

# INLS 672

## Web Development 2

### Web Performance



UNC  
SCHOOL OF INFORMATION  
AND LIBRARY SCIENCE

Joan Boone  
[jpboone@email.unc.edu](mailto:jpboone@email.unc.edu)

**Part 1: Overview**

Part 2: Improving and Measuring Performance

# Why Performance Matters for UX

- When asked in the study, users all claimed that speed was the most important factor impacting their mobile web usage.
- A Google study found that 53% of mobile site visits are abandoned if pages take longer than 3 seconds to load.



# How Page Speed Affects UX

Page speed impacts visitor satisfaction and conversion rates. A Google study found that when it comes to user experience, pages with a longer load time tend to have higher bounce rates and lower average time on page.



As page load time goes from:

**1s to 3s** the probability of bounce **increases 32%**



**1s to 5s** the probability of bounce **increases 90%**



**1s to 6s** the probability of bounce **increases 106%**



**1s to 10s** the probability of bounce **increases 123%**



# Why Performance Matters for Business

**WPO stats:** Case studies and experiments demonstrating the impact of web performance optimization (WPO) on user experience and business metrics

- **Yelp** reduced First Contentful Paint (75th percentile) by 45% and Yelp Page Complete (75th percentile) by 25% and saw a 15% improvement in their conversion rate.
- **Vodafone** improved their LCP by 31%, resulting in an 8% increase in sales, a 15% increase in their lead to visit rate, and an 11% increase in their cart to visit rate.
- **Rossignol.com** improved their load time by 1.9 seconds and cut their Speed Index by a factor of 10, contributing to a 94% improvement in conversion rate when compared to the year prior.
- A **Google study** over millions of page impressions found that when a site meets the recommended thresholds for the **Core Web Vitals** metrics, users are at least 24% less likely to abandon a page before it finishes loading.

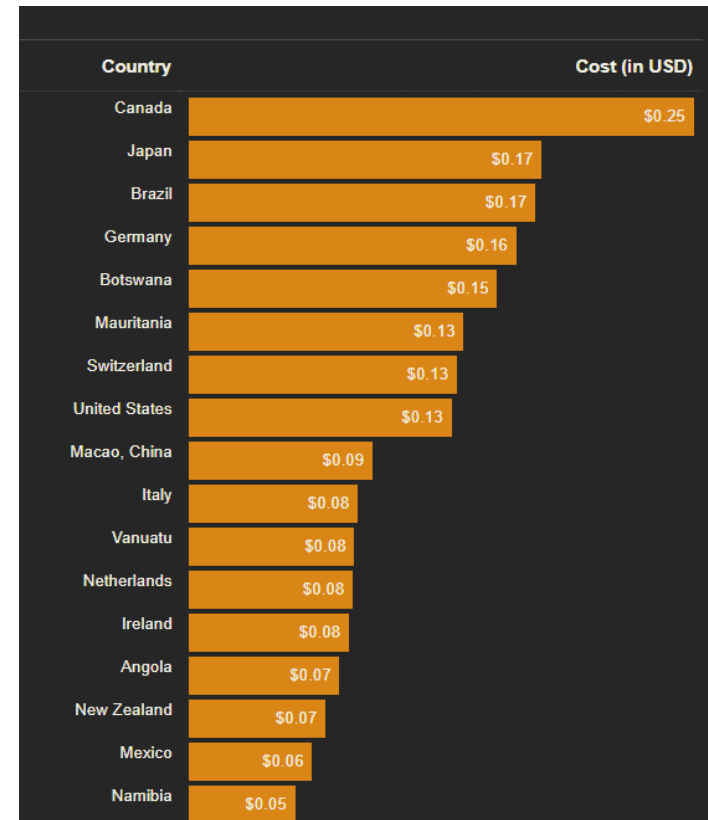
# Performance and Accessibility

MDN Web docs describes the [loading experience of CNN.com](#), which at the time of their evaluation, made over 400 HTTP requests with a file size over 22.6MB. Consider 3 scenarios:

- Using a desktop connected to a fiber optic network
- Using tethered mobile data on a 9-year-old iPad while commuting on public transportation
- Using a \$35 Huawei device in rural India with limited/no coverage

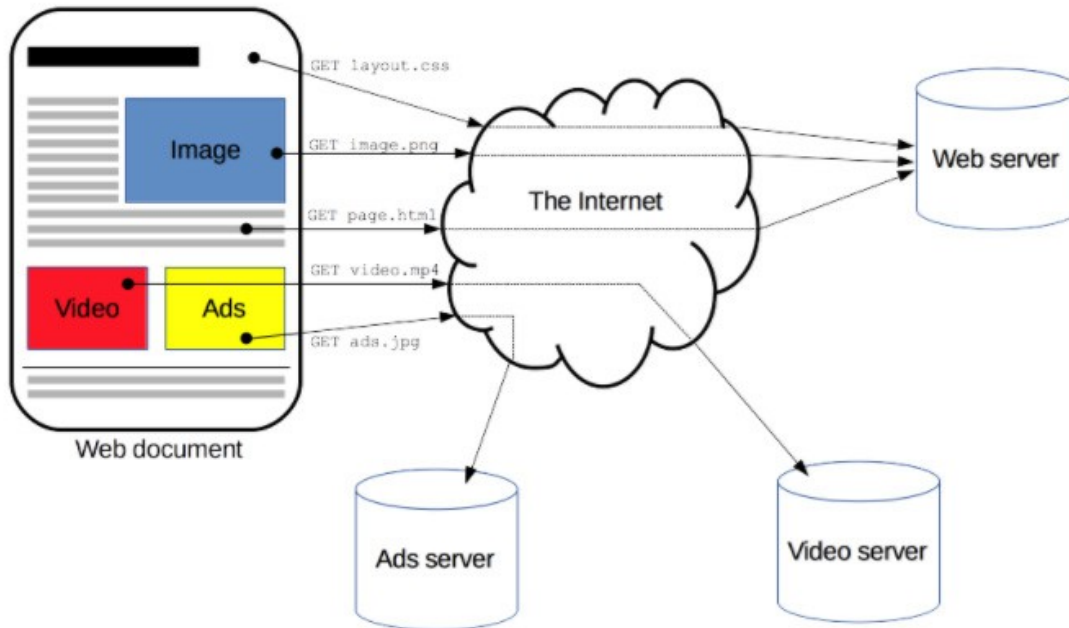
And it is not just the time, it is the cost to download where internet service is billed per megabyte.

According to the March 1, 2021 run of [HTTP Archive](#), the median site now weighs 2063kb. The approximate cost around the world is shown in the chart.



# How Browsers Work

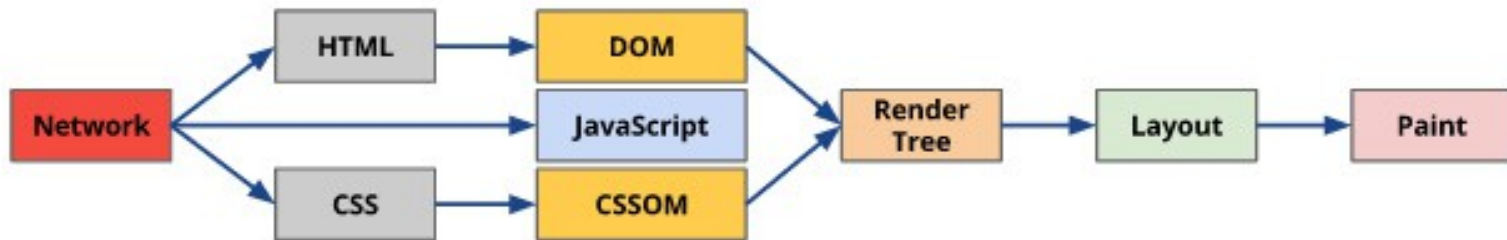
First, a DNS lookup is performed to locate the resources for a web page



Followed by a series of HTTP requests and responses to get the resources and populate the page

# How Browsers Render a Web Page

- After the browser has made the HTTP requests for the web resources it needs, it creates models of the page (DOM and CSSOM).
- A render tree is built, and the styles are computed for all visible content. Then layout occurs where the location and size of all render tree elements are defined. Once complete, the page is 'painted' on the screen
- Lastly, any JavaScript is given to the interpreter for execution.

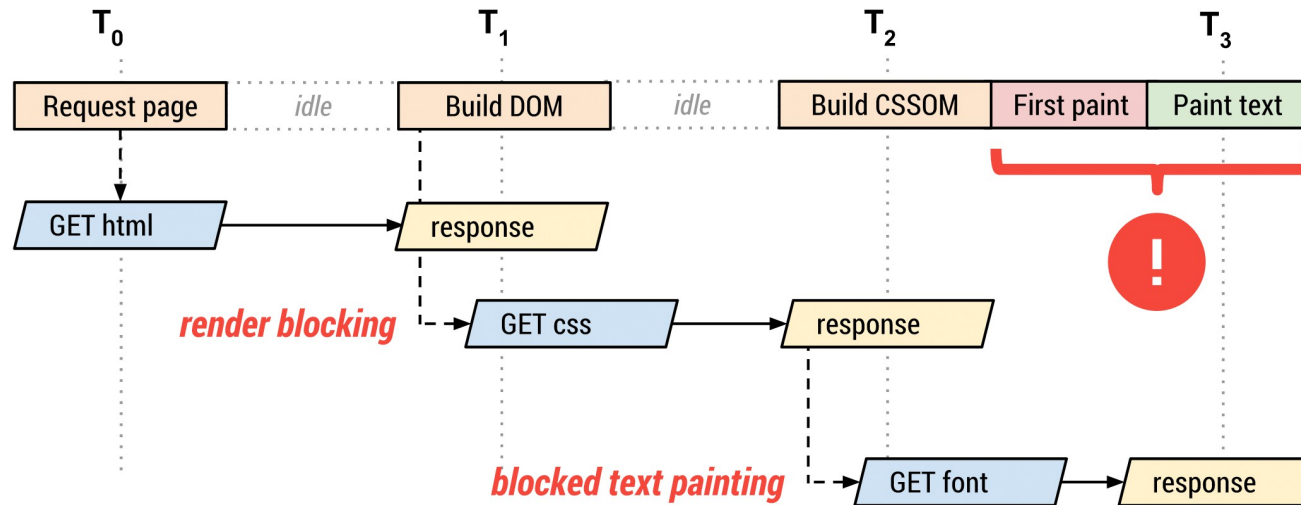


- This sequence of steps is the Critical Rendering Path that the browser goes through to convert your page to pixels on the screen.
- In addition to HTML, CSS, and JavaScript, your browser may also need other resources, such as images, videos, and fonts, to complete the rendering.
- The number of requests needed to obtain all of these resources, and their size, affect the page load time.



# Rendering Pages with Web Fonts

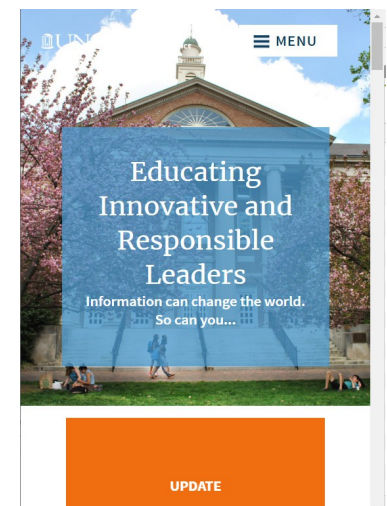
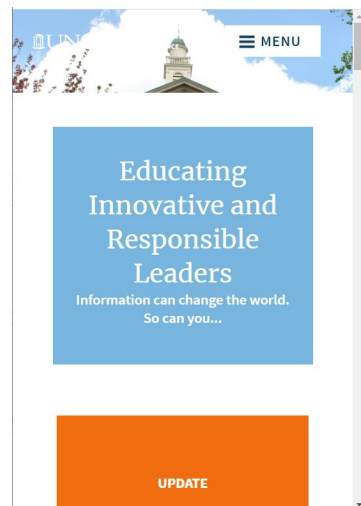
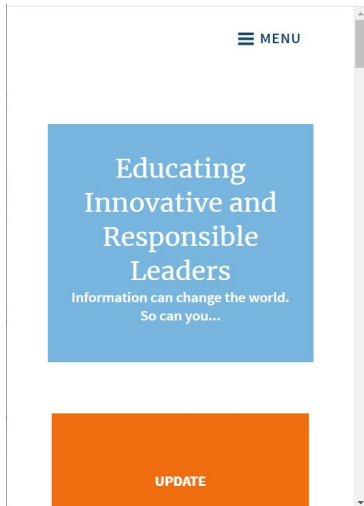
A potential disadvantage of web fonts is that font requests are delayed until the render tree is constructed, which can result in delayed text rendering



- Font requests are dispatched after the render tree is constructed
- The browser performs layout and paints content to the screen
  - If the font is not yet available, the browser may
    - Not display any text, resulting in a *flash of invisible text* (FOIT)
    - Display unstyled text, resulting in a *flash of unstyled text* (FOUT)
  - After the font is available, the browser paints the text pixels ([short demo](#))

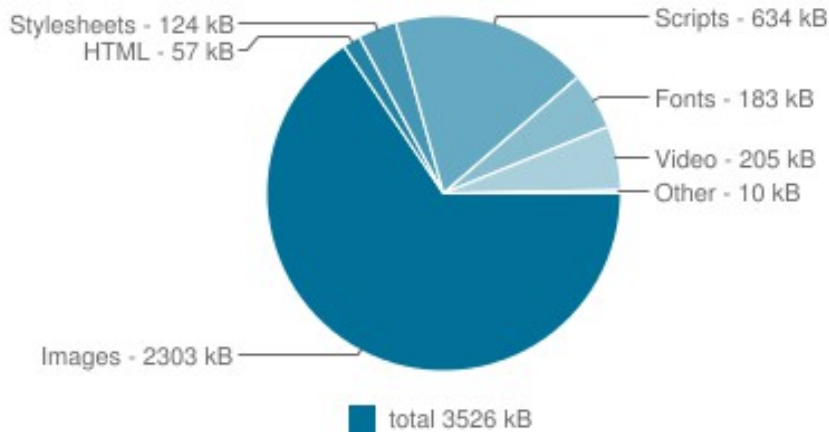
# Render a Web Page...in slow motion

- Often it is not visually apparent that there is a delay between requesting a web resource and displaying it. If you have a slow network connection, you may see the delay in rendering large images or fonts.
- One way to see this delay is to visit the [SILS home page](#) in Chrome
  - Right click on the page, and select 'Inspect'. On the right side in the DevTools pane, select the 'Network' tab at the top.
  - Then, reload the page again, and you should see the incremental rendering of the page as shown in the screenshots below. Try this with your website.



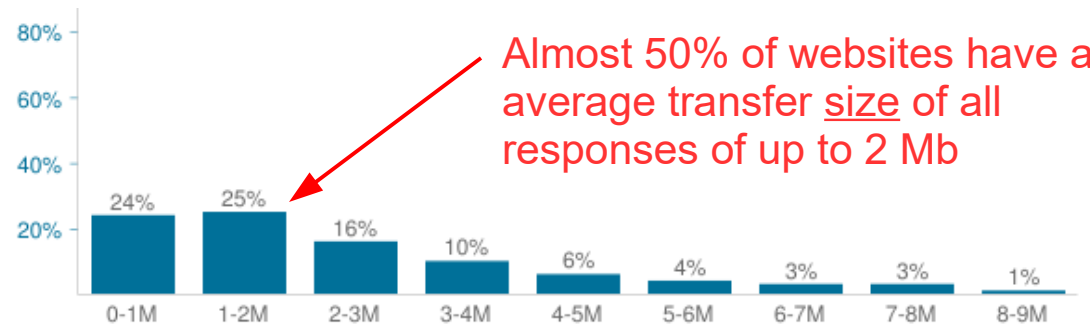
# What affects performance?

Average Bytes per Page by Content Type



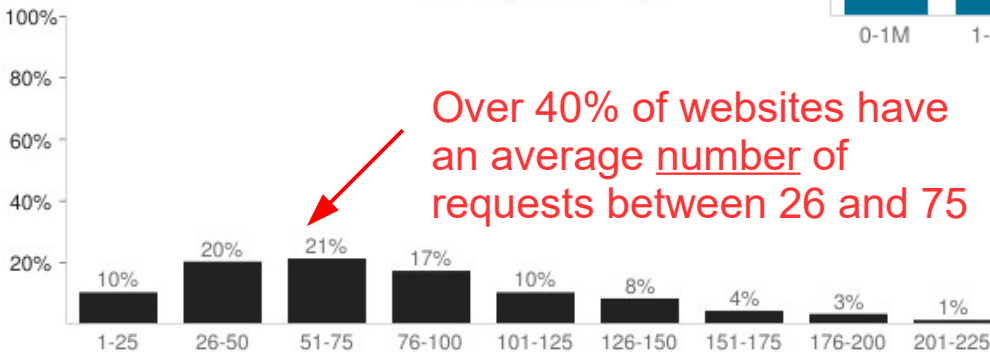
Many factors, but as a developer, you have control over the size and number of web resources

Total Transfer Size per Page



Almost 50% of websites have an average transfer size of all responses of up to 2 Mb

Total Requests per Page



Over 40% of websites have an average number of requests between 26 and 75

Part 1: Overview

**Part 2: Improving and Measuring Performance**

# A few ways to improve performance

## Reduce the number of requests

- Inventory your site assets: do you need every resource?
- Merge and/or inline multiple resources (CSS, JavaScript)
- Find an alternative to a web font, e.g., use a web safe font that is already installed on your computer

## Reduce the size of a requested resource

- Reduce text file size with **minification**, compression
- Image optimization
  - Replace images with CSS effects (shadows, gradients) or SVGs
  - Select the right image format (PNG, JPG, WebP)
  - Deliver scaled images
- Avoid bulky frameworks for simple tasks
  - Use small, lightweight frameworks for CSS and JavaScript
  - Use native/plain JavaScript

# Measuring Performance

How many requests?

How many bytes transferred?

How long did it take to load the page?

UNC SCHOOL OF INFORMATION AND LIBRARY SCIENCE

MENU

## Educating Innovative and

Elements Console Sources **Network** Performance Memory Application Security Audits

View: Group by frame Preserve log  Disable cache Offline Online

Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Status	Type	Initiator	Size	Time	Waterfall
ad_data_204	204	xhr	www-embed-player.js:384	115 B	443 ms	
cast_sender.js	200	script	remote.js:30	42.7 KB	20 ms	
generate_204?ICXvTQ	204	text/plain	VM449:1	0 B	153 ms	
sw.js	200	text/javascript	www.youtube.com/sw.js:-Infinity	0 B	246 ms	
favicon.ico	200	vnd.microsoft.icon	Other	0 B	27 ms	
log_event?alt=json&key=AlzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_11qcW8	200	xhr	base.js:2319	151 B	226 ms	

87 requests | 3.0 MB transferred | 4.5 MB resources | Finish: 27.72 s | DOMContentLoaded: 13.58 s | Load: 20.00 s

**Using Chrome Developer Tools**

- Inspect the page
- Select 'Network' tab
- Select 'Disable cache' checkbox
- Reload the page



# Google's Core Web Vitals

## Measuring the quality of user experience

The screenshot shows a browser window with the URL `nature.org/en-us/` and the Lighthouse performance report. The page content includes a header for 'The Nature Conservancy' and a main section titled 'Join Us in Speaking Up for Nature!' with a 'WATCH OUR EVENT' button. The Lighthouse report displays a performance score of 48. The metrics table is as follows:

Metric	Value
First Contentful Paint	3.2 s
Time to Interactive	20.5 s
Speed Index	10.6 s
Total Blocking Time	230 ms
Largest Contentful Paint	4.8 s
Cumulative Layout Shift	0.011

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

[View Original Trace](#)

**Opportunities** — These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

Opportunity Estimated Savings

Two ways....

- Enter the page URL at [PageSpeed Insights](#)
- Go to the page: Inspect it > Select Lighthouse tab > Select Performance Category > Generate report

# Performance Resources

## References

- web.dev: [Why does speed matter?](#)
- [Front End Performance Checklist 2021](#) by Vitaly Friedman
- MDN Web Docs: [What is web performance?](#)
- [Designing for Mobile Performance](#) by Mustafa Kurtuldu

## Evaluation Tools

- Chrome DevTools: Inspector > Audits > Performance
- Firefox: [Developer Tools](#), [Network Monitor](#), [Performance analysis](#)
- Google: [PageSpeed Insights](#)
- Google: [Core Web Vitals](#)
- [WebPageTest](#)