Introduction to WWW, Web Protocols and URLs

Common Terms

- Internet vs. Web
- Web Browsers
- URL
- Web Server
- DNS
- HTTP Protocol
- HTTPS

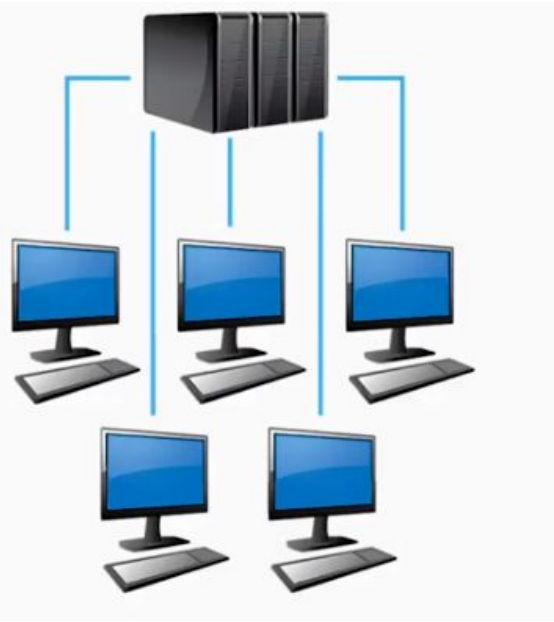
Introduction to WWW, Web Protocols and URLs

Internet vs. WWW

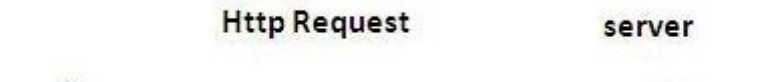
Introduction to WWW, Web Protocols and URLs

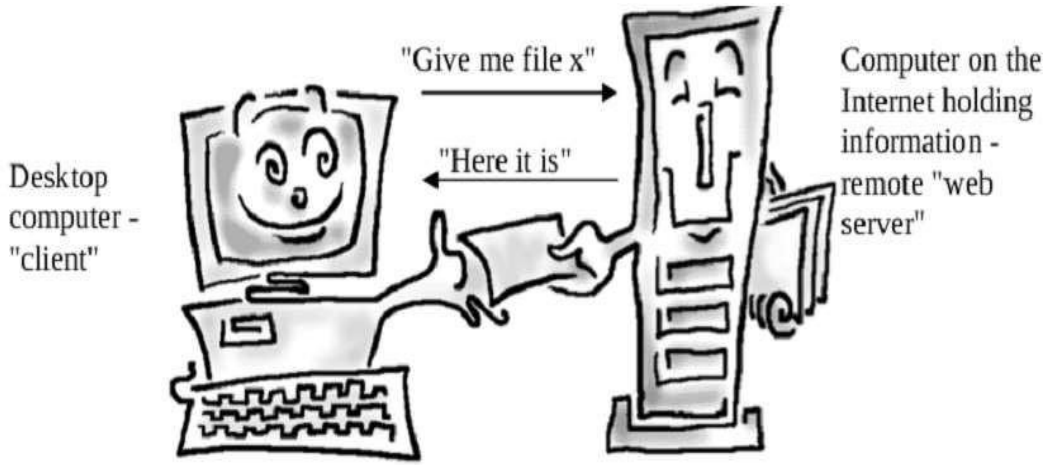
How does WWW work?

1. Client/Server Architecture



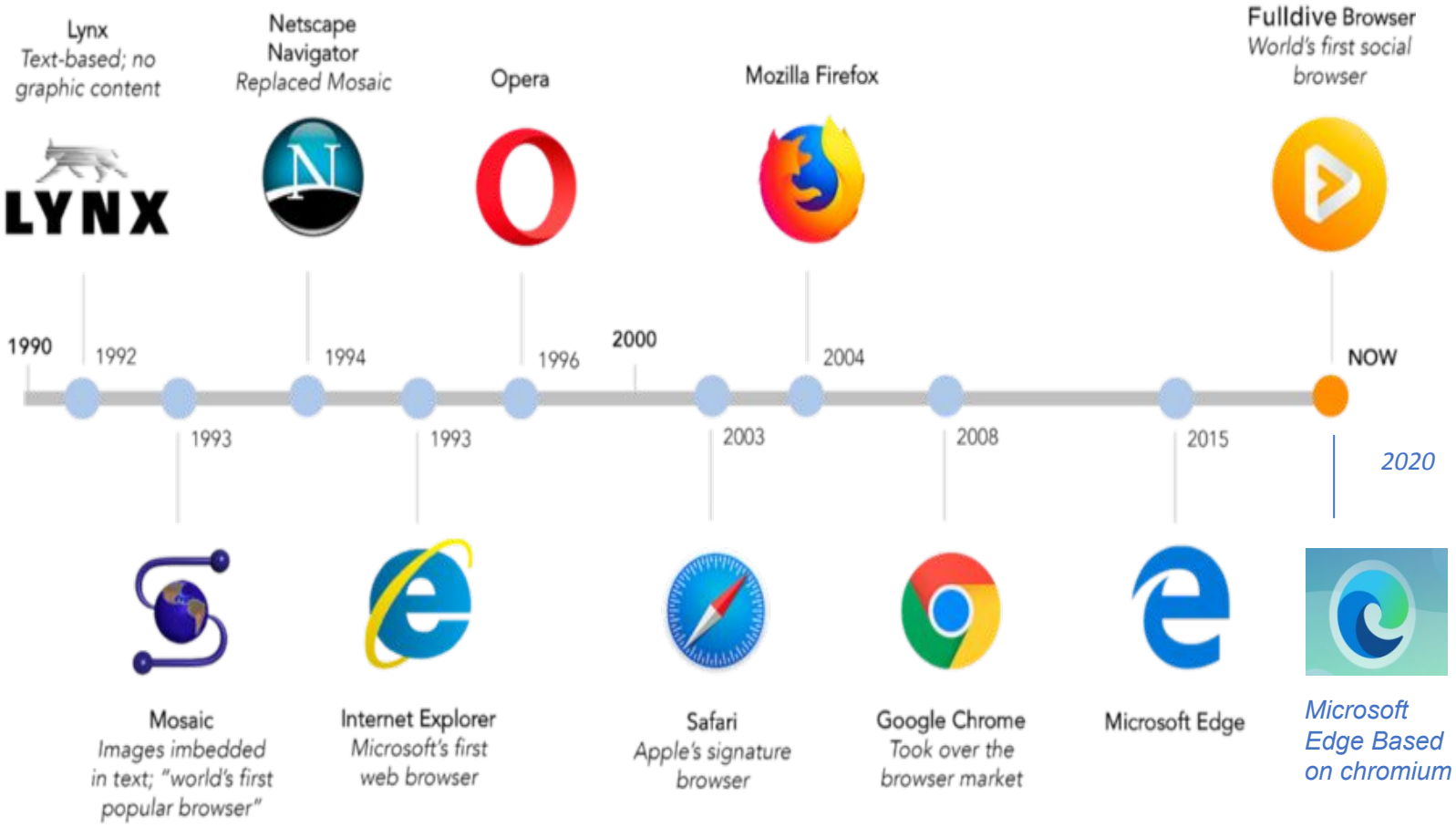
2. Request/Response Pattern





Introduction to WWW, Web Protocols and URLs

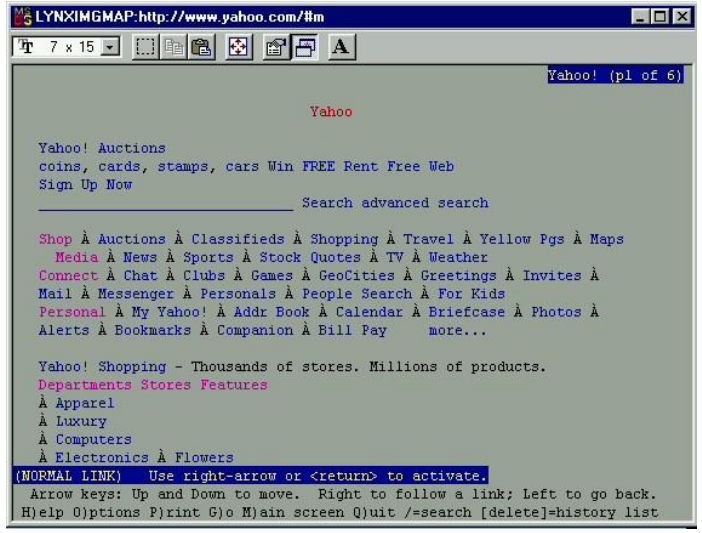
History of Web Browsers



Source:A Brief History of Browsers https://medium.com/@fulldive/a-brief-history-of-browsers-57669527c0cf

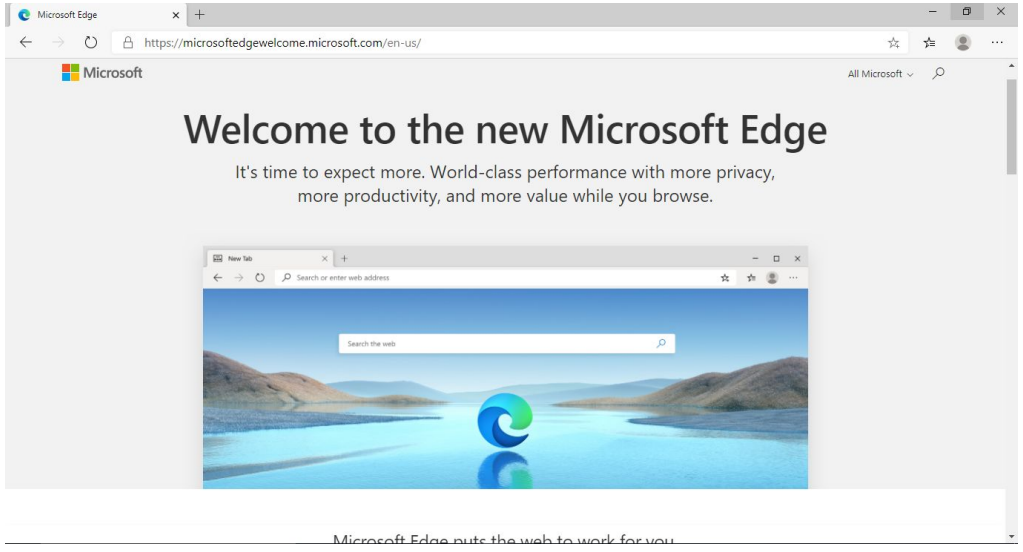
Introduction to WWW, Web Protocols and URLs

Browser Evolution



Lynx – A text based browser

Mosaic – the first graphical browser



Source: Browser Museum
http://www.donmouth.co.uk/web_design/browsermuseum/browsermuseum.html

URLs

- URL stands for Uniform Resource Locator
- General form:

  scheme:object-address

- For the http protocol, the object-address is:

  fully qualified domain name/doc path

Example:

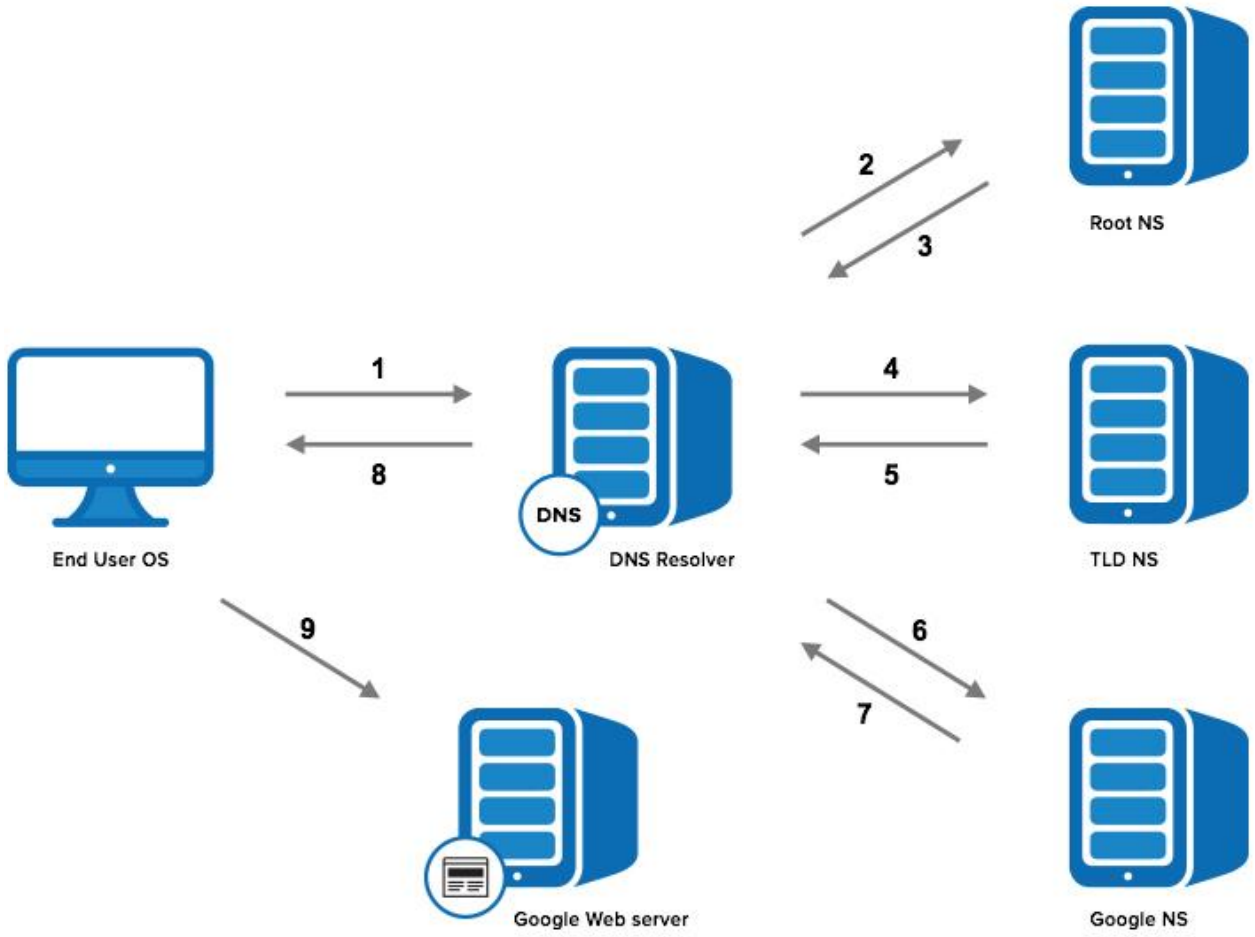
https://www.amazon.com/international-sales-offers.html

Web Servers

- General Web Server Characteristics
  - Web servers have two main directories:
    1. Document root (servable documents)
    2. Server root (server system software)
  - Document root is accessed indirectly by clients
    - Its actual location is set by the server configuration file
    - Requests are mapped to the actual location
- Popular Examples
  - Apache
  - IIS

Introduction to WWW, Web Protocols and URLs

Domain Name Service



Step 1: OS Recursive Query to DNS Resolver
Step 2: DNS Resolver Iterative Query to the Root Server
Step 3: Root Server Response
Step 4: DNS Resolver Iterative Query to the TLD Server
Step 5: TLD Server Response
Step 6: DNS Resolver Iterative Query to the Google.com NS
Step 7: Google.com NS Response
Step 8: DNS Resolver Response to OS
Step 9: Browser Starts TCP Handshake

Introduction to WWW, Web Protocols and URLs

How to get your own website?

PES UNIVERSITY
CELEBRATING 50 YEARS

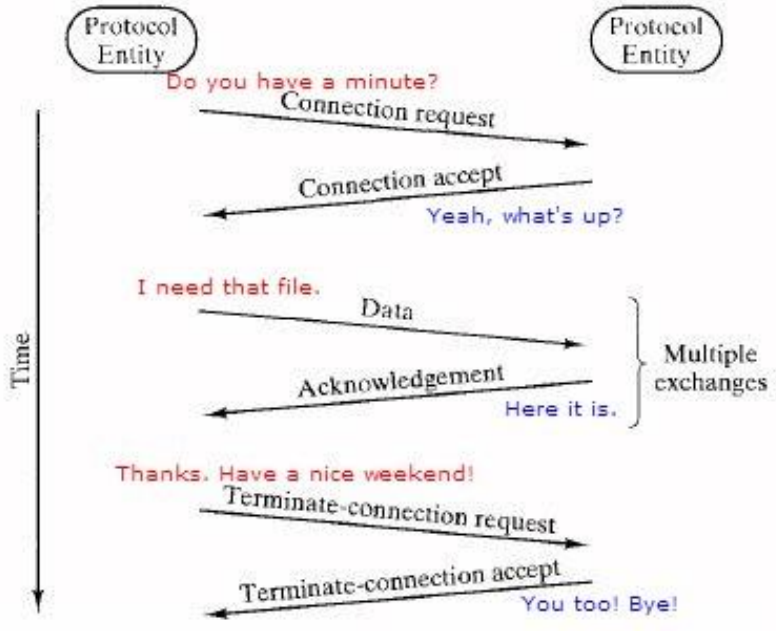
Steps:

1. Choose a domain name
2. Register a domain and sign up with web hosting
3. Set up a website using WordPress/Name cheap/Go Daddy (through web host)
4. Customize your website design and structure
5. Add pages and content to your website

Introduction to Web Protocols and HTTP

What is a Protocol?

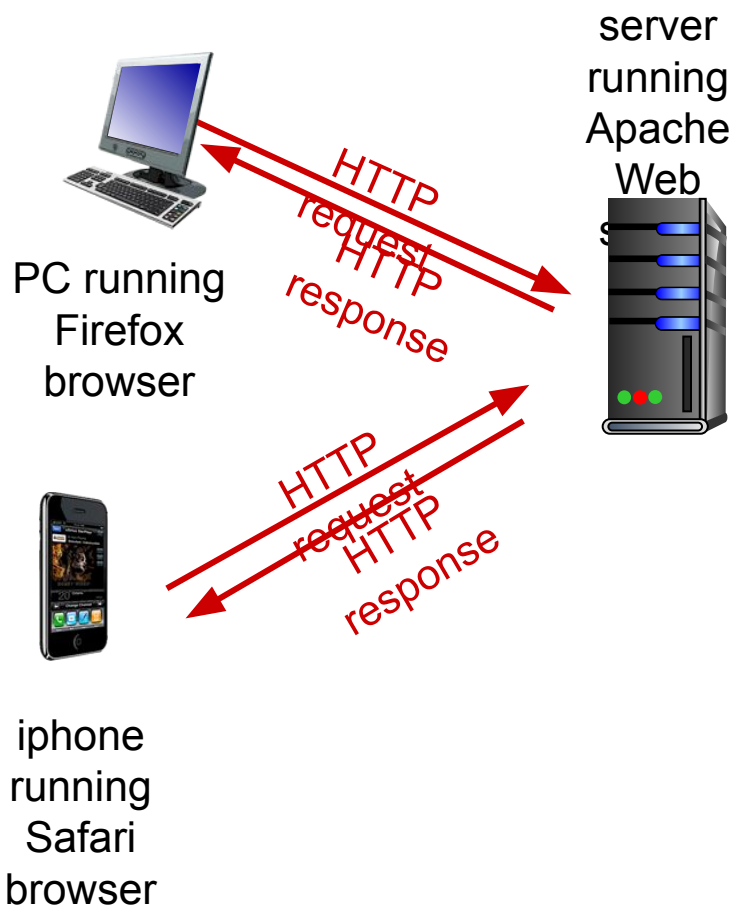
- A protocol is a set of rules and guidelines for communicating data.
- Different applications use different protocols
- The web, in particular, uses multiple protocols to communicate.
- The most important and visible protocols are HTTP and HTTPS.



HTTP Overview

HTTP: HyperText Transfer Protocol

- Application Protocol used by the Web
- Client/Server model
  - *Client:* browser that requests, receives, and "displays" Web Objects
  - *Server:* Web server sends Web Objects (using HTTP protocol) in response to requests



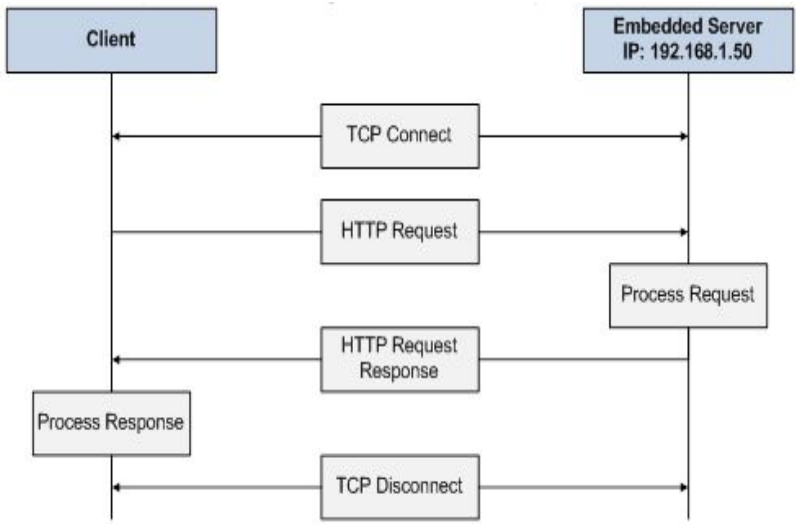
HTTP Overview…(cntd.)

uses TCP:

- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- HTTP messages (application-layer protocol messages) exchanged between browser (HTTP client) and Web server (HTTP server)
- TCP connection closed

HTTP is "stateless"

- server maintains no information about past client requests



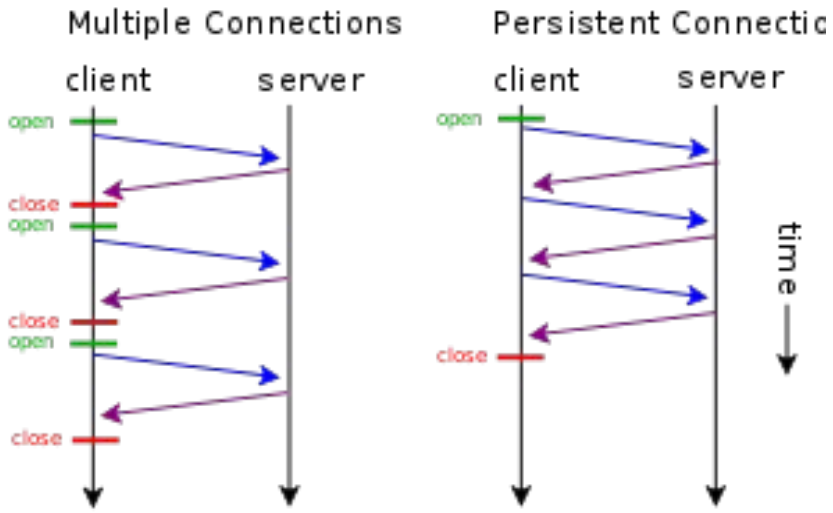
HTTP Connections

non-persistent HTTP

- at most one object sent over TCP connection
  - connection is then closed
- downloading multiple objects required multiple connections

persistent HTTP

- multiple objects can be sent over single TCP connection between client, server



HTTP Requests

- HTTP request is a *request line*, followed by zero or more *request headers*
- Request line: <method> <uri> <version>
  - <version> is HTTP version of request (HTTP/1.0 or HTTP/1.1)
  - <uri> is typically URL for proxies, URL suffix for servers.
  - <method> is either GET, POST, OPTIONS, HEAD, PUT, DELETE, or TRACE.
- Request Header
- Blank line (CRLF)
- Message Body

```
GET /test.html HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 4.01;
Windows 98)
Host: euro.ecom.cmu.edu
Connection: Keep-Alive
CRLF (\r\n)
```

HTTP Request Methods

- HTTP methods:
  - GET: Retrieve static or dynamic content
  - POST: Send content to server through request body
  - OPTIONS: Get server or file attributes
  - HEAD: Fetches only header field without any response body
  - PUT: Write a file to the server
  - DELETE: Delete a file on the server

HTTP Response

- HTTP response is a response line followed by zero or more response headers.
- Response line:
- <version> <status code> <status msg>
  - <version> is HTTP version of the response.
  - <status code> is numeric status.
- Response headers:

  <header name>: <header data>
  - Provide additional information about response
  - Content-Type: MIME type of content in response body.
  - Content-Length: Length of content in response body.

```
HTTP/1.1 200 OK
Date: Thu, 22 Jul 1999 04:02:15 GMT
Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)
Last-Modified: Thu, 22 Jul 1999 03:33:21 GMT
ETag: "48bb2-4f-37969101"
Accept-Ranges: bytes
Content-Length: 79
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
CRLF
<html>
<head><title>Test page</title></head>
<body>
<h1>Test page</h1>
</html>
```

HTTP Response : Status Codes

- Three-digit number; first digit specifies the general status
  - 1 => Informational
  - 2 => Success
  - 3 => Redirection
  - 4 => Client error
  - 5 => Server error
- <status msg> is corresponding English text.
  - 200 OK => Request was handled without error
  - 403 Forbidden => Client lacks permission to access file
  - 404 Not found => Server couldn’t find the file.

HTTP Secure (HTTPS)

- A common security attack
- Need to encrypt data to save it from such attacks