

08. How the Web Works

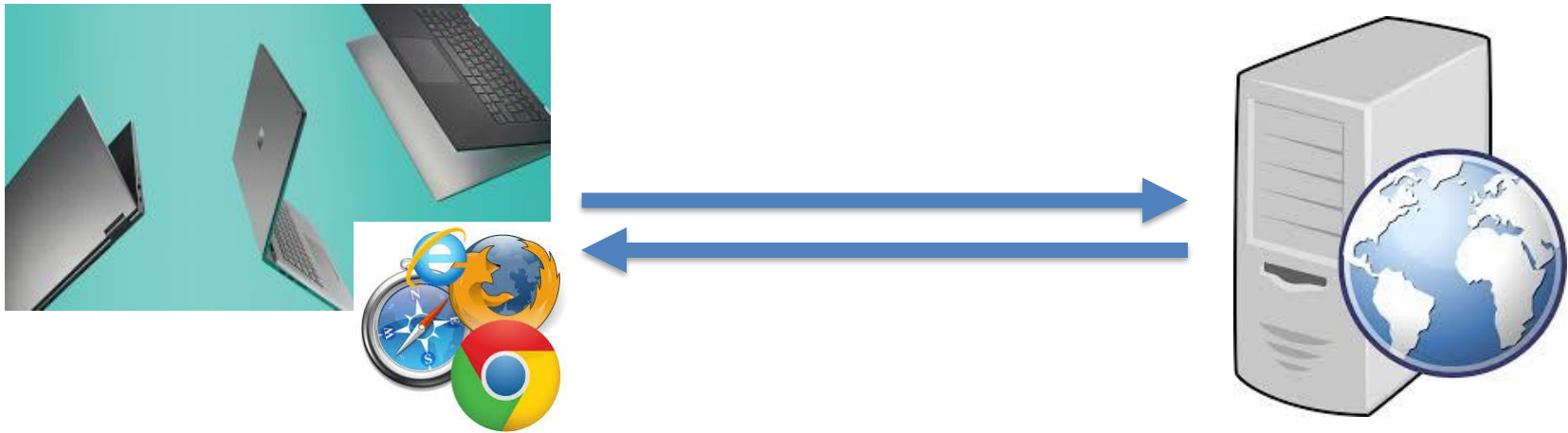
Blase Ur and David Cash
January 24th, 2020
CMSC 23200 / 33250



THE UNIVERSITY OF
CHICAGO

Your interface to the web

- Your web browser contacts a web server



A 10,000 Foot View of Technologies

- Where things run:



HTML / CSS

JavaScript
(Angular/React)

Browser Extensions



HTTP



CGI / PHP / Django
/ Node.js / Ruby on Rails

Databases (MySQL)

The Anatomy of a Webpage

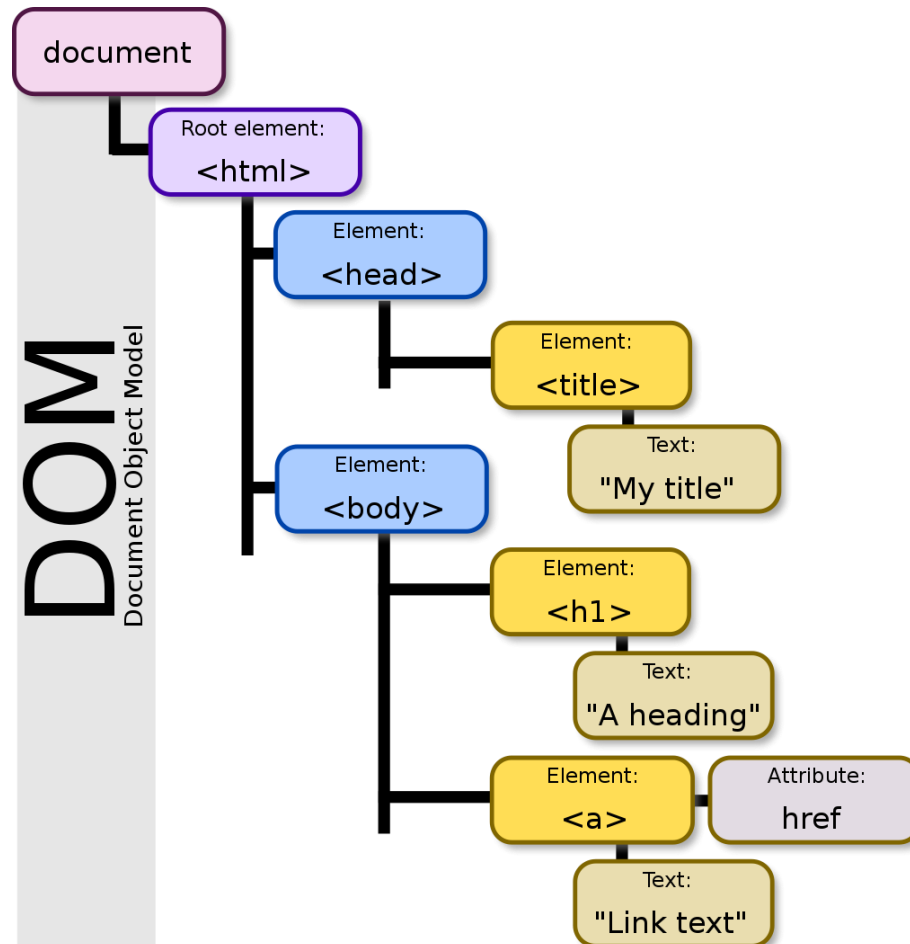
- view-source:https://www.cs.uchicago.edu/
- HTML (hypertext markup language)
 - Formatting of a page
 - All sorts of formatting: `
` `<div></div>`
`<p></p>`
 - Links: `Click here`
 - Pictures: ``
 - Forms
- HTML 5 introduced many media elements

The Anatomy of a Webpage

- CSS (cascading style sheets)
- `<link href="/css/main.css?updated=20181020002547" rel="stylesheet" media="all">`
- view-source:<https://www.cs.uchicago.edu/css/main.css?updated=20181020002547>

The Anatomy of a Webpage

- DOM (document object model)



You type `uchicago.edu` into Firefox

- DNS (domain name service)
 - Resolves to IP address 128.135.164.125
- URL (uniform resource locator)
- `https://www.cs.uchicago.edu`
 - Protocol: `https`
 - Hostname: `www.cs.uchicago.edu`
 - Filename: `index.html` or similar (implicit)

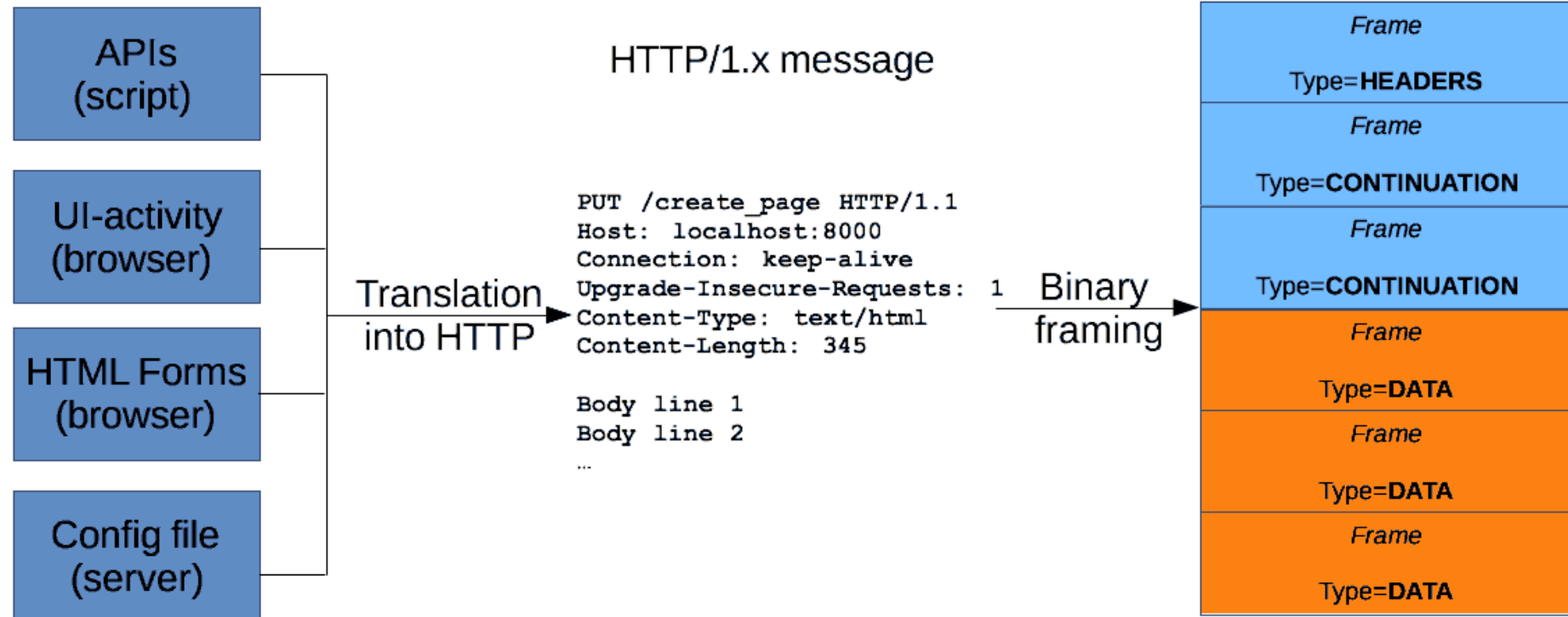
HTTP Request

- HTTP = Hypertext Transfer Protocol
- Start line: method, target, protocol version
 - GET /index.html HTTP/1.1
 - Method: GET, PUT, POST, HEAD, OPTIONS
- HTTP Headers
 - Host, User-agent, Referer, many others
 - <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>
- Body (not needed for GET, etc.)
- In Firefox: F12, “Network” to see HTTP requests

HTTP Request

- GET /index.html HTTP/1.1

Activity initiation



From <https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages>

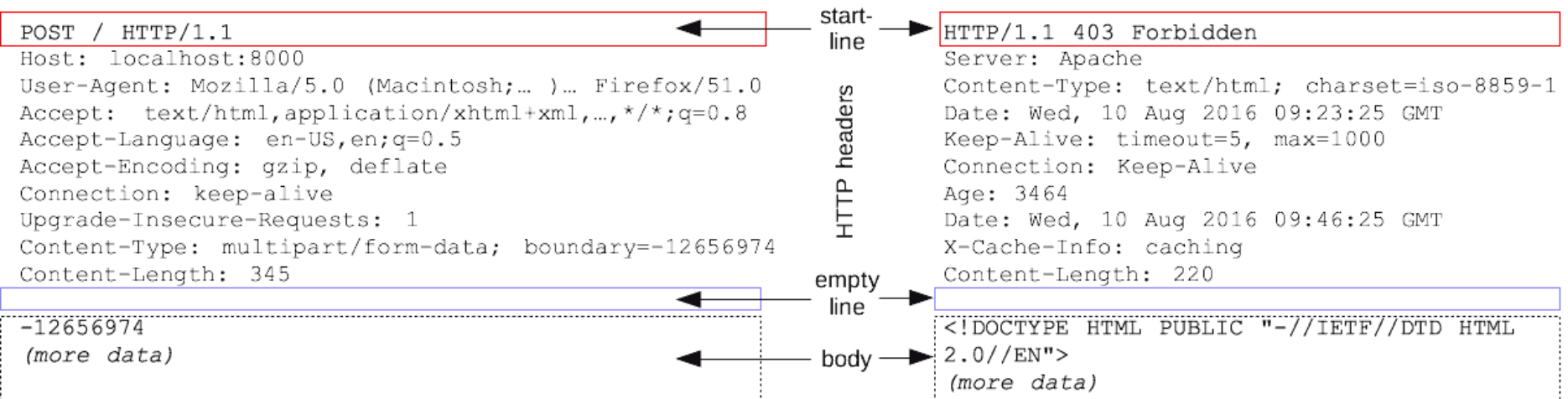
HTTP Response

- Status
 - 200 (OK)
 - 404 (not found)
 - 302 (redirect)
- HTTP Headers
- Body

HTTP

Requests

Responses



Sending Data to a Server

- GET request
 - Data at end of URL (following “?”)
- POST request
 - Typically used with forms
 - Data *not* in URL, but rather (in slightly encoded form) in the HTTP request body
- PUT request
 - Store an entity at a location

URL Parameters / Query String

- End of URL

- <https://www.cs.uchicago.edu/?test=foo&test2=bar>

The screenshot shows a web browser displaying the University of Chicago Department of Computer Science website. The address bar shows the URL <https://www.cs.uchicago.edu/?test=foo&test2=bar>. The website header includes the University of Chicago logo and navigation links: ABOUT, PEOPLE, RESEARCH, UNDERGRADUATE, GRADUATE, and ADMISSION. The main content area features the text "Department of Computer Science" and a background image of a light bulb. A network inspector is open at the bottom, showing a list of requests. The first request is a GET request to <https://www.cs.uchicago.edu/?test=foo&test2=bar> with a status of 200. The right pane of the network inspector shows the "Params" tab, displaying the query string parameters: `test: foo` and `test2: bar`.

Status	Method	F...	Domain	Cause	Type	Transferred	Size
200	GET	/?test=...	www.cs.uchi...	document	html	6.76 KB	23.87 KB
302	GET	fonts.css	cloud.typogr...	stylesheet	css	154.58 KB	205.03 KB
200	GET	main.cs...	www.cs.uchi...	stylesheet	css	cached	189.57 KB
200	GET	moder...	www.cs.uchi...	script	js	cached	5.65 KB
200	GET	jquery....	ajax.googlea...	script	js	cached	0 B
200	GET	jquery-...	ajax.googlea...	script	js	cached	0 B

Keeping State Using Cookies

- Cookies enable persistent state
- Set-Cookie HTTP header
- Cookie HTTP header
 - *Cookie: name=value; name2=value2; name3=value3*
- Cookies, once stored locally, are automatically sent with all requests your browser makes
- Session cookies vs. persistent cookies

Other Ways to Keep State

- Local storage
- Flash cookies
- (Many more)

HTTPS

- An extension of HTTP over TLS (i.e., the request/response itself is encrypted)
- Which CAs (certificate authorities) does your browser trust?
 - Firefox: Options → Privacy & Security → (all the way at the bottom) View Certificates
- How do you know if a cert is still valid
 - CRLs (certificate revocation lists)
 - OCSP (online certificate status protocol)

So... Interactive Pages?

- Javascript!
 - The core idea: Let's run (somewhat) arbitrary code on the **client's** computer
- Math, variables, control structures
- Imperative, object-oriented, or functional
- Modify the DOM
- Request data (e.g., through AJAX)
- Can be multi-threaded (web workers)

Common Javascript Libraries

- JQuery (easier access to DOM)
 - `$(".test").hide()` hides all elements with `class="test"`
- JQueryUI
- Bootstrap
- Angular / React
- Google Analytics (*sigh*)

What If You Make Poor Life Decisions?



Processing Data on the Server

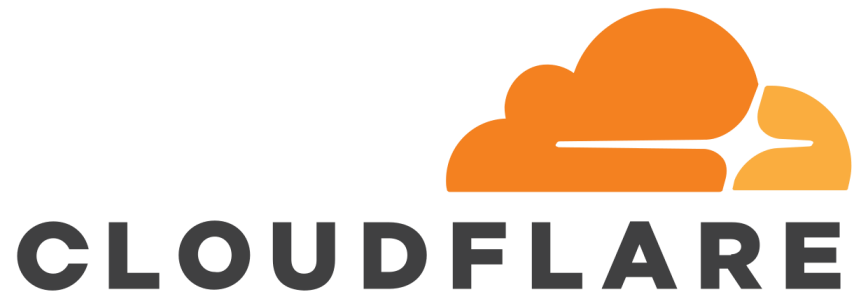
- Javascript is client-side
- Server-side you find Perl (CGI), PHP, Python (Django)
- Process data on the server
- What happens if this code crashes?

Storing Data on the Server

- Run a database on the server
- MySQL, SQLite, MongoDB, Redis, etc.
- You probably don't want to allow access from anything other than *localhost*
- You definitely don't want human-memorable passwords for these

What If You Get Lots of Traffic?

- CDNs (content delivery networks)



What If You Don't Want To Code?

- CMS (content management system)
 - WordPress (PHP + MySQL), Drupal

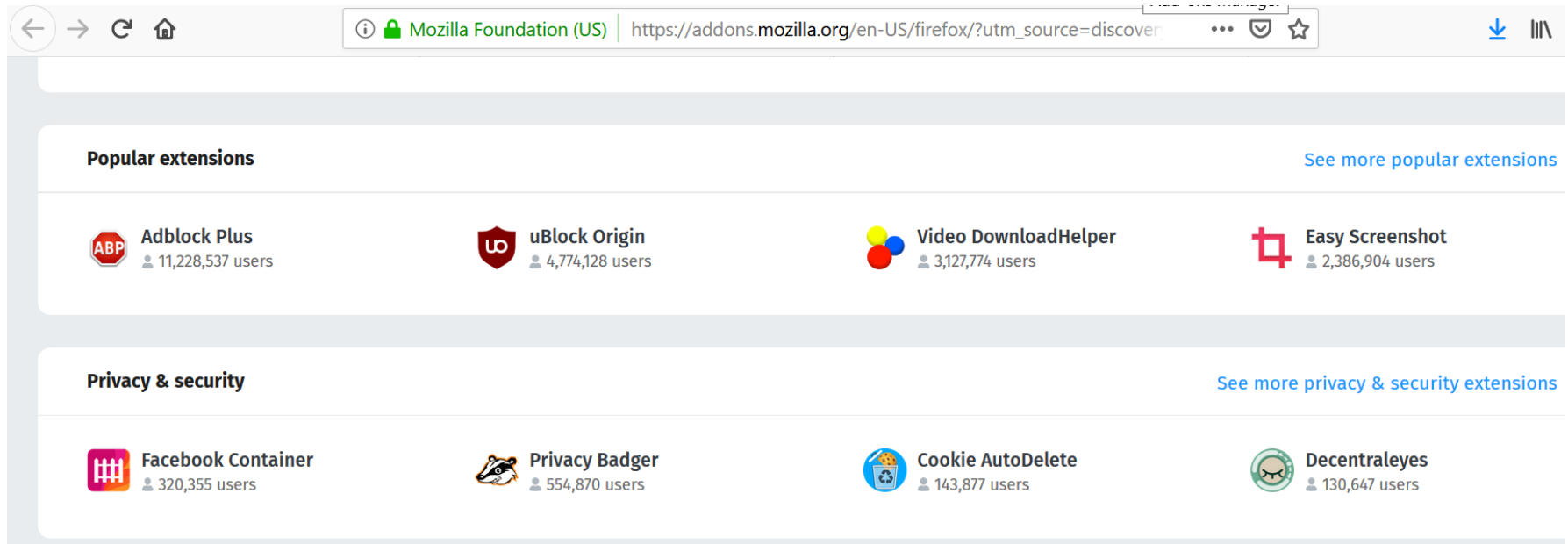
The screenshot displays the WordPress dashboard interface. At the top, the site name 'Restaurant World Tou...' is visible, along with navigation links for 'Upgrade to Pro', 'New Post', and a user profile 'Dave'. The left sidebar contains a menu with options like 'Dashboard', 'Home', 'Comments I've Made', 'Site Stats', 'Akismet Stats', 'My Blogs', 'Blogs I Follow', 'Store', 'Posts', 'Media', 'Links', 'Pages', 'Comments', 'Feedbacks', 'Appearance', 'Users', 'Tools', 'Settings', and 'Collapse menu'. The main content area is titled 'Dashboard' and features several widgets:

- Right Now:** A summary of site statistics.

CONTENT	DISCUSSION
8 Posts	9 Comments
1 Page	9 Approved
5 Categories	0 Pending
52 Tags	0 Spam
- QuickPress:** A form for quickly creating a new post, including fields for title, content, tags, and buttons for 'Add Media', 'Save Draft', 'Reset', and 'Publish'.
- Recent Comments:** A list of recent comments, showing a comment from 'Dave' on 'Arctic Char #'.
- Storage Space:** A widget showing storage usage: '3,072MB Space Allowed' and '0.08MB (0%) Space Used'.
- Recent Drafts:** A section indicating 'There are no drafts at the moment'.
- Stats:** A section stating 'No stats are available for this time period.'

Browser Extensions

- Can access most of what the browser can
- Requires permissions system
- Malicious extensions!



Same-Origin Policy

- Prevent malicious DOM access
- Origin = URI scheme, host name, port
- Only if origin that loaded script matches can a script access the DOM
 - Not where the script ultimately comes from, but what origin **loads** the script
- Frames / iframes impact origin
- CORS (Cross-Origin Resource Sharing)