INLS 672
Web Development 2

Web Foundations:
HTTP, Browser, Servers

UNC
SCHOOL OF INFORMATION
AND LIBRARY SCIENCE

Joan Boone
jpboone@email.unc.edu

Part 1: URLs, HTTP

Part 2: Browsers and Servers

Web Foundations

Tim Berners-Lee conceived of a simple, stateless, request-response web protocol that did not require a persistent connection.

The foundational building blocks that achieve this goal are

- Uniform Resource Locator (URL) – a notational scheme for accessing resources anywhere on the Internet
- HTTP – a protocol for transporting messages over the network
- Hyper Text Markup Language (HTML) – a standard markup language for creating web documents

Uniform Resource Locator (URL)

URLs, often called web addresses, are a notational scheme for accessing Internet resources.

Example: **`https://en.wikipedia.org/wiki/URL`**

- *Scheme* is the underlying protocol, e.g., https, http, ftp
- *Host* is the registered domain name or IP address for the web server that hosts the referenced resource
- *Port* is an optional component that specifies the port number that the server listens on. The default port for HTTP is 80, and for HTTPS it is 443. The port number appears after a colon following the host name. Example: **`https://www.example.com:80/path/`**
- *Path* defines a sequence of segments, similar to a file system path, that specifies how the server locates the resource

MDN Web Docs: What is a URL?

Uniform Resource Locators: Query Parameters and Fragments

Example: **`https://www.google.com/search?q=url`**

Query is an optional string, separated from the previous part by a question mark (?). By convention, the string contains one or more attribute-value pairs (or parameters), where each pair is separated by an ampersand (&)

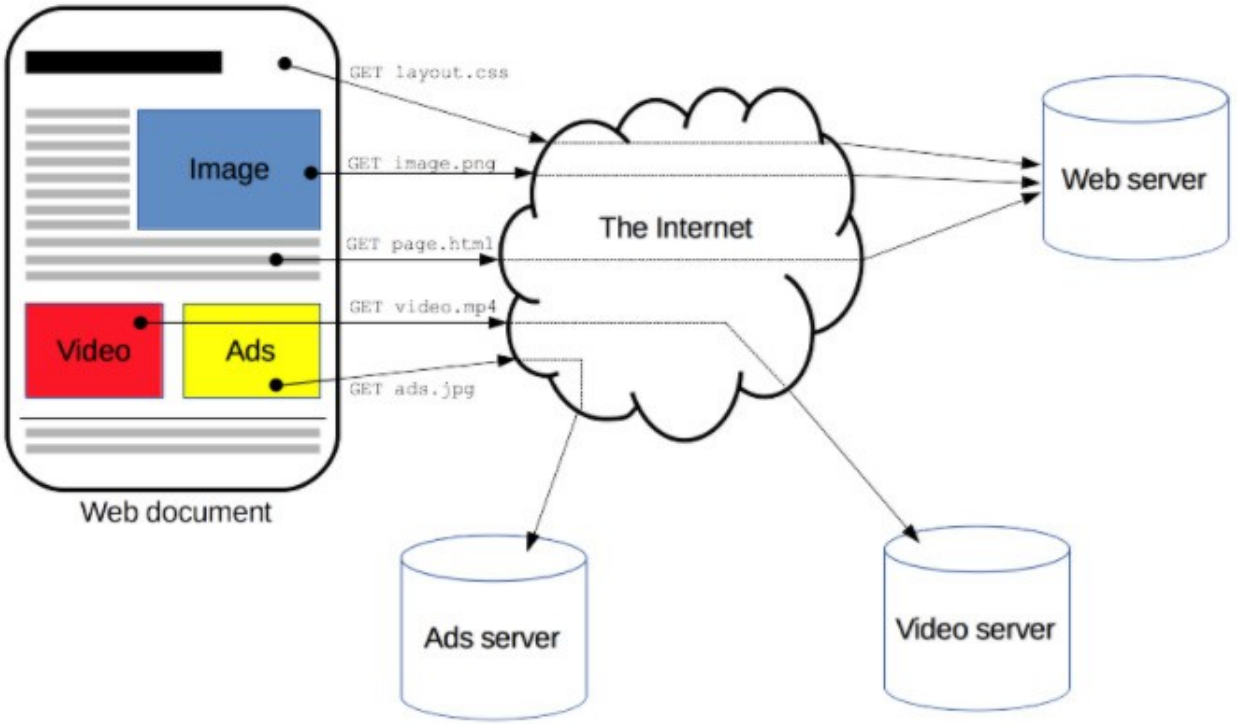
Example: **`https://en.wikipedia.org/wiki/URL#Syntax`**

Fragment *(or anchor)* s an optional string, separated from the previous part by a hash mark (#). The fragment contains a fragment identifier which is often used in HTML documents to identify a specific section on a web page that can be linked to.

MDN Web Docs: What is a URL?

HTTP Overview

- HTTP is a protocol that allows access to network resources, and is the foundation of any web data exchange.
- It uses a client-server protocol which means requests are made by the client, often a web browser, and servers respond with the resources to satisfy the request.



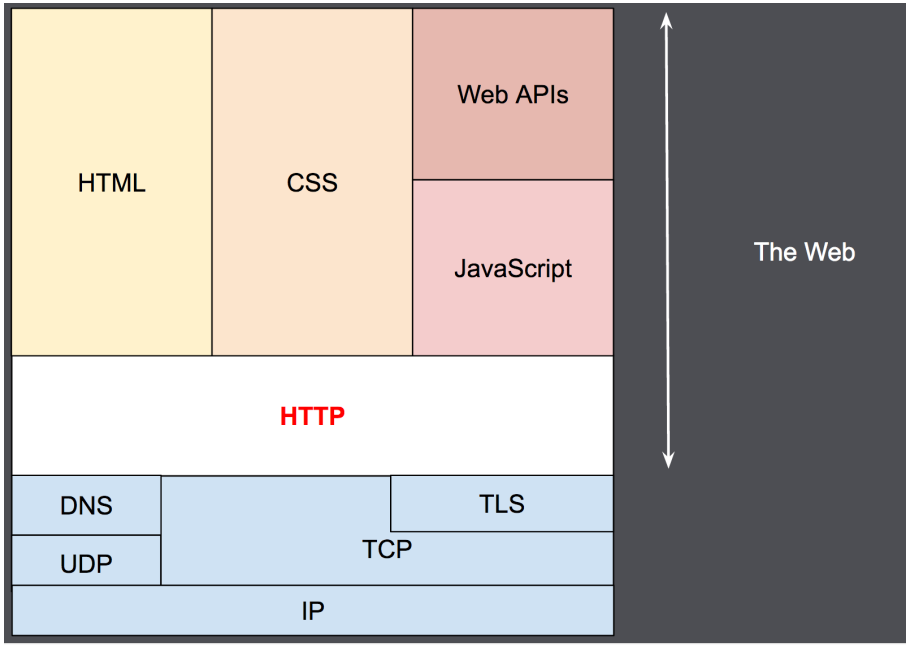
MDN Web Docs: An overview of HTTP

HTTP Overview

HTTP (Hypertext Transfer Protocol) is an application layer protocol that is used to fetch HTML documents, as well as all of the resources associated with a web page, e.g., images, video, scripts, style sheets.

HTTP uses the ***TCP*** (Transmission Control Protocol) to establish a network connection between a client and a server so that data can be exchanged.

For secure communication over a network, the ***HTTPS*** protocol is use to send data over a Transport Layer Security (TLS)-encrypted TCP connection.



MDN Web Docs: An overview of HTTP

HTTP Characteristics

HTTP is simple

HTTP is generally designed to be simple and human-readable, so that HTTP messages can be read and understood by humans.

HTTP is extensible

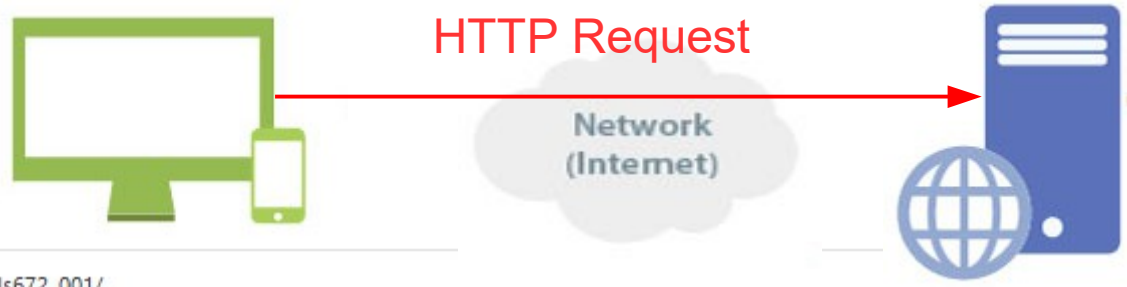
With HTTP/1.0, HTTP headers were introduced, which made the protocol easy to extend. New functionality can be added by simple agreement between a client and a server about the semantics of new headers. HTTP/2 became available in 2015, and supports many performance improvements.

HTTP is stateless, but not session-less

HTTP is stateless in that there is no link between 2 successive requests made on the same connection. The 2 requests have no knowledge of each other, or the subsequent responses. In order to retain information across a session, HTTP cookies can be used to allow session creation on each HTTP request to share the same context

MDN Web Docs: An overview of HTTP

HTTP Requests



GET https://ils.unc.edu/courses/2021_spring/inls672_001/

Status	200 OK
Version	HTTP/1.1
Transferred	5.69 KB (5.31 KB size)

Response Headers (392 B) Raw

Request Headers (1.461 KB) Raw

```
GET /courses/2021_spring/inls672_001/ HTTP/1.1
Host: ils.unc.edu
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Cookie: _ga=GA1.2.879385170.1589044379; __utma=109521629.879385170.1589044379.1591312529.1591322663.2; AMCV_2E27401053DB4FE50A4
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

HTTP requests contain the following elements

- An HTTP method, such as GET or POST, defines the operation to perform. For example, use GET to fetch a web page, or POST to post the values in an HTML form
- Path of the resource to fetch. The URL is stripped of elements that are obvious from context, e.g., protocol and domain
- Version of the HTTP protocol
- Optional headers that convey additional information for servers

HTTP Response

HTTP responses contain the following elements

- A status code indicating if the request was successful, or not, and why
- Version of the HTTP protocol
- A status message that describes the status code
- HTTP headers, like those for requests
- Optionally, a body containing the fetched resource

HTTP Response Status Codes

HTTP response status codes indicate whether a specific HTTP request has been successfully completed.

Status codes are grouped into 5 categories:

- Informational responses (1xx) provide information about further processing
- Successful responses (2xx) indicate that the request was successfully completed and that the requested resource is being sent to the client
- Redirects (3xx) indicate that additional actions are required to satisfy the original request. Normally this involves a redirection: the browser is instructed to resubmit the request to another URL
- Client errors (4xx) indicate an abnormal condition with the client request. Common examples are a malformed request, an authorization challenge, access denial, or the server's inability to find the requested resource
- Server errors (5xx) indicate a server problem that prevents it from satisfying an otherwise valid request

MDN Web Docs: HTTP response status codes

HTTP Common Status Codes

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
HTTP/1.1 300 Multiple Choices
HTTP/1.1 301 Moved Permanently
HTTP/1.1 302 Found
HTTP/1.1 304 Not Modified
HTTP/1.1 307 Temporary Redirect
HTTP/1.1 400 Bad Request
HTTP/1.1 401 Unauthorized
HTTP/1.1 403 Forbidden
HTTP/1.1 404 Not Found
HTTP/1.1 500 Internal Server Error
HTTP/1.1 501 Not Implemented
HTTP/1.1 503 Service Unavailable
HTTP/1.1 550 Permission denied
```

Complete definition of status codes: RFC 2616 Section 10

Part 1: HTTP

Part 2: Browsers and Servers

Web Browsers

Web browsers are the most common example of a client in a web application. The main responsibilities of a browser are to

- Generate and submit requests to web servers on the user's behalf
- Accept responses from web servers and interpret them to produce a visual representation for the user, and render the results in a browser

Additionally, browsers are responsible for

- Caching data
- Authentication and authorization
- State maintenance, e.g., via cookies
- Requesting any supporting data required by a web page, e.g., images, style sheets, scripts.

Browser Usage and Statistics

Browser usage and popularity has changed significantly over the last decade, due in part to the proliferation of mobile devices.

Here are a few sources that track browser usage:

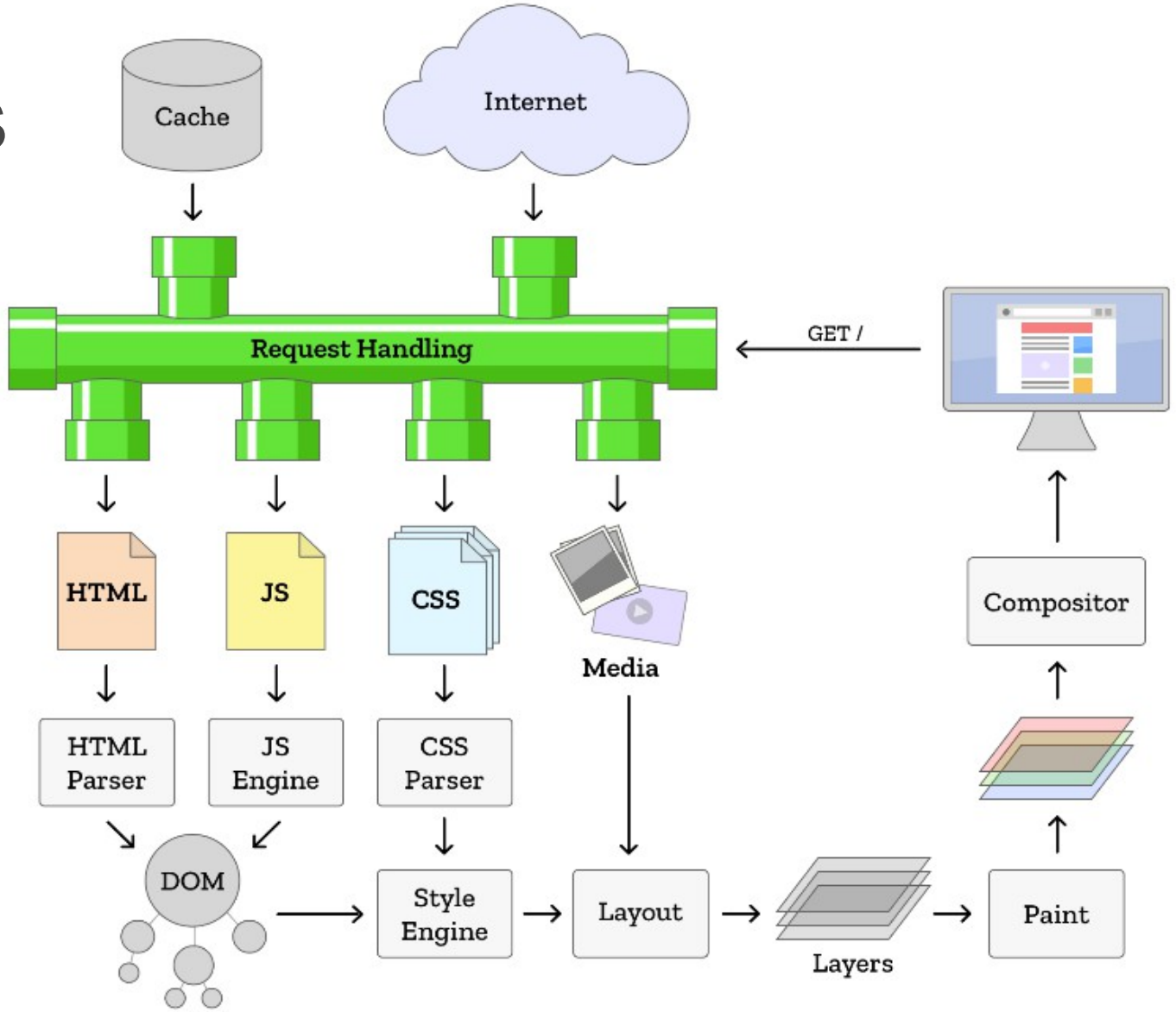
- w3Schools summarizes the most popular browsers since 2002
- StatCounter Global Stats: Browser Market Share Worldwide
- W3Counter: Web Browser Usage Trends

Although Chrome clearly dominates at this time, it is always highly recommended when developing web applications that you test with multiple browsers that run on desktop/laptop and mobile devices.

Browser support for front-end technologies

- Modern browsers make every effort to stay current with the latest standards but that is not always the case.
- Refer to the Can I use... website to determine whether a given browser supports a specific HTML, CSS, or JavaScript feature.

How Browsers Work



Mozilla Hacks: Quantum Up Close: What is a browser engine?

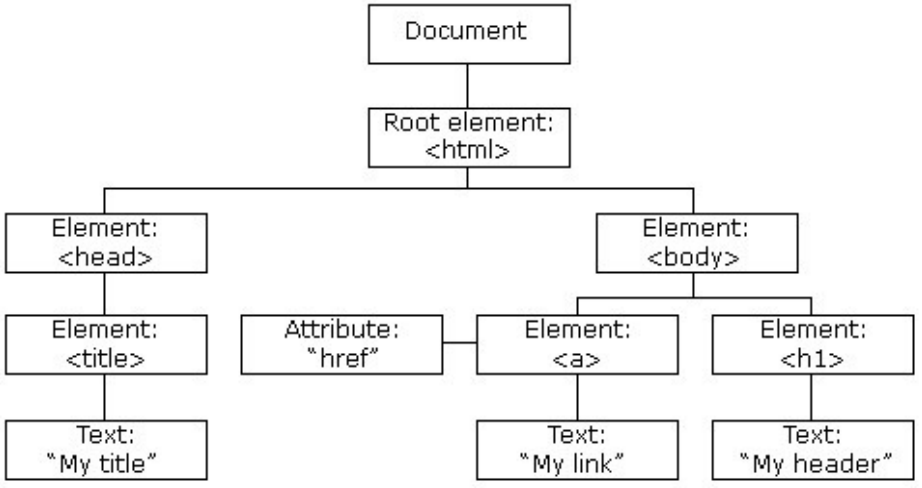
How Browsers Work: HTML Parsing and the DOM

- The DOM is a W3C standard that defines how to access and change every element in an HTML document. It “allows programs and scripts to dynamically access and update the content, structure and style of documents.”
- Every HTML page has an underlying representation called the Document Object Model (DOM). This is a hierarchical tree structure where everything in your HTML document is represented as a node.

HTML Document

```
<html>
 <head>
  <title>My title</title>
 </head>
  <body>
   <a href="http://mylink.com”>My link</a>
   <h1>My header</h1>
  </body>
</html>
```

w3schools
HTML DOM Tree



MDN Web Docs: Populating the page: how browsers work

Web Servers

'Web server' can refer to both hardware and software entities

- As a ***hardware*** entity, it refers to a computer that hosts web server software and website resources (HTML, CSS, JavaScript, images). It is connected to the Internet and supports data interchange among network-connected devices.
- As a ***software*** entity, a web server includes an HTTP server which is software that understands HTTP requests, and delivers associated content to clients.

MDN Web Docs: What is a web server?

Web Servers: Static vs. Dynamic

- A ***static web server*** consists of a computer (hardware) with an HTTP server (software). It is called "static" because the server sends its hosted files as-is to your browser.
- A ***dynamic web server*** consists of a static web server plus extra software, most commonly an application server and a database. It is called "dynamic" because the application server updates the hosted files before sending content to your browser via the HTTP server. Often, the application server will access a data store to obtain the most current content to populate a web page.

MDN Web Docs: What is a web server?

Web Server Technology

- Usage of web servers for websites
- Web Server Survey (December 2020)
- Stack Overflow: 2020 Developer Survey
- Web Server Usage Distribution

  in Top 1 Million Sites