



CS 146: Intro to Web Programming and Project Development

Instructor: Patrick Hill

Email: phill@stevens.edu





Intro To the Web III

HTTP & Browsers



What is HTTP?

- *HyperText Transfer Protocol*
- An application protocol for distributed, collaborative, hypermedia information systems
- Hypermedia is an extension of hypertext
 - non linear medium of information including graphics, video, audio, plain text and hyperlinks
- Multimedia is a broader term that can also describe non-interactive linear presentation of graphics, but hypermedia as well



How it all started?

- First version was HTTP/0.9, created by early developers of the Web, which had its beginnings at CERN (1989)
- HTTP/1.1 was released in June 1997
- Protocol for client-server communication
- HTTP/2 in 2015
 - According to W3Techs, as of May 2017, 13.7% of all websites supported HTTP/2
 - As of September 2018, it's up to 29.7%



HTTP vs HTML

- HTML: hypertext **markup language**
 - Definitions of tags that are added to Web documents to control their content/appearance
- HTTP: hypertext transfer **protocol**
 - The rules governing the conversation between a Web client and a Web server

Both were invented at the same time by the same person



Application Layer Protocols

Protocol	Application
HTTP: Hypertext Transfer	Retrieve and view Web pages
FTP: File Transfer	Copy files from client to server or from server to client
SMTP: Simple Mail Transport	Send email
POP: Post Office	Read email



Three Important Facts about HTTP

- **Connectionless protocol:**
 - After making the request, the client disconnects from the server, then when the response is ready, the server re-establishes the connection and delivers the response
- HTTP can deliver any sort of data, as long as the two computers are able to read it
- **Stateless protocol:**
 - The client and server know about each other just during the current request
 - If it closes and the two computers want to connect again, they handle the connection as if it was the first time

How does it work?

How the Web works? And How Http makes that possible?

The Request Response Cycle

The Globe

Web Server





HTTP Client Request Methods

(most common in bold)

GET

- Retrieve a representation of the resource

HEAD

- Like GET but without content

POST

- Submit data to be processed, includes the data in the message

PUT

- Upload a representation

DELETE

- Deletes the resource

TRACE

- Send the received request to see if it changed on the way

OPTIONS

- Request a list of active functions supported by the server

CONNECT

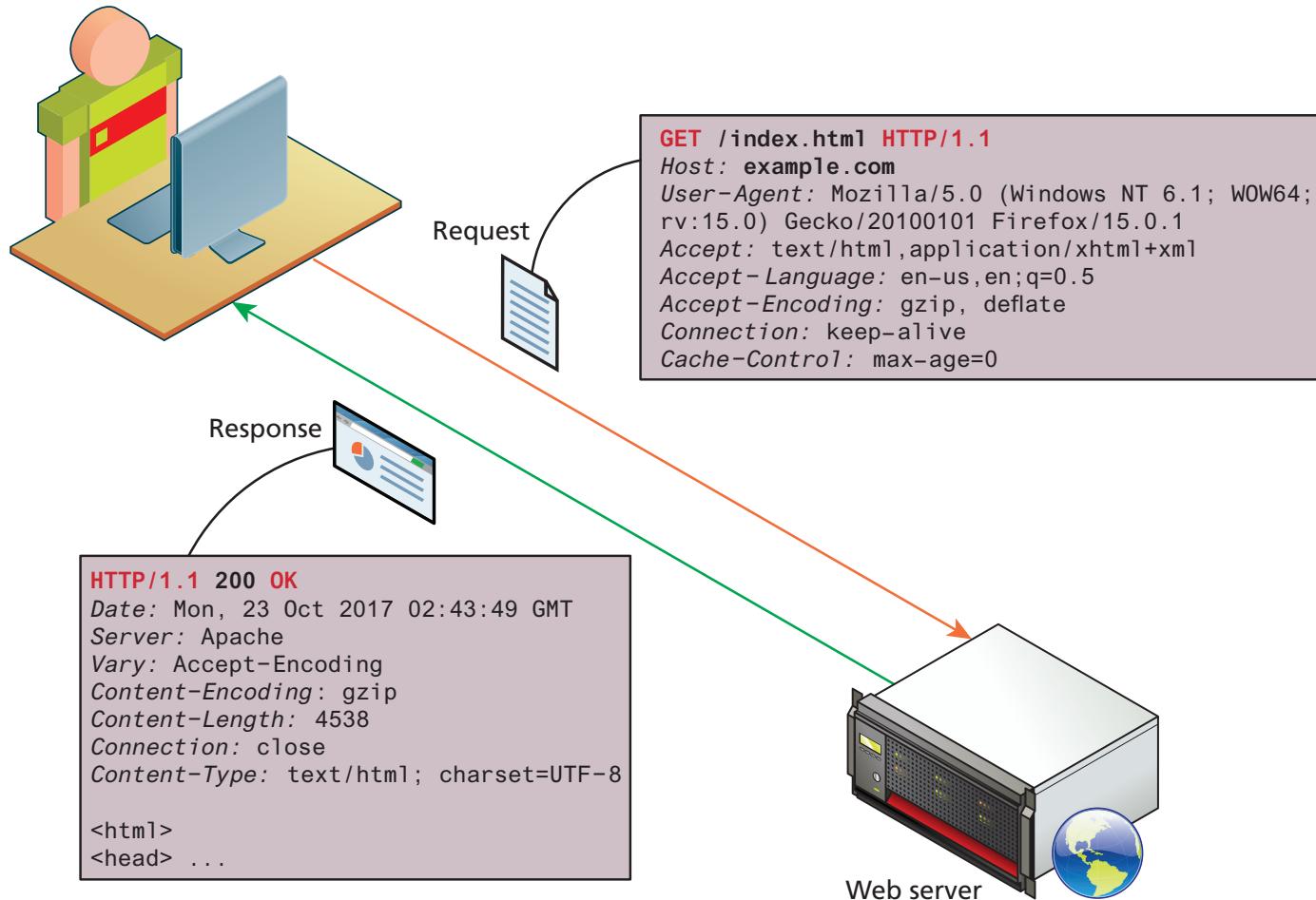
- Used to simplify SSL by translating communications to TCP/IP

PATCH

- To slightly modify a resource

Hypertext Transfer Protocol

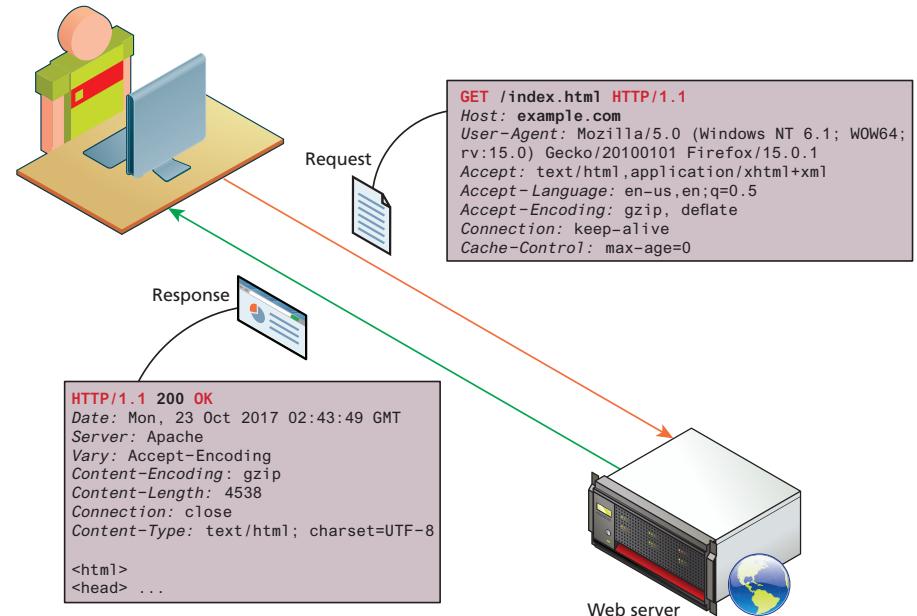
Headers



Hypertext Transfer Protocol

Headers

- **Request headers** include data about the client machine.
- **Response headers** have information about the server answering the request and the data being sent





HTTP Request

GET /index.html HTTP/1.1

Request Line

Date: Thu, 20 May 2004 21:12:55 GMT

General Headers

Connection: close

Host: www.myfavoriteamazingsite.com

Request Headers

From: joebloe@somewebsitesomewhere.com

Accept: text/html, text/plain

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

Entity Headers

**HTTP
Request**

Message Body



HTTP Response

HTTP/1.1 200 OK

Status Line

Date: Thu, 20 May 2004 21:12:58 GMT

General Headers

Connection: close

Server: Apache/1.3.27

Response Headers

Accept-Ranges: bytes

Content-Type: text/html

Entity Headers

Content-Length: 170

Last-Modified: Tue, 18 May 2004 10:14:49 GMT

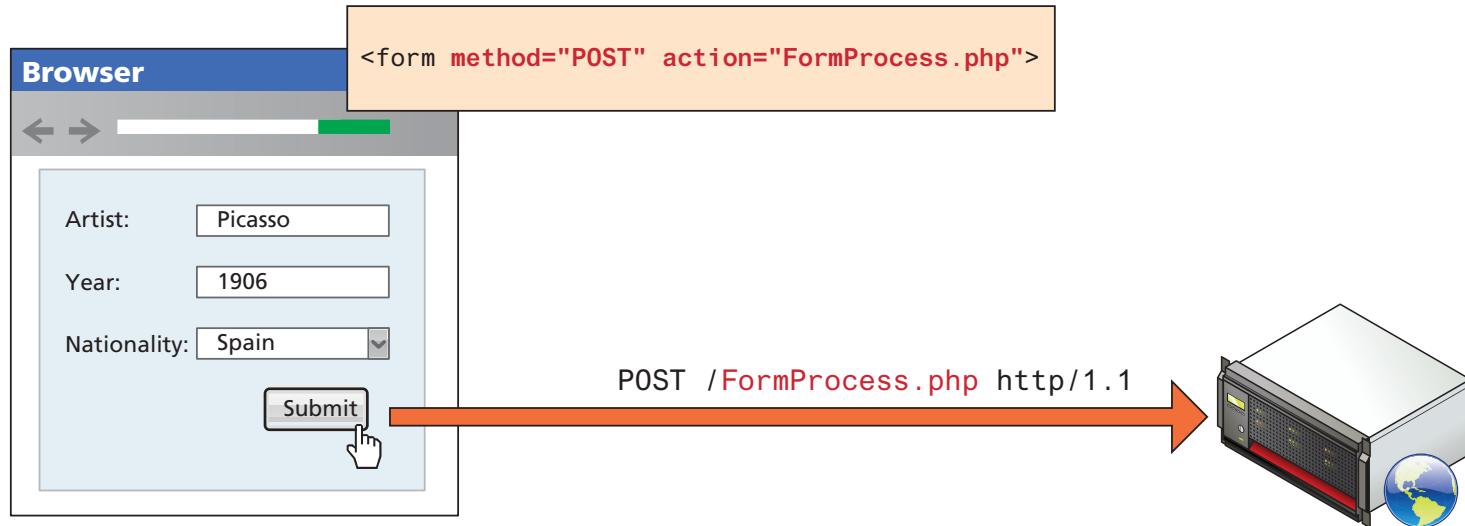
**HTTP
Response**

```
<html>
<head>
<title>Welcome to the Amazing Site!</title>
</head>
<body>
<p>This site is under construction. Please come
back later. Sorry!</p>
</body>
</html>
```

Message Body

Hypertext Transfer Protocol

Request Methods





Hypertext Transfer Protocol

Response Codes

- 2## codes are for successful responses,
- 3## are for redirection-related responses,
- 4## codes are **client** errors,
- 5## codes are **server** errors.



Hypertext Transfer Protocol

(Some) Response Codes

200: OK

301: Moved Permanently

304: Not Modified

307: Temporary redirect

400: Bad Request

401: Unauthorized

404: Not found

414: Request URI too long

500: Internal server error



HTTPS (HTTP Secure)

- HTTPS is HTTP-within-SSL/TLS
- SSL (TLS) establishes a secured bidirectional tunnel for arbitrary binary data between two hosts
- HTTP is meant to run over a bidirectional tunnel for arbitrary binary data; when that tunnel is a SSL/TLS connection, then the whole is called "HTTPS"



HTTPS

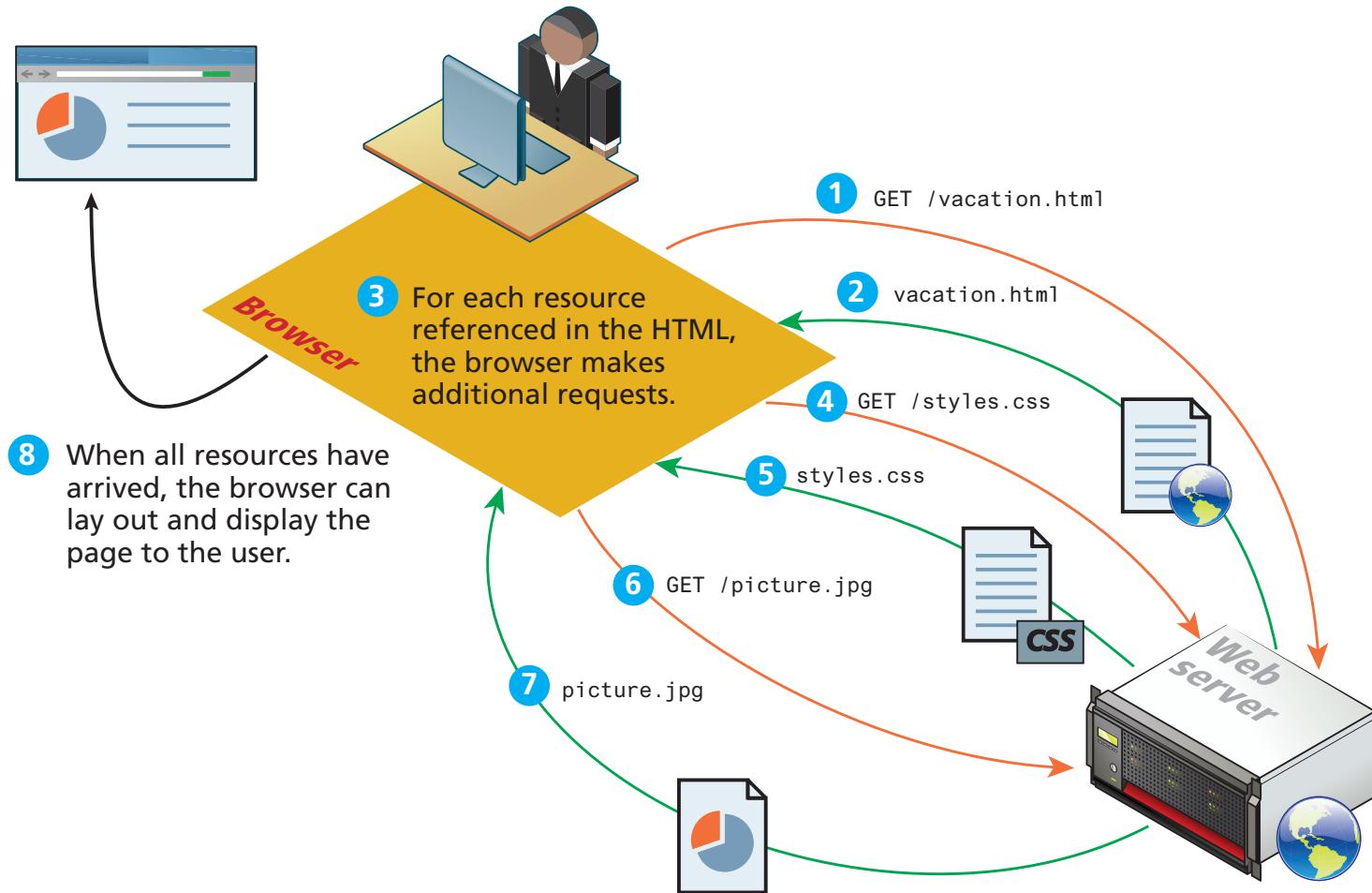
- Uses TLS (Transport Layer Security) / SSL (Secure Sockets Layer) to encrypt the transmission of data
- Syntactically identical to HTTP
- Only needs one party to be authenticated
- Relies on certificates issued by several companies
- Usually runs on port 443



Web Browsers

Web Browsers

Fetching a Web Page





Web Browsers

Fetching a Web Page – Load Times and Cascades

The screenshot shows a web browser window displaying the "Fundamentals of Web Development" website. The page title is "Chapter 3: Introduction to CSS". The browser's developer tools are open, specifically the Network tab, which shows the network traffic for the page load. The table below details the requests made:

Status	Method	File	Domain	Cause	Type	Transferred	Size	0 ms	640 ms	1.28 s	1.92 s	2.56 s	3.20 s
▲ 304	GET	thumbs_chapter1-29.png	funwebdev.com	img	png	—	8.95 KB				→ 142 ms		
▲ 304	GET	chapter3-95.png	funwebdev.com	imageset	png	—	47.66 KB				→ 129 ms		
▲ 304	GET	slide-javascript-50x50.jpg	funwebdev.com	imageset	jpeg	—	2.12 KB				→ 138 ms		
▲ 304	GET	dog-adoption-50x50.jpg	funwebdev.com	imageset	jpeg	—	2.09 KB				→ 139 ms		
▲ 304	GET	Dollarphotoclub_92872465-web-50x50.jpg	funwebdev.com	imageset	jpeg	—	1.88 KB				→ 134 ms		
▲ 304	GET	exam-takers-50x50.jpg	funwebdev.com	imageset	jpeg	—	2.39 KB				→ 121 ms		
▲ 304	GET	adoptions2015-50x50.jpg	funwebdev.com	imageset	jpeg	—	2.08 KB				→ 152 ms		
● 200	GET	like.php?href=http://funwebdev.com/samples/chapters/cha...	www.facebook.com	subdocument	html	8.40 KB	26.91 KB				→ 114 ms		
○ 200	GET	i-u7L8opqbV.js	www.facebook.com	xhr	js	cached	389.56 KB				→ 24 ms		
▲ 304	GET	analytics.js	www.google-analytics.com	script	js	11.32 KB	27.15 KB				→ 24 ms		
● 200	GET	collect?v=1&v=j47&a=101546702&t=pageview&l_s=1&...	www.google-analytics.com	img	gif	35 B	35 B				→ 75 ms		
○ 200	GET	like.php?href=http://funwebdev.com/samples/chapters/cha...	www.facebook.com	subdocument	html	8.37 KB	26.87 KB						
○ 200	GET	i-u7L8opqbV.js	www.facebook.com	xhr	js	cached	389.56 KB						
○ 200	GET	style.css?ver=4.8.1	funwebdev.com	stylesheet	css	cached	273.26 KB						
○ 200	GET	rgs.css?ver=4.8.1	funwebdev.com	stylesheet	css	cached	2.29 KB						



Tools to View HTTP Traffic

- Chrome/Dev. Tools - Network (fav. amongst web-devs)
- Fiddler (Windows), Charles Proxy (Mac)
 - web debugging proxies (not only inspect, but offer expanded capabilities to interact with your site)
- Command line tools:
 - curl, tcpdump, tshark (wireshark with GUI)



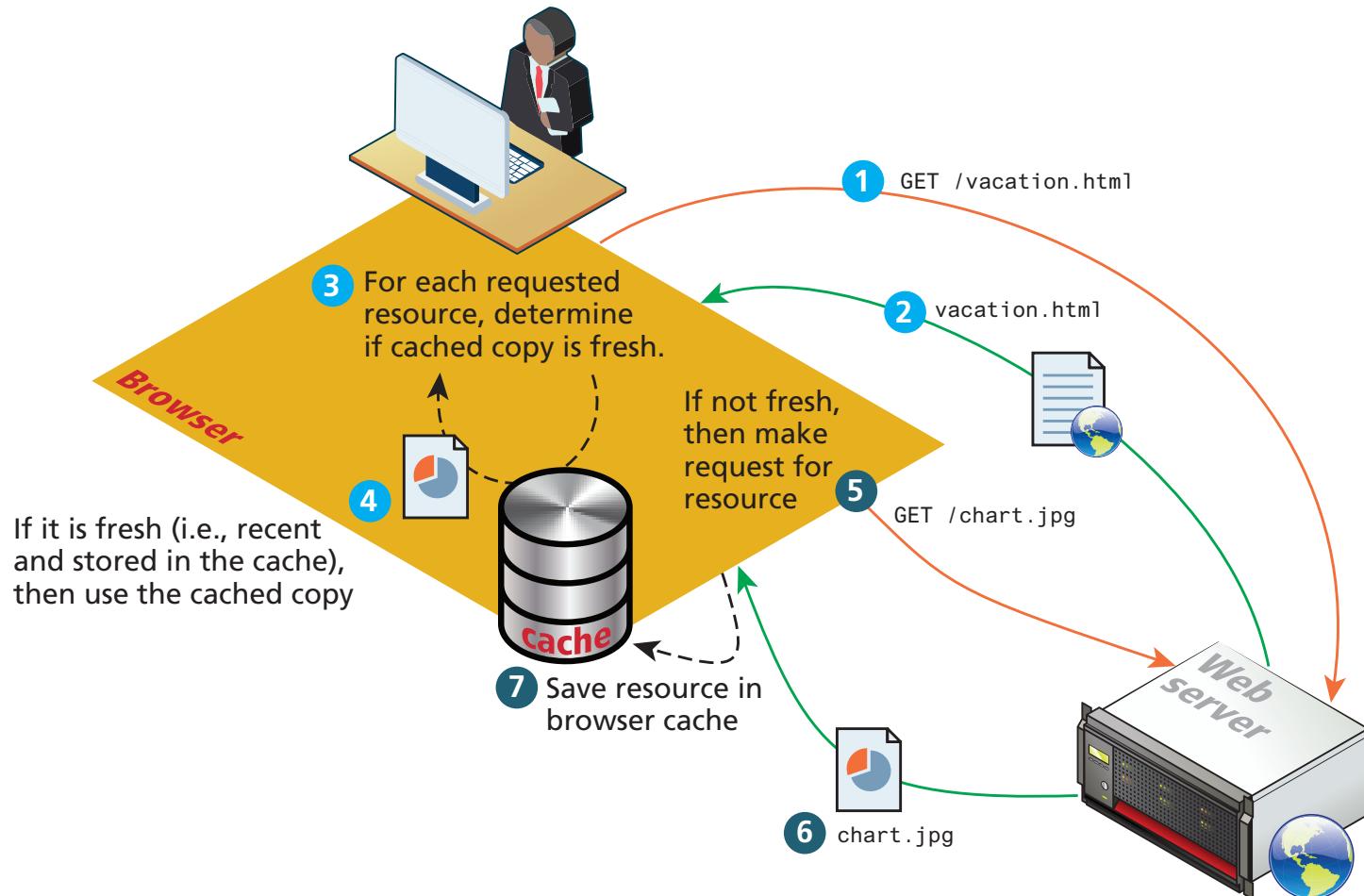
Web Browsers

Browser Rendering

- Interpreting the entire HTML markup together with the images and other assets into a grid of pixels for display within the browser window is called rendering the webpage.
- Implemented differently for each browser (Firefox, Chrome, Safari, Explorer, and Opera)

Web Browsers

Browser Caching





Web Browsers

Browser Features

- search engine integration,
 - URL autocompletion,
 - Form autocompletion,
 - cloud caching of user history/bookmarks,
 - phishing website detection,
 - secure connection visualization,
- and much more



Web Browsers

Browser Extensions

Can change what is shown to the end user. Newer challenge for web developers.

For developers, extensions like

- Firebug (discontinued) and
- Yslow

For the general public:

- Adblock
- Third Party Plugins