The Internet... How Does THAT work?! 

Today's Topic: Viewing A Webpage
"Glossing over things"

This thing is glossy
So that we can see a truth
Reflected in lies
Brief highlights of Web History
FTP (1972)

Archie (1991)
WorldWideWeb (1990)
Welcome to the GNU Project web server, www.gnu.org. The GNU Project was launched in 1984 to develop a complete UNIX style operating system which is free software: the GNU system. (GNU is a recursive acronym for GNU's Not UNIX - &quot;GNU&quot; pronounced &quot;go.&quot;). Variants of the GNU operating system, which use the kernel Linux, are now widely used; though these systems are often referred to as &quot;Linux&quot; or &quot;Unix&quot; they are more accurately called &quot;GNU/Linux.&quot; This is the web site of the Free Software Foundation, Inc. FSF is an international non-profit organization devoted to the preservation, protection and promotion of the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free Software Foundation.

This is a new version of the NextStep WorldWideWeb application with the lttWWW library. Bug reports to timo@info.cern.ch, quoting the version information above. Check the list of known bugs in the web too.

This was the original prototype for the WorldWideWeb. Many browsers for other platforms now exist (Read the web for details). After many years being fallow, this application has now sprouted images and nested HTML elements and things (you have an Internet connection and an Internet help server? Do you know what to do with this application? If not, don't install it, connect to Gopher too.)

If you would like to read more news, you should set the name of your local news server in the preferences.
Warning: this is a beta release of ViolaWWW. Updates of this software may be found in ftp://ftp.ora.com/pub/www/viola. Bug reports, etc, would be greatly appreciated.

ViolaWWW is a World Wide Web browser. ViolaWWW is built using the Viola hypermedia language/toolkit, and now also comes with a Motif front end.

Viola’s support of HTML 3.0 (aka HTML+) so far includes:

- Paragraph as container.
- Nesting lists.
- Input forms.
- Tables.

URL: http://berkeley.ora.com/proj/viola/vw/about_3.3.html
Welcome to MCLI

... the Maricopa Center for Learning and Instruction, located at the District office of the Maricopa Community Colleges, Arizona. MCLI is considered a national model for motivating, infusing, and promoting innovation and change in the community college environment. Listen to our audio Greeting!

On the World Wide Web with MCLI

Places to go, things to do...
- HOT Links to the World Wide Web
- Teaching and Learning on the Web
- Into the Internet with Mosaic; as printed, it makes a nice handout.
- Get a start creating Mosaic pages or jump into our tutorial on Writing HTML Documents
- If you plan to travel the Information SuperHypeWay, then pay a visit Dr. Internet. Or, fill up with a bowl of Internet Soup. At least, make sure that you have earned your drivers license
- Information about the MCLI World Wide Web server and latest statistics on its use.
- What's in our URL? What is "hakatai"?

Using Mosaic...
- MCLI's Tips n' Tricks for Mosaic
- Compare the screens for Macintosh and Windows Mosaic clients
- NCSA Mosaic for Macintosh User's Guide
- NCSA Mosaic for Microsoft Windows User's Guide
Percent Of Total Packets Transferred Per Month By Service

- FTP
- WWW
- Gopher

Statistics provided by Merit MIC 1993
Graphs by James Pitkow, GYU
(pitkow@co.gatech.edu)

1994
1995

Prodigy et al. go online

Transition to new architecture starts to take effect
"Worse is Better"  
(VHS vs Betamax)  

Surviveable Chaos → Evolution
Key elements of the Web Browser:

- Embed navigation into documents (hyperlinks)
- Allow for experimental features (ignore stuff)
- Separate content from style (css)
- Provide client storage (cookies)
- Allow for dynamic interaction (javascript)
- Allow for dynamic server interaction (AJAX)
Web Browser (aka Web Client)

- Download a document
- Display ("render") the document
- Execute the document (javascript)
(that was the most important slide)
Now the "details"
You → Your Friend

You: "Hey! You should checkout optoro.com!"

Friend: "OK."
Friend → Browser

Friend: "Uh... go to optoro.com" (in location)

Browser: "OK. Gimme a sec..."
Browser → DNS Server

Request: "What is the IP address for optoro.com?"

Response: 23.20.80.43
'host' demo
Browser → 23.20.80.43 (aka optoro.com)

Request: "Give me /

23.20.80.43: Gives file "<html><head>........"
'curl' demo
Browser starts "rendering" aka "drawing" on your screen.
<h1>blah</h1>

Becomes a big header "blah"
<p>I've got a lovely bunch of coconuts</p>

Becomes a paragraph
Sometimes stuff wraps to fit, adjust for different fonts/sizes, margins get shifted, etc.
This is very tedious work!
Browser gets to:
```
<img src='/assets/optoro-logo--grey.svg'>
```

or something like that.
Now to fetch this image file from the server!

Browser → 23.20.80.43 (aka optoro.com)

Request: "Give me /assets/optoro-logo--grey.svg"

23.20.80.43: Gives file
And so on, asking the server for any bits needed to render the site.
Network Debugger Demo
First bit of trickery:
Server-side dynamic content
Next bit of trickery:

Client-side dynamic content
In summary...
• Type in a location (aka URL, address, or website)
  -- Parts: protocol, server, path, params, "anchor"

• Translate server name → IP Address
  -- DNS

• Connect to server
  -- AKA "computer"

• Request document
  -- Just one to start with
  -- Might secretly be dynamically generated!

• Render document
  -- Document Object Model, CSS
  -- ... do more requests for pieces (images, css)

• Execute Javascript
  -- DYNAMIC!

• Background requests (AJAX/XHR)
  -- MOAR DYNAMIC!!!