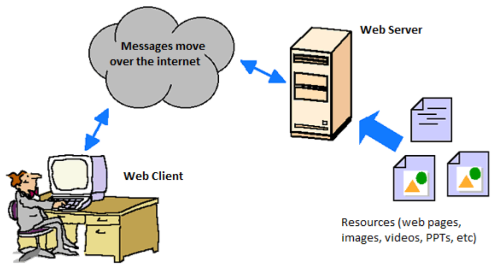
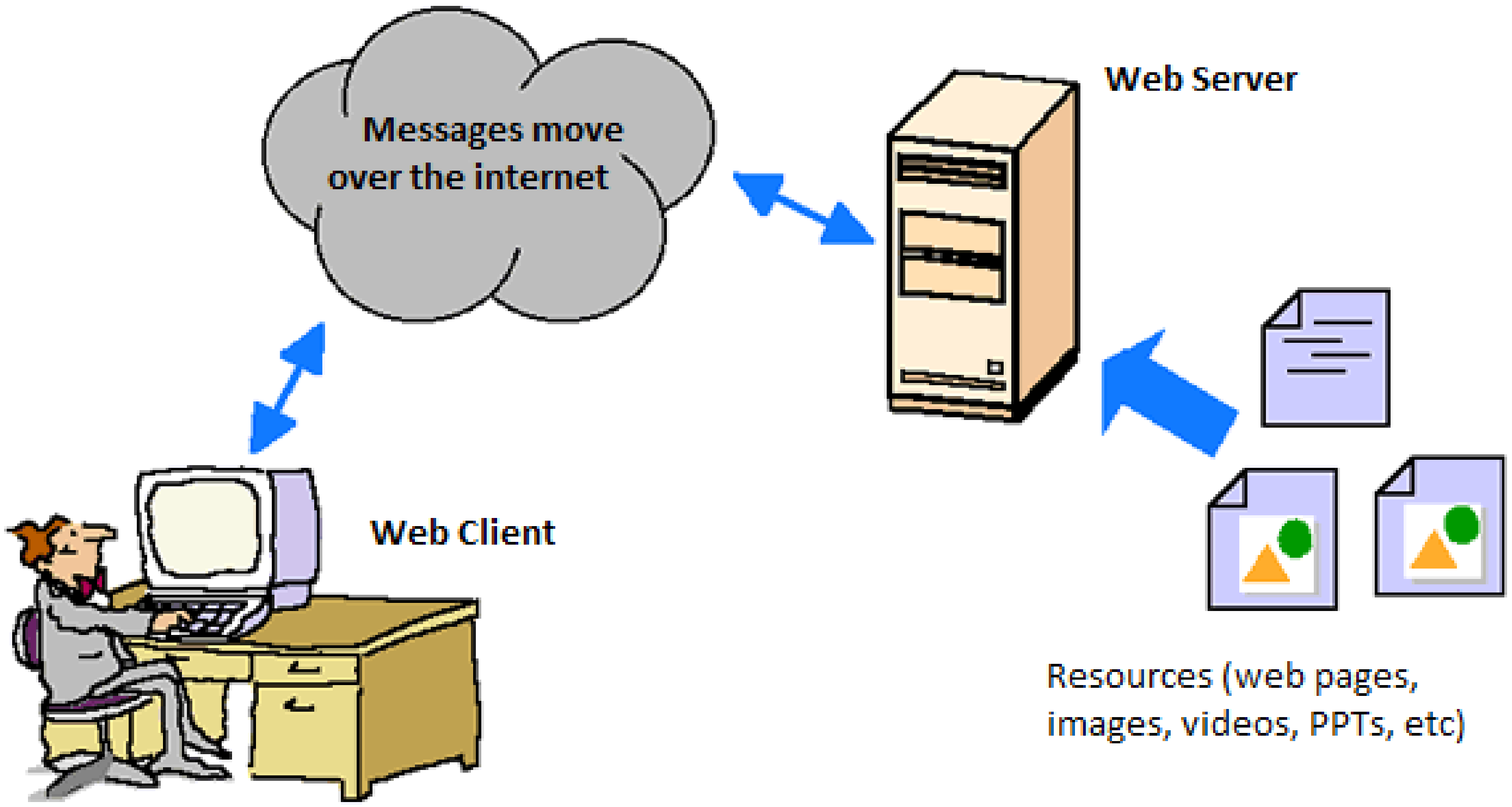
Introduction to web applications and http protocol

Requesting a page - Overview

1. A *web client* (usually in the form of a web browser) makes an *HTTP request* to a specific *web server*.
2. The *web server* receives the request and sends back the requested document (usually – though not always) in the form of an HTML page.
3. The web client examines the returned object.
    1. If the resource is an HTML file, the client parses the file and renders (displays) it on the screen.
    2. If the resource is a file that the browser “understands” (jpeg images, gif images, mp4 videos, etc), the client will play/display the resource.
    3. If the resource is a type of file that the client is unable to display (e.g. a Powerpoint presentation), the client will offer you the opportunity of saving it to your computer, or to open it in its native application (e.g. Microsoft PowerPoint).


Messages move over the internet
Web Server
Web Client
Resources (web pages, images, videos, PPTs, etc)

Requesting a page - Overview


Messages move over the internet
Web Server
Web Client
Resources (web pages, images, videos, PPTs, etc)

The Protocol: HTTP

- Stands for “hypertext transfer <u>protocol</u>”

  (You don’t need to memorize this!)
- A protocol is a series of rules and standards that are agreed to by a committee or organization.
- For the web, HTTP is the protocol used that enables effective communication between web clients and web servers.

HTTP Client

(the web browser)

- Also known as: Web Client or Web Browser
- Software that is used to:
  - generate **HTTP requests**
  - send those requests to an **HTTP server**
  - interpret the **HTTP response** that is provided by the server
- From Wikipedia:

  An HTTP client uses HTTP to connect to a web server over the Internet to transfer documents or other data. The most well known types of HTTP Clients include web browsers.

- By http client, we are almost always referring to a web browser
  - **Examples:** Internet Explorer, Firefox, Safari, Chrome, etc

HTTP Server

- This is software (a program). It is *not* a computer.
  - However, you *do* need a computer to run the http server software!
  - That being said, because the computer running the server software is typically dedictated exclusively to that task, we frequently call the *computer* the server, even though that is not technically correct.
  - HTTP server software is entirely different software than HTTP client software.
- The web server software:
  - “listens” for incoming requests from **HTTP clients**
  - “parses” (reads & interprets) that request
  - sends out an **http response**
  - Typically, this response includes sending a document or other resource back to the **web client** (aka ‘http client’)
- Examples of http server software:
  - Apache, Google Web Server, Internet Information Services

Overview: How to set up a web server

1. Have a computer to act as a ‘server’ with a big enough hard-drive to store all the pages you want to make available on the web.

2. Ensure your computer is connected to the internet

3. Install web server software (e.g. Apache web server)
   – Configure the web server (takes a little bit of know-how)

4. Copy any documents and resources you wish to make available from your local computer to the web server.
   – This requires ‘FTP’ (discussed later)

Server computer

Have a computer to act as a 'server' to send pages along when requested:

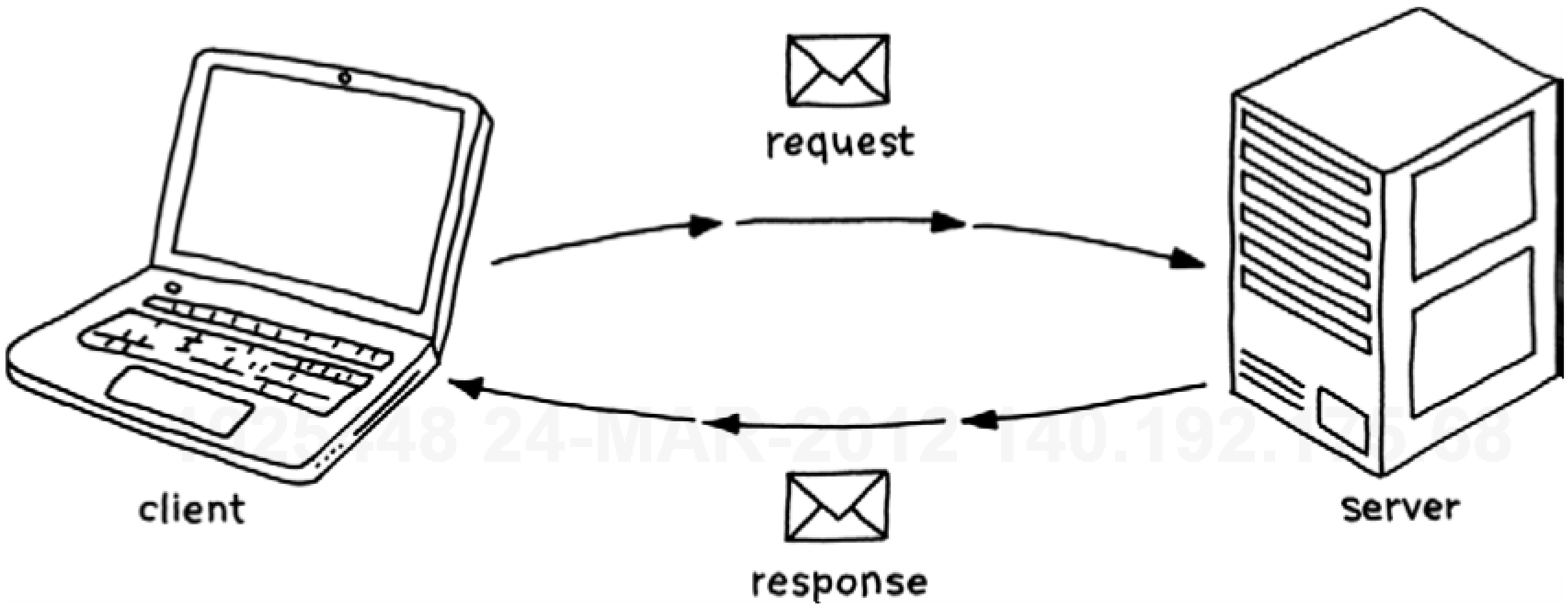
A server can be any computer capable of connecting to the internet and running server-software.

- The busier the site, the more powerful the computer.
- A server can be as simple as your desktop/laptop, or as complex as a rack of high-end dedicated computers.
- Busy sites (e.g. Google, Microsoft) will have multiple redundant computers running server software all over the world.

The **Server Software**

- Software that sits there and patiently listens for imcoming HTTP requests.
- Once a request comes in, the server responds by sending back an HTTP response along with the requested resource.
  - The resource might be a web page (html file), video, PDF, PowerPoint document, etc
- Several software companies publish web server software
  - Some of the better known server software programs include:
    - Apache HTTP Server
    - Microsoft IIS (Internet Information Services)
    - Google’s GWS (Google Web Server)

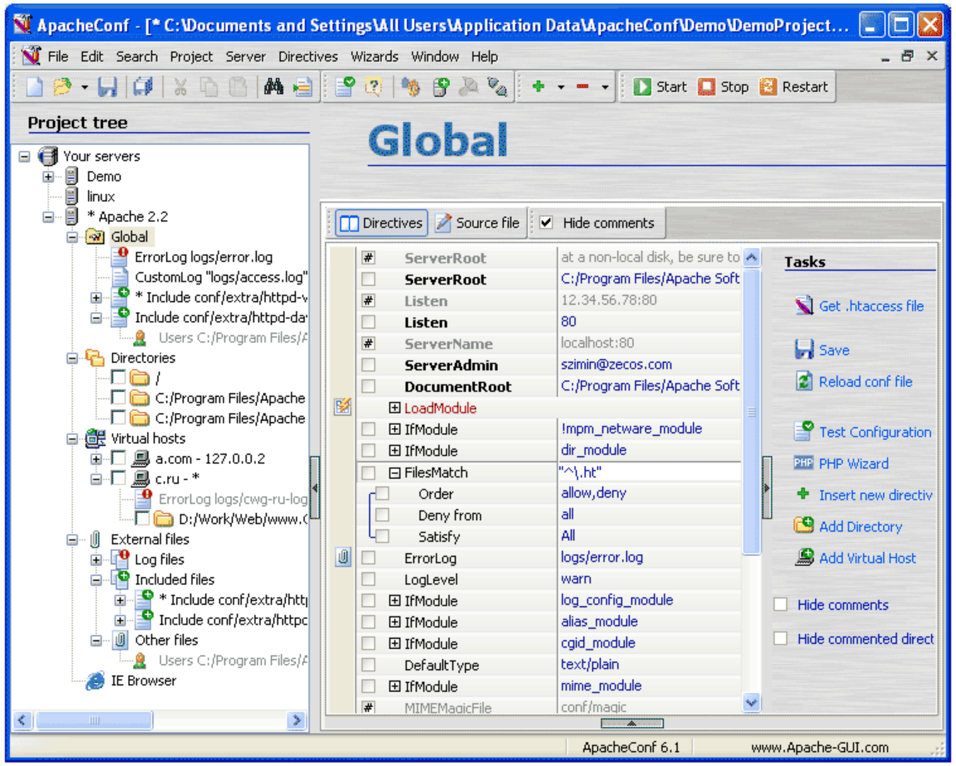
The Request/Response Process



Taken from "The Internet of Things", O'Reilly Media, page 30

Client sends a request to the server.

Server sends a response back to the client.



```
Your servers
  Demo
  linux
  * Apache 2.2
    Global
      ErrorLog logs/error.log
      CustomLog "logs/access.log"
      * Include conf/extra/httpd-v
      Include conf/extra/httpd-da
        Users C:/Program Files/A
    Directories
      /
      C:/Program Files/Apache
      C:/Program Files/Apache
    Virtual hosts
      a.com - 127.0.0.2
      c.ru - *
        ErrorLog logs/cwg-ru-log
        D:/Work/Web/www.c
    External files
      Log files
      Included files
        * Include conf/extra/http
        Include conf/extra/httpc
      Other files
        Users C:/Program Files/A
    IE Browser
```

#	ServerRoot	at a non-local disk, be sure to
	ServerRoot	C:/Program Files/Apache Soft
#	Listen	12.34.56.78:80
	Listen	80
#	ServerName	localhost:80
	ServerAdmin	szimin@zecos.com
	DocumentRoot	C:/Program Files/Apache Soft
	⊞ LoadModule	
	⊞ IfModule	!mpm_netware_module
	⊞ IfModule	dir_module
	⊟ FilesMatch	"^\.ht"
	Order	allow,deny
	Deny from	all
	Satisfy	All
	ErrorLog	logs/error.log
	LogLevel	warn
	⊞ IfModule	log_config_module
	⊞ IfModule	alias_module
	⊞ IfModule	cgid_module
	DefaultType	text/plain
	⊞ IfModule	mime_module
#	MIMEMagicFile	conf/magic

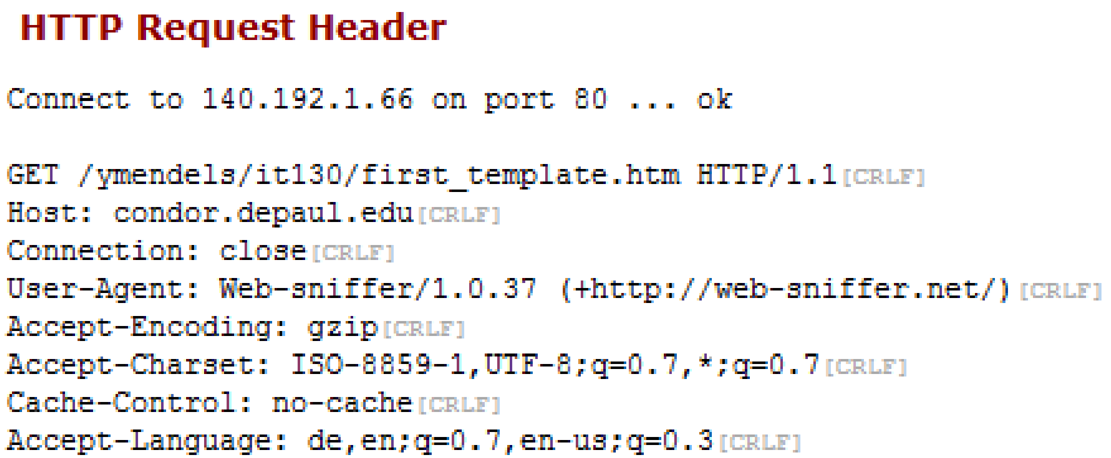
The HTTP **Request** object

- When you type a URL, say, for the syllabus of this course, your HTTP client (browser) creates something called an ‘HTTP request’
- This request is routed (via a mechanism we will ignore for now) over the internet to an HTTP server that is connected to the internet
  - For the class web page, this server lives somewhere in downtown Chicago
- The HTTP request contains all kinds of information that the server may need to know:
  - The name of the document being requested
  - The time the request was made
  - The “return address” (the internet address of the http client)
  - The type/name of the http client (e.g. Pocket Safari for iPhone)
  - etc

Example of an http request header:

- I would like to retrieve the file first_template.htm
- The URL to this file is:

  http://condor.depaul.edu/ymendels/130/first_template.htm
  - The name of this web server is ‘condor.depaul.edu’
  - The request file is ‘first_template.htm’
  - The file is stored in a folder called ‘ymendels’ and a subfolder called ‘130’.
- When I type the URL into my client, you can see part of the http request that is generated here:



HTTP Request Header

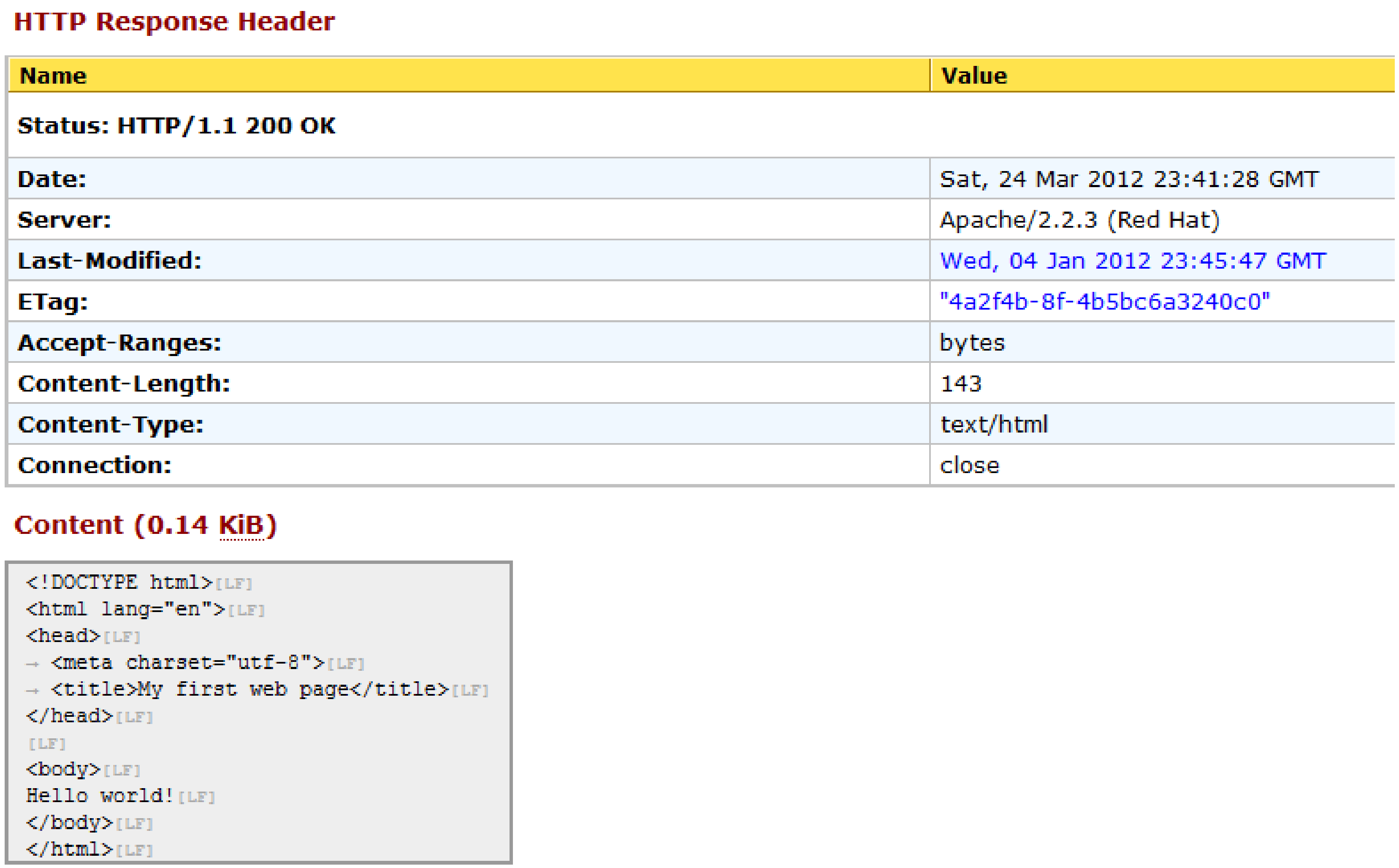
```
Connect to 140.192.1.66 on port 80 ... ok

GET /ymendels/it130/first_template.htm HTTP/1.1[CRLF]
Host: condor.depaul.edu[CRLF]
Connection: close[CRLF]
User-Agent: Web-sniffer/1.0.37 (+http://web-sniffer.net/)[CRLF]
Accept-Encoding: gzip[CRLF]
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7[CRLF]
Cache-Control: no-cache[CRLF]
Accept-Language: de,en;q=0.7,en-us;q=0.3[CRLF]
```

The HTTP **Response** object

- Once the server has received and parsed the http request, it answers by generating an http response. This response includes:
  - a status line: e.g. ‘OK’ or ‘ERROR-404’
  - a header with information such as the type of document being returned (JPG, MPG, HTML, etc)
  - This is also packaged with the actual document that was requested. This document is commonly, but by no means always an ‘html’ document.
    - It can be other things such as an image, a PPT file, etc

Example of an http request and response:
2. **The HTTP Response**



HTTP Response Header

Name	**Value**
Status: HTTP/1.1 200 OK	
Date:	Sat, 24 Mar 2012 23:41:28 GMT
Server:	Apache/2.2.3 (Red Hat)
Last-Modified:	Wed, 04 Jan 2012 23:45:47 GMT
ETag:	"4a2f4b-8f-4b5bc6a3240c0"
Accept-Ranges:	bytes
Content-Length:	143
Content-Type:	text/html
Connection:	close

Content (0.14 KiB)

```
<!DOCTYPE html>[LF]
<html lang="en">[LF]
<head>[LF]
→ <meta charset="utf-8">[LF]
→ <title>My first web page</title>[LF]
</head>[LF]
[LF]
<body>[LF]
Hello world![LF]
</body>[LF]
</html>[LF]
```

404-Error

If the web server is unable to find the requested resource, it will generate a message with the error code 404 indicating that there was an error and that the request could not be fulfilled.

The error may say something like “HTTP 404” or “Error 404” or something similar.

A common way to encounter a 404 error is when there is a typo in the URL:

http://condor.depaul.edu/ymendels/130/intro_to_htttp.htm

(note the extra ‘t’)

Anatomy of a URL

http://www.math.rs/files/130/intro_to_html.ppt

1. The request finds its way to the math.rs domain
2. Within the domain, the request is navigated to a specific server named ‘www’
3. On the condor server, look for a folder called ‘files’
4. Within the folder files, look for a subfolder called ‘130’
5. Within that folder, look for a file called ‘intro_to_html.ppt’

Thanks for your attention!