com/advt.gif>
  - this causes a page to be fetched from DoubleClick.com
  - which now knows your IP address and what page you were looking at
- **DoubleClick sends back a suitable advertisement**
  - with a cookie that identifies "you" at DoubleClick
- **next time you get <u>any</u> page that contains a doubleclick.com image**
  - the last DoubleClick cookie is sent back to DoubleClick
  - the set of sites and images that you are viewing is used to
    - update the record of where you have been and what you have looked at
    - send back targeted advertising (and a new cookie)
- **this does not necessarily identify you personally so far**
- **but if you ever provide personal identification, it can be (and will be) attached**
- **defenses:**
  - turn off all cookies; turn off "third-party" cookies
  - don't reveal information
  - clean up cookies regularly

Plugins

- **programs that extend browser, mailer, etc.**
  - browser provides API, protocol for data exchange
  - extension focuses on specific application area
  - e.g., documents, pictures, sound, movies, scripting language, ...
  - may exist standalone as well as in plugin form
  - Acrobat, Flash, Quicktime, RealPlayer, Windows Media Player, ...
- **scripting languages interpret downloaded programs**
  - Javascript
  - Java

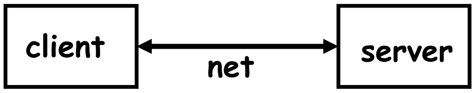
    compiled into instructions for a virtual machine
    (like toy machine on steroids)

    instructions are interpreted by virtual machine in browser

Active X (Microsoft)

- **write programs in any language (C, C++, Visual Basic, ...)**
- **compile into machine instructions for PC**
- **when a web page that uses an ActiveX object is accessed**
  - browser downloads compiled native machine instructions
  - checks that they are properly signed ("authenticated") by creator
  - runs them
- **each ActiveX object comes with digital certificate from supplier**
  - can't be forged
  - run the program if you trust the supplier
- **more efficient than an interpreter**
- **no restrictions on what an ActiveX object can do**
  - no assurance that it works properly!
- **the most risky of the active-content models**

Potential security & privacy problems



- **attacks against client**
  - release of client information

    cookies: client remembers info for subsequent visits to same server
  - adware, phishing, spyware, viruses, ...

    spyware: client sends info to server upon connection (Sony, …)

    often from unwise downloading
  - buggy/misconfigured browsers, etc., permit vandalism, theft, hijacking, ...
- **attacks against server**
  - client asks server to run a programs when using cgi-bin

    server-side programming has to be careful
  - buggy code on server permits break-in, theft, vandalism, hijacking, …
  - denial of service attacks
- **attacks against information in transit**
  - eavesdropping

    encryption helps
  - masquerading

    needs authentication in both directions

Privacy on the Web

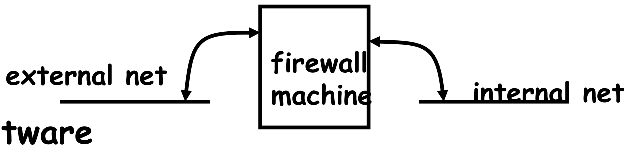
- **what does a browser send with a Web request?**
  - IP address, browser type, operating system type
  - referrer (URL of the page you were on)
  - cookies
- **what do "they" know about you?**
  - whatever you tell them, implicitly or explicitly
  - public records are really public
  - lots of big databases like phone books
  - universal numbers make it easier to track you (SSN, telephone, Ethernet)
  - log files everywhere
  - aggregators really collect a lot of information for advertising
  - spyware, key loggers and similar tools collect for nefarious purposes
- **who owns your information?**
  - in the USA, they do

Viruses

- **old threat, new technologies**
  - new connectivity makes them more dangerous
- **basic problem: running someone else's software on your machine**
  - bugs and ill-advised features make it easier
- **operates by hiding executable code inside something benign**
  - e.g., .EXE file or script in mail or document, downloaded content
- **Melissa, ILoveYou, Anna Kournikova viruses use Visual Basic**
  - applications (Word, Excel, Powerpoint, Outlook) have VB interpreter
  - a document like a .doc file or email message can contain a VB program
  - opening the document causes the VB program to be run
- **virus detectors**
  - scan for suspicious patterns, suspicious activities, changes in files

Defenses

- **use strong passwords**
- **popups off, cookies off, spam filter on**
- **turn off previewers and HTML mail readers**
- **anti-virus software on and up to date**
  - turn on macro virus protection in Word, etc.; turn off ActiveX
- **run spyware detectors**
- **use a firewall**



- **try less-often targeted software**
  - Mac OS X, Linux, Firefox, Thunderbird, ...
- **be careful and suspicious all the time**
  - don't view attachments from strangers
  - don't view unexpected attachments from friends
  - don't just read/accept/click/install when requested
  - don't install file-sharing programs
  - be wary when downloading any software