

Lecture

CS571 - Course Introduction

This content is protected and may not be shared, uploaded, or distributed.

What are We Talking About Today?

- **Course Structure and Administration**
 - Lecture schedules and resources (D2L Brightspace)
 - Assignments, exams, and final project details
 - Office hours and support (TAs, Piazza)
- **Learning Objectives and Technologies**
 - Skills: RESTful APIs, cloud services, mobile apps
 - Technologies: HTML, CSS, JavaScript, AJAX, Python, NoSQL
 - Frameworks: Bootstrap, Angular, React, AWS, Google Cloud, Azure
- **Prerequisites and Expectations**
 - Requires strong programming background
 - Advanced course with a comprehensive curriculum
 - Significant workload and time commitment required

General Information

- Lectures:
 - **Session 30378: Tue, Th 5:30PM – 7:20PM (SGM 124)**
 - **Session 30016: Tue, Th 5:30PM – 7:20PM (DEN@Viterbi)**
- Course website **retired**: <https://www.csci571.com> (still used for code samples)
- **D2L Brightspace** for all class materials, exams, grades, etc.:
<https://brightspace.usc.edu>
- Assignments – yes, attendance – up to you
- Two **Exams**, one at half course and one towards the end
- Exams auto-graded using **D2L Brightspace Quiz Tool**
- Cloud Deployment: **Google Cloud**

CS571: Web Technologies

- Instructor: Prof. Marco Papa
- Office hours:
 - **Thursday 1:00PM–5:00PM PDT**
 - **On Zoom**
- E-mail: `papa@usc.edu`
- Lectures, live on Zoom and recorded, stored on D2L Brightspace, exclusively for DEN-enrolled students
- **TA Office Hours:** daily (Mon–Fri) **on Zoom**
- 24/7 Q&A access: Piazza, **30 min. average response time**
- All office hours starting Week 2, on Jan. 19, 2026
- Instructor access: weekly
- TA access: daily
- Piazza access: 24/7
- Quick way to ask a “personal” question: Private post to Instructors on Piazza

General Rules

COURSE OVERLAP

Will allow, but you agree it is “your” problem

Fixed dates / times:

Exam #1: February 24, 5:30PM PDT

Exam #2: April 30, 5:30PM PDT

Final Project: May 7

Exams, “in-person”, auto-graded using D2L Brightspace Quiz Tool

Final Project on cloud and on Zoom video uploaded to D2L Brightspace

Benefits of Prof. Papa CSCI571 Staff

- Top 10 students in all sections combined offered available Course Producer or Grader positions at end of semester. **BY INVITATION ONLY**
- “Proof of skill” for H1B applications available at:
<https://www.cs.usc.edu/skills-verification/>
- Prof. Papa will act as “reference” in job / PhD applications for CSCI571 TAs, Course Producers

Learning Objectives

- By the end of the semester, you will be able to:
 - Write RESTful API applications
 - Set up Cloud services
 - Design and code back-end scripts in Python and JavaScript
 - Design and code front-end Web Applications
 - Design and code Mobile Apps in Java/Kotlin
 - Design and code web front-end asynchronous applications using AJAX
 - Design and code responsive web apps
 - Write front-end programs using JavaScript
 - Design graphical user interfaces using HTML and CSS
 - Read Web Services API documentation and use it in building Web applications
 - Write Microservices and Containers
 - Use a NoSQL database like MongoDB Atlas
 - **Use AI tools for "vibe coding"**

Prerequisites

- The course is recommended for students with deep knowledge of at least two programming languages (JavaScript, Java, Python, C#, C++)
- CSCI 571 is **not a "beginner course"** for the Web; **it is an "advanced course"**
- If you are interested in beginner courses, check out these ITP courses:
 - TAC 104, "Introduction to Web Development".
 - TAC 301, "Front-End Web Development"
 - TAC 303, "Full-Stack Web Development"
 - TAC 304, "Back-End Web development"
 - TAC 341, "Android App Development"
 - TAC 342, "iOS App Development"
- In CSCI 571 you have all of the above in 1 course!
- Understand that the **course load will be massive**
- You need to have the time available to be successful
- Consider yourself **warned**

Course Technologies

- This course focuses on the phenomenon known as the World Wide Web
- Core technologies
 - HyperText Markup Language (HTML) and Cascading Style Sheets (CSS)
 - HyperText Transfer Protocol (HTTP)
 - Web servers, their configuration and performance properties
 - Server-Side programming using JavaScript and Python
 - Client-side programming using JavaScript and JS Frameworks
 - Ajax Development Style
- Newer Technologies of Interest
 - Responsive Website Design (Bootstrap, etc.)
 - JS Frameworks (Angular, React and Node.js)
 - Web Services (REST), including NoSQL databases (MongoDB)
 - Web security, TOR, Dark web
 - Native Mobile frameworks (Java / Android and Swift / iOS)
 - React native
 - Cloud computing
 - Serverless Applications, Containers, Docker
 - AWS Lambda, Google Cloud Functions, Azure Functions

Software and Storage

- **Website / Web Services in the cloud**
 - **All assignments deployed on Google Cloud (GC)**
 - Serverless: Google Cloud Run functions, Google Cloud Run
 - Python / Flask
 - Node.js / Express

Other Issues

- Piazza class news group
 - Activate your membership by self-joining at:
`piazza.com/usc/spring2026/cs571`
 - Class Access Code: **`lafc3252usc`**
 - **Discussions limited to course materials, assignments and exams**
- Academic Integrity Policy
 - Do NOT submit the same program; you can discuss the project with fellow students, but do not develop code with other students; do not download code online; do not post code online; we use MOSS to check for plagiarism.
 - AI Tool generated code is expected to be "similar."
- **D2L Brightspace** main site for course materials
 - **`https://brightspace.usc.edu`**
 - Course slides (PDF)
 - Exams (Quiz Tool)
 - Assignments (Folders)
 - Grades

BRIGHTSPACE

Student Evaluations

- Comments:
 - "Amazing assignments. Learnt a lot on the course."
 - "Projects seemed similar to an actual client for web development would ask for."
 - "Even though the assignments were hard I learnt a lot from them."
 - "It is not a fair game for beginners. I've spent almost 3 weeks to do a homework, and I still can not finish it on time."
 - "Course projects are impressive!"
 - "This class has posted assignments easily x10 times larger than other classes."
 - "last two assignments take *forever*."
 - "The homework assignments are so difficult."
 - "I had to do so much googling on my own to learn about concepts used in the homework assignments."
 - "Tough class with a lot of valuable assignments."
 - "Massive assignments."
 - "The workload of this course is too much, especially the last two homework."

Academic Integrity Violations

- Spring 2018 violations (16):
 - Sanctioned: 16, Appeals Panel: 8, Appeals to Dean: 2: no changes
 - **F in course: 12**, 0 + full letter grade reduction: 4 (C-, C-, C-, B-)
- Fall 2018 violations (0):
 - None!
- Spring 2019 violations (4):
 - **F in course: 1**, 0 + full letter grade reduction: 1 (B-), C in the course: 2
- Fall 2019 violations (6):
 - **F in course: 1**, 0 + full letter grade reduction: 5 (B-, C)
- Spring 2020 violations (4):
 - 0 + full letter grade reduction: 1 (A-, C)
- Spring 2021 violations (2)
 - Full letter grade reduction
- Spring 2022 violations (52)
 - **F in course: 2**, 0 in assignments, C, and D in course
- Fall 2022 violations (5)
 - 0 in assignments, **F in course: 1**, point deductions
- Spring 2023 violations (4)
 - 0 in assignments
- Fall 2023 violations (4)
 - 0 in assignments
- Spring 2024 violations (10)
 - 0 in assignments
- Fall 2024 violations (21)
 - 0 in assignments

Use of AI in course

- Course policy on AI
 - Using an AI tool such as ChatGPT, Gemini, Claude, Grok or similar AI chatbot is **encouraged** for all assignments.
- IDEs + AI tool plug-ins + LLMs to generate code will be recommended.
- “Vibe coding” is expected.
- **Process Log Policy:** all assignments require a process log (10% of score). AI users: submit prompts and fixes. Non-AI users: describe your coding process.
- Using AI-generated tools to produce flash cards or cheat sheets from the class slides PDFs is also prohibited, as a violation of copyright (i.e., modification of existing material).
- MOSS will be used to discover plagiarism.

MOSS Plagiarism discovery tool

The screenshot displays the MOSS (Measure of Software Similarity) interface in a web browser. The browser's address bar shows the URL: `moss.stanford.edu/results/1/5195737631033/match16.html`. The page title is "Matches for 7521422733PurvilRake: X".

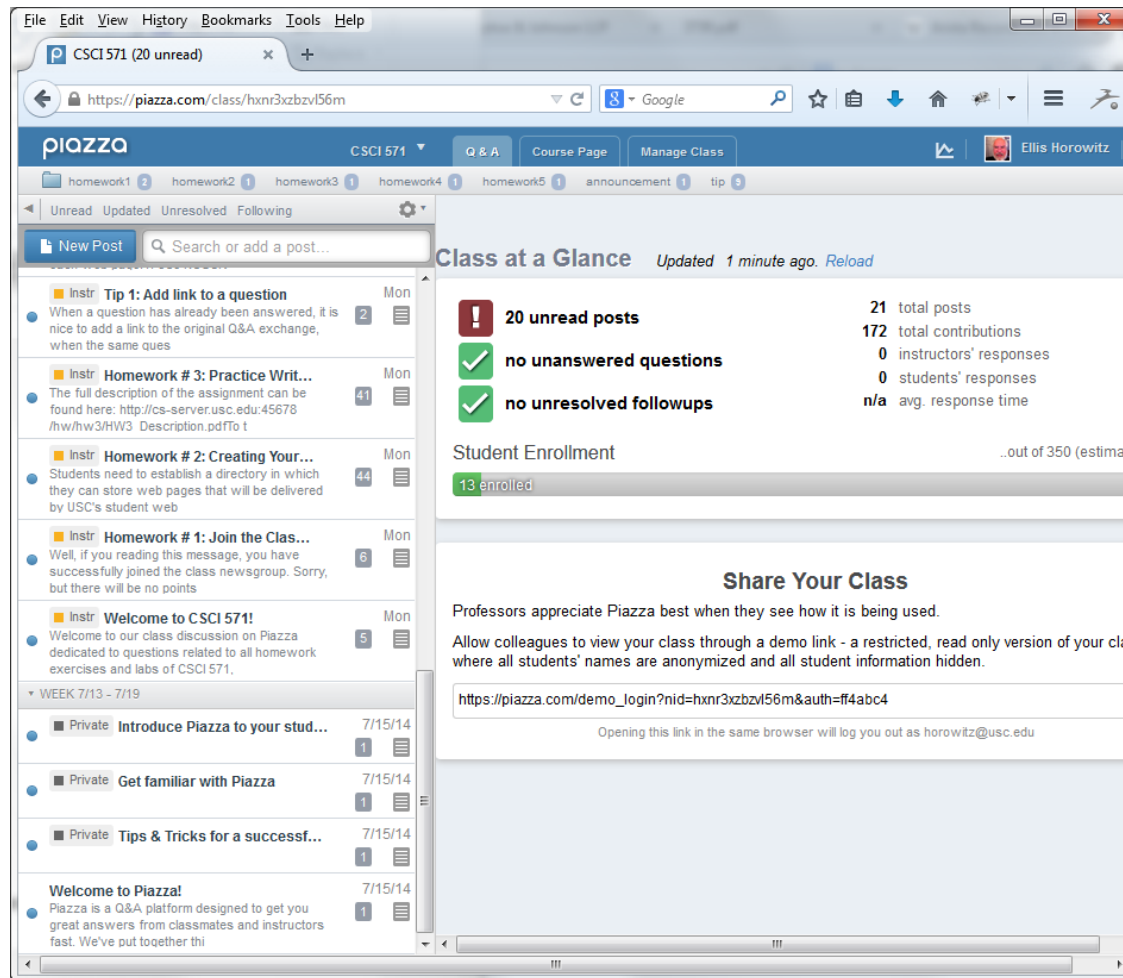
At the top, there is a table of matches. The first match is highlighted in red, indicating a high similarity score. The table has columns for the document ID, the document name, the similarity percentage, and the range of lines that match.

Document ID	Document Name	Similarity (%)	Match Range
7521422733PurvilRake: X	Feb2420221027PM/ (82 %)	82 %	224-294
7520422733PurvilRake: X	Feb2420221023PM/ (83 %)	83 %	181-250



Below the table, the source code of the two documents is displayed side-by-side for comparison. The code is a JavaScript configuration for a Highcharts chart, specifically for a stock price area chart. The code is identical in both documents, showing a high degree of similarity.

```
plotOptions: {
  series: {
    pointWidth: 3,
    pointPlacement: 'on'
  }
},
rangeSelector: {
  selected: 0,
  inputEnabled: false,
  allButtonsEnabled: true,
  buttons: [
    {
      type: 'day',
      count: 7,
      text: '7d'
    },
    {
      type: 'day',
      count: 15,
      text: '15d'
    },
    {
      type: 'month',
      count: 1,
      text: '1m'
    },
    {
      type: 'month',
      count: 3,
      text: '3m'
    },
    {
      type: 'month',
      count: 6,
      text: '6m'
    }
  ]
},
series: [
  {
    name: 'Stock Price',
    type: 'area',
    data: stockPriceData,
    tooltip: {
      valueDecimals: 2
    },
    fillColor: {
      linearGradient: {
        x1: 0,
        y1: 0,
        x2: 0,
        y2: 1
      },
      stops: [
        [0, Highcharts.getOptions().colors[0]],
        [1, Highcharts.color(Highcharts.getOptions().colors[0]).setOpacity(0.5)]
      ]
    },
    threshold: null
  }
]
```

Piazza



Who am I?

- PhD in CS from USC, class of '88
- Initial career: MS Windows, Commodore Amiga developer
- System Architect → Team Lead → IT Project Manager → VP Engineering → CTO
- CTO (Chief Technology Officer) at Luckman Interactive and CareerBuilder
- **Part-time Faculty at USC 2003-2022**
- **Full-time Faculty at USC starting on August 2022**
- Chief Technologist at LASC (Los Angeles Superior Court) 2002-2022
- Microsoft 365 SharePoint Project Manager 2014-2022
- Active member of LA CTO Forum (Silicon Beach CTOs)
- Season ticket holder of USC Football and  LAFC (Los Angeles Football Club) and  supporter

Characterizing Web Content

There are very few studies that examine the types of content on the web, however . . .

(From IEEE Spectrum, Jan. 2004, pp. 75):

- Claim: 30% of the web is porn
- Claim: 30% of the web is duplicate information
- 50,000,000 pages are either new or changed *each day*
- 65% of the web pages are in English

(From Personal Computer World, Optenet, Sep. 2008):

- Claim: 35% of the web is porn, 11% is e-commerce
- <http://www.optenet.com/en-us/new.asp?id=162> (N/A)

(From Forbes, Sept. 2011):

- Claim: 4% is porn, 13% are porn Web Searches
- <http://www.forbes.com/sites/julieruvolo/2011/09/07/how-much-of-the-internet-is-actually-for-porn/>

(From BBC, July 2013):

- Claim: is porn 4% or 37%?
- <http://www.bbc.com/news/technology-23030090/>

Sample Web Site (Small size)

- Here is a popular strategy for maintaining a web site
- The facts:
 - **www.autobytel.com**, new and used cars (now **AutoWeb**)
 - Market Cap: \$33.92M (Dec. 2019)
 - Quarterly Revenues: \$28.6M (3rd Quarter 2019)
 - Lead traffic: 31.7M visits (3rd Quarter 2019)
 - Mobile version launched in 2012
 - Stock symbol: **AUTO** (Nasdaq) - delisted
- Original Microsoft solution:
 - Microsoft Windows Server
 - Microsoft IIS 7.5 web server
 - Microsoft SQL server database
 - Akamai CDN
- Today:
 - Microsoft Azure hosting
 - F5 BIG-IP OS, BigIP web server

Sample Web Site (Medium Size)

- Running a web site can get complicated; here is one example.
- The facts:
 - **`https://www.on-running.com`**, online sale of shoes and clothing goods
 - Public company (ONON on NYSE)
 - Market Cap: \$8.5B
 - Revenues: \$1.2B (2022)
 - 10.3M page visits per month
 - 5.69 average pages per visit
- The solution:
 - **Cloudflare** CDN hosting, DNS, SSL/TLS certs
 - Linux OS
 - cloudflare-nginx web server
 - Gzip compression, HSTS, HTML5

Sample Web Site (large size)

- The facts:
 - **www.etrade.com**, online investing services and resources
 - Market Cap: \$10.37B (Dec. 2019)
 - Yearly Revenues: \$2.9B (12/2018)
 - 60 million-page views per month
 - average of 53,000 unique visitors per day
 - 4.9 million accounts (Jan. 2015)
 - 25,000 new retail accounts opened (Oct 2015)
 - 1,952,000 customer transactions per month
 - Taken over by Morgan Stanley in 2020 for \$13 billion
- The solution:
 - **AWS** hosting
 - IBM 90 xSeries running Linux/**Citrix Netscaler**, **Apache** and **Tomcat** web servers, **AWS** Route 53 (DNS)
 - Hardware facility for load balancing and redundancy
 - Oracle database system
 - Proprietary programming systems

Web Server Farms

- Until recently all serious web sites were maintained using web server farms;
 - A group of computers acting as servers and housed in a single location;
 - Internet Service Providers (ISP's) provide web hosting services using a web server farm
- Hardware and software is used to load balance requests across the machines
- Other issues addressed by web server farms include:
 - **Redundancy** eliminates single point of failure; backup and failover strategy is required
 - **Security**, secure areas are placed behind firewalls which monitor web traffic, network address translation, port translation, SSL





















Popular Web Hosting Services

- ***For individuals and small business:***
 - **1&1**
<https://www.ionos.com>
 - **GoDaddy.com**
<https://www.godaddy.com/hosting/web-hosting>
 - **Yahoo**
<http://www.iwebhostingplans.com/yahoo/yahoowebhosting.asp>
- ***For companies willing to pay MUCH higher costs:***
 - **Rackspace**
https://www.rackspace.com/?CMP=Google_hosting
 - **Network Solutions**
<http://www.networksolutions.com/hosting/>
- ***Reviews and price comparisons:***
 - <https://www.hosting-review.com/top-ten-list/>
See next slide "monthly" prices
Top: <https://www.hostpapa.com/buy/shared-hosting/>
 - <http://www.pcmag.com/categories/web-hosting>

Web Hosting Services

Updated June 2025

Top 10 Web Hosting Providers in the US

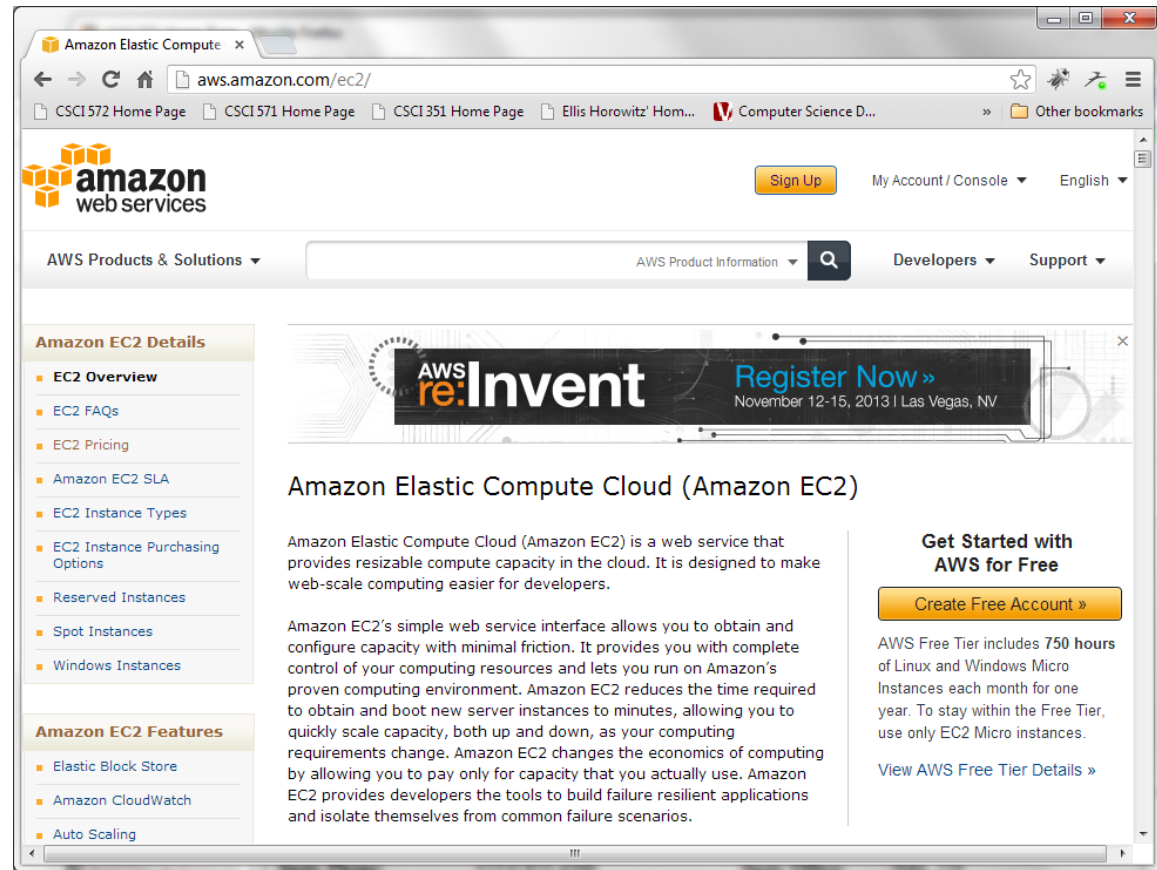
RANK	WEB HOST	PRICE	SPEED & UPTIME	CUSTOMER REVIEWS	TOTAL SCORE & REVIEW	VISIT
1	 HostPapa Show More ▾	\$1.95/mo		★★★★★ 4.9/5	🏆 98% Hostpapa Review	Visit Now →
2	 bluehost Show More ▾	\$1.99/mo		★★★★★ 4.7/5	88% Bluehost Review	Visit Now →
3	 HOSTARMADA Show More ▾	\$2.49/mo		★★★★☆ 4/5	80% HostArmada Review	Visit Now →
4	 A2 HOSTING Show More ▾	\$2.99/mo		★★★★☆ 3.5/5	78% A2 Hosting Review	Visit Now →
5	 Verpex Show More ▾	\$2.70/mo		★★★★☆ 3.5/5	76% Verpex Review	Visit Now →
6	 SiteGround Show More ▾	\$1.99/mo		★★★★☆ 3.3/5	72% SiteGround Review	Visit Now →
7	 IONOS Show More ▾	\$1/mo		★★★★☆ 3.3/5	72% IONOS Review	Visit Now →
8	 inmotion. hosting Show More ▾	\$2.29/mo		★★★★☆ 3.0/5	70% InMotion Hosting Review	Visit Now →
9	 DreamHost Show More ▾	\$2.95/mo		★★★★☆ 3.0/5	70% DreamHost Review	Visit Now →
10	 GoDaddy Show More ▾	\$6.99/mo		★★★★☆ 3.0/5	70% GoDaddy Review	Visit Now →

Cloud Computing

- **Cloud computing** is Internet-based computing, whereby shared resources, software, and information are provided to computers and other devices **on demand**, like the electricity grid.
- Users no longer have need for expertise in, or control over, the technology infrastructure "in the cloud" that supports them.
- It typically includes web-based tools or applications that users can access and use through a web browser as if it were a program installed locally on their own computer. ¹
- Typical cloud computing providers deliver common business applications online that are accessed from another Web service or software like a Web browser, while the software and data are stored on servers.
- The major cloud service providers include Amazon, Google, Microsoft, Salesforce, Skytap, HP, IBM, Amazon, Google and Apple (iCloud).

An Example – Amazon’s Elastic Compute Cloud

- A web service providing resizable compute capacity
- The “elastic” nature means the service instantly scales to meet demand with no up-front investment
- Users create an Amazon Machine Image (AMI), a virtual computer running your selected operating system (Linux, Windows, etc)
- Users use Amazon’s Simple Storage Service (S3) for large-scale, persistent storage
- You only pay for running AMI
- All accounts are limited to 5 Elastic IPv4 addresses per region
- See: aws.amazon.com/ec2

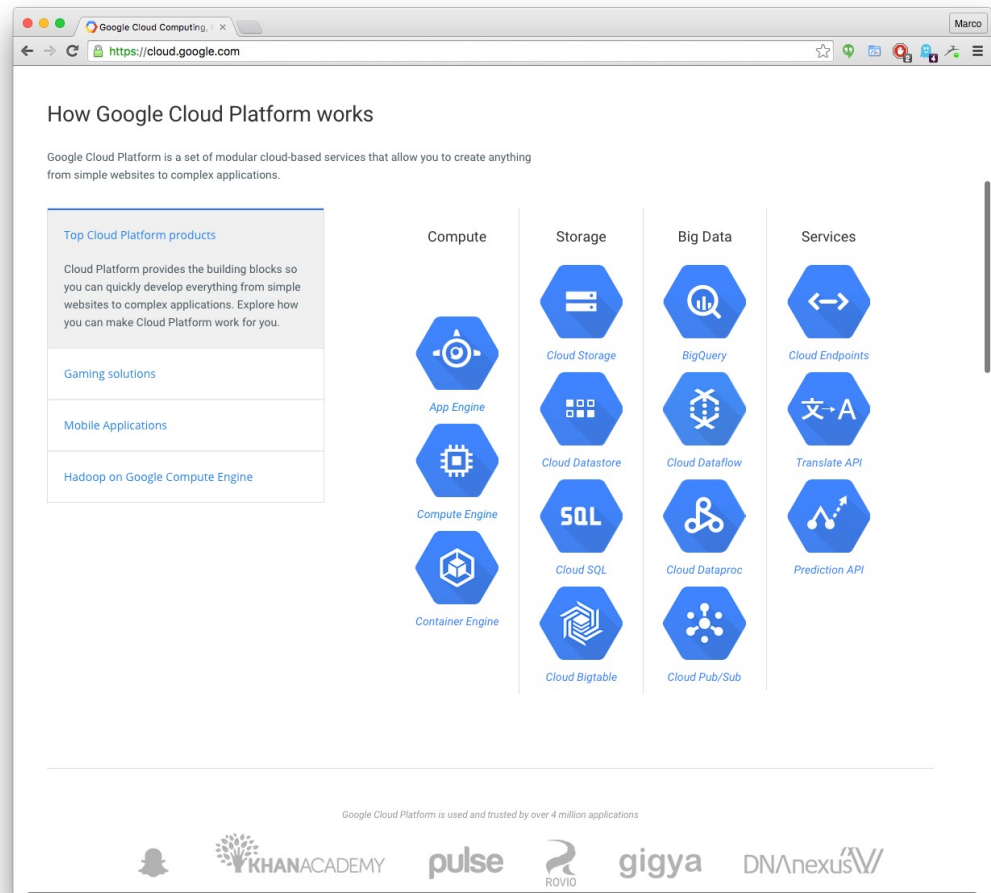


Amazon currently runs in 8 regions: US East, US West (Oregon), US West (Northern CA), Ireland, Asia Pacific (Singapore), Asia Pacific (Tokyo), Asia Pacific (Sydney), South America (Sao Paulo)

An Example – Google Cloud

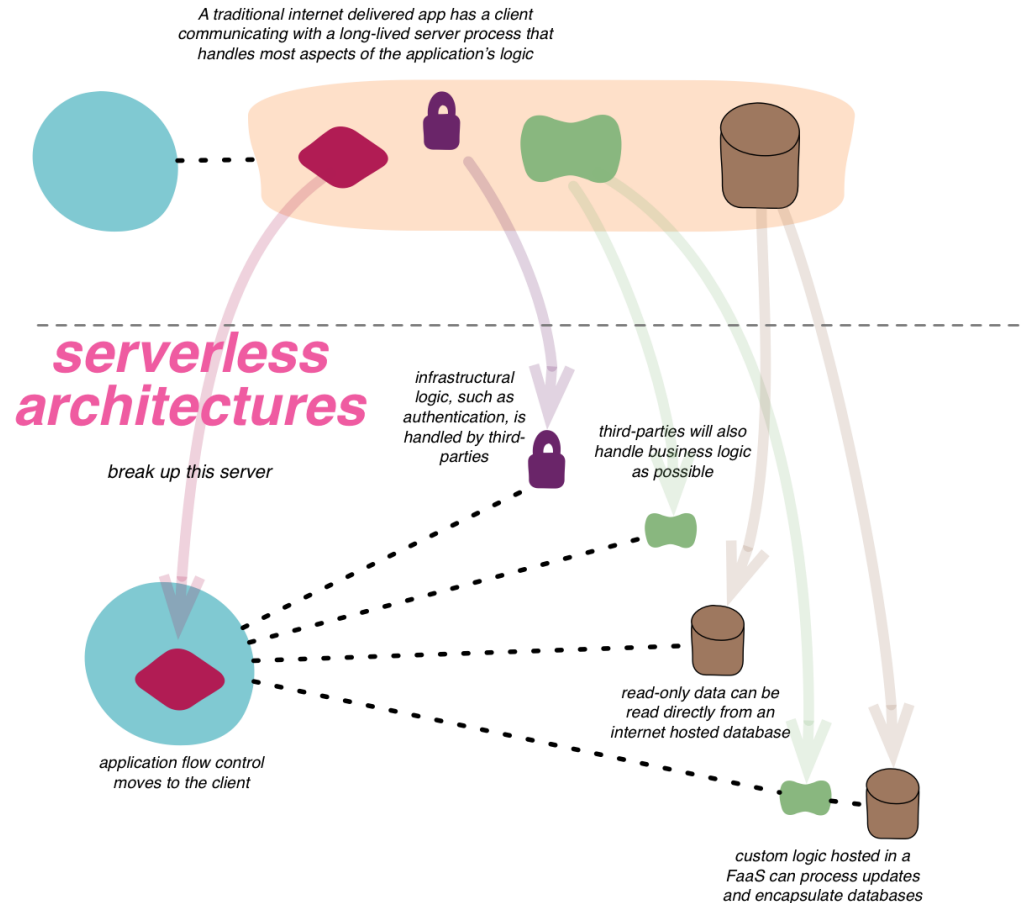
- A web service providing basic Compute, Storage, Big Data and Services.
- Additional services for massively scalable Gaming solutions, Mobile Applications backend, and Apache Hadoop.
- App Engine – A platform for building scalable web applications and mobile backends. App Engine scales applications automatically in response to the amount of traffic it receives.
- Compute Engine - Offers predefined virtual machine configurations: Debian, CentOS, CoreOS, SUSE, Ubuntu, Red Hat, FreeBSD, or Windows 2008/2012.

Google uses software-defined networking technology to route packets across the globe and enable fast edge-caching so that data is where it needs to be to serve users.



Serverless Architecture

- Internet based systems where the application development does not use the usual server process.
- They rely solely on a combination of:
 - third-party services, or Backend as a Service (BaaS)
 - client-side logic
 - service hosted remote procedure calls, or Function as a Service (FaaS).
- AWS Lambda is one of the most popular implementations of FaaS at present, but there are others. See: <https://aws.amazon.com/lambda/>
- Serverless and containers will be covered later in the course



USC Has Many Web Servers Running

[LEARN MORE](#)[REPORT FRAUD ↗](#)

Hostnames matching usc.edu

► 🔍 Search with another pattern?

120 results (showing 1 to 20)

Rank	Site	First seen	Netblock	OS	Site Report
72414	login.usc.edu ↗	March 2020	Amazon Technologies Inc.	Linux	
79277	bytes.usc.edu ↗	September 2017	University of Southern California	Linux - Ubuntu	
93270	shibboleth.usc.edu ↗	February 2007	University of Southern California	unknown	
104928	mychart.musc.edu ↗	June 2012	Medical University of South Carolina	Citrix NetScaler	
143668	libguides.usc.edu ↗	August 2018	Amazon.com, Inc.	Linux	
150569	viterbigradadmission.usc.edu ↗	March 2017	WPEngine, Inc.	Linux	
152228	gradadm.usc.edu ↗	October 2017	Google LLC	Linux	
155572	www.usc.edu ↗	August 1995	Amazon.com, Inc.	Linux	
160240	camel2.usc.edu ↗	January 2004	University of Southern California	unknown	
193115	my.usc.edu ↗	April 2004	University of Southern California	unknown	
196005	catalogue.usc.edu ↗	April 2013	Amazon Technologies Inc.	Linux	
207538	www.scf.usc.edu ↗	August 1995	University of Southern California	unknown	
219979	dornsife.usc.edu ↗	May 2011	University of Southern California	unknown	
242432	itservices.usc.edu ↗	April 2014	University of Southern California	unknown	

- Netcraft lists **137** separate sites / web servers with usc.edu in their name, e.g.
- `www.usc.edu`
- `mat.usc.edu`
- `www.cs.usc.edu`
- `dornsife.usc.edu`
- `web-applusc.edu`
- `www-scf.usc.edu`
- However, some may not be connected to USC, e.g.
- `www.usc.edu.au`
- Check at:
<https://searchdns.netcraft.com>

Web Browsers Use Standard Layout Engines

- **WebKit** is a software component used to render web pages; it is open source.
 - It is used by Apple's Safari web browsers, and previously by Google's Chrome
 - WebKit is also the name of the macOS system framework version of the engine that's used by Safari, Dashboard, Mail, and many other macOS applications
- **Gecko** is a layout engine developed by Mozilla Corporation, known as the layout engine of the Firefox web browser.
 - It is used to display web pages and, in some cases, an application's user interface.
 - It offers a rich programming API that makes it suitable for a wide variety of roles in Internet-enabled applications, such as web browsers
 - Its development originated with Netscape
- **Chromium** is a web browser project, developed by Google.
 - Engine originally based on Blink, a fork of Apple WebKit
 - Used in Chrome, Microsoft Edge, Opera, Samsung Internet and many others
- Some web kits and the browsers that use them:
 - **Gecko-based:** FireFox (Mozilla), Flock, Netscape
 - **Trident-shells:** Internet Explorer (Microsoft, retired), Edge (retired in Aug. 2021)
 - **WebKit-based:** Midori, Safari and Mobile Safari (Apple), Symbian^3 (Nokia) and many others
 - **Chromium-based:** Chrome, Microsoft Edge (2021+)

Capabilities of a Browser

- Web browsers fetch and display documents from other WWW sites; their capabilities include:
 - A mouse-driven graphical user interface
 - Display of
 - Hypertext documents conforming to latest HTML standard
 - Text with fonts, styles, and varying point sizes
 - Foreign-language character sets conforming to ISO-8859
 - Forms composed of edit boxes, check boxes, radio boxes, lists, text areas, etc.
 - Graphics in different formats (GIF, JPEG, MPEG, PNG, XBM) including monochrome, color

GIF = graphic interchange format, MPEG = Motion Picture Experts Group, JPEG = Joint Photographic Experts Group, PNG = Portable Network Graphics, XBM = x bitmap

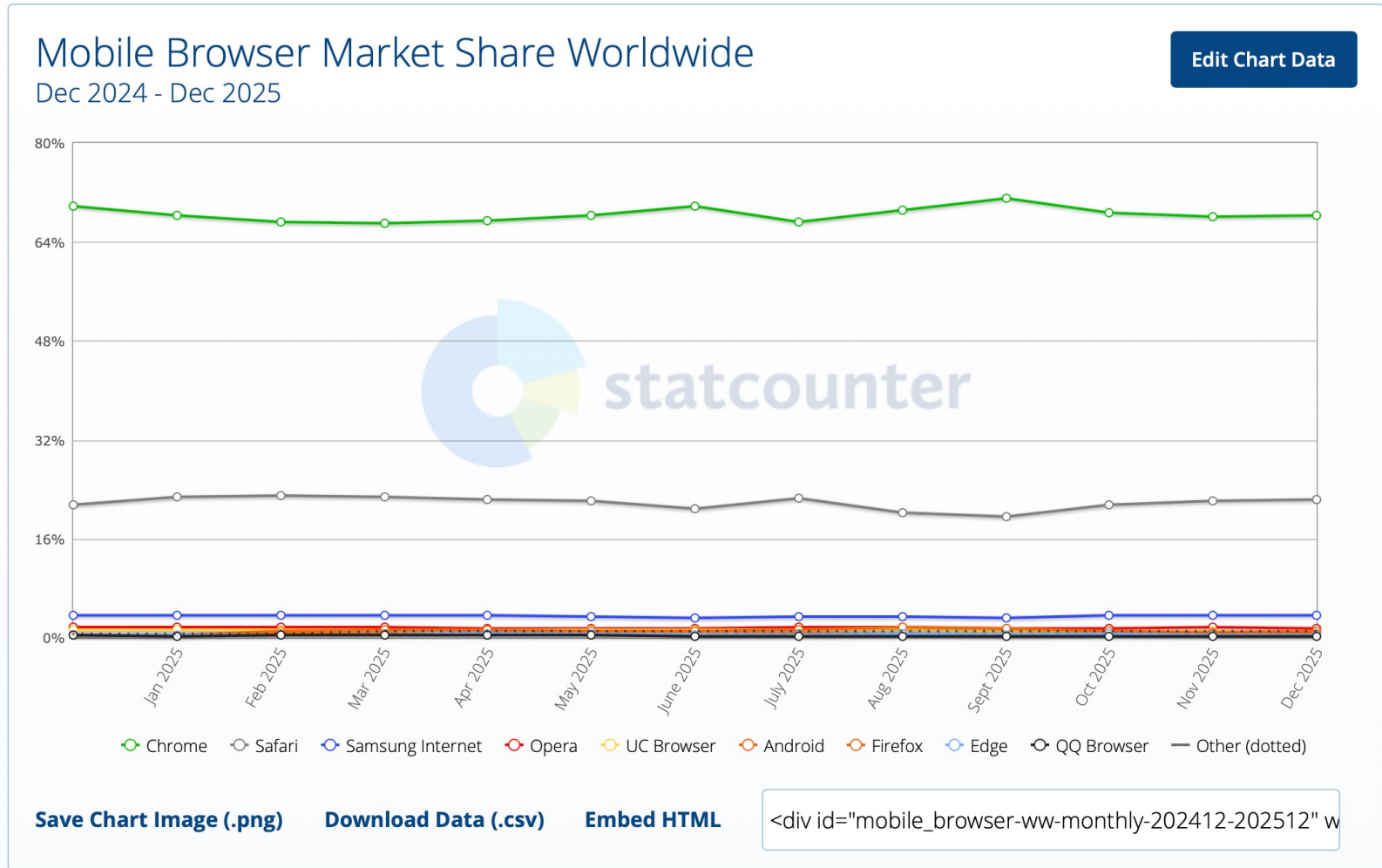
The Browser Wars - Desktop Statistics

2025	<u>Chrome</u>	<u>Edge</u>	<u>Firefox</u>	<u>Safari</u>	<u>Opera</u>
September	78.6 %	12.3 %	3.5 %	3.9 %	1.0 %
August	80.4 %	11.2 %	3.5 %	3.4 %	0.9 %
July	80.5 %	11.3 %	3.6 %	3.3 %	0.8 %
June	79.8 %	11.4 %	3.6 %	3.7 %	0.9 %
May	79.1 %	11.6 %	3.7 %	4.0 %	1.0 %
April	79.1 %	11.4 %	3.8 %	4.2 %	0.8 %
March	78.5 %	11.6 %	3.8 %	4.2 %	0.8 %
February	78.4 %	11.7 %	3.9 %	4.3 %	0.9 %
January	79.4 %	10.7 %	3.9 %	3.9 %	1.4 %

Conclusion of the above study:
- Chrome is the clear winner
- Edge comes second, with Chrome engine
- Firefox next, but losing ground
- Safari and Opera having small percentages
- WebKit-related total over 90%

http://www.w3schools.com/browsers/browsers_stats.asp

The Browser Wars - Mobile Statistics



StatCounter Global Stats, See <http://gs.statcounter.com>

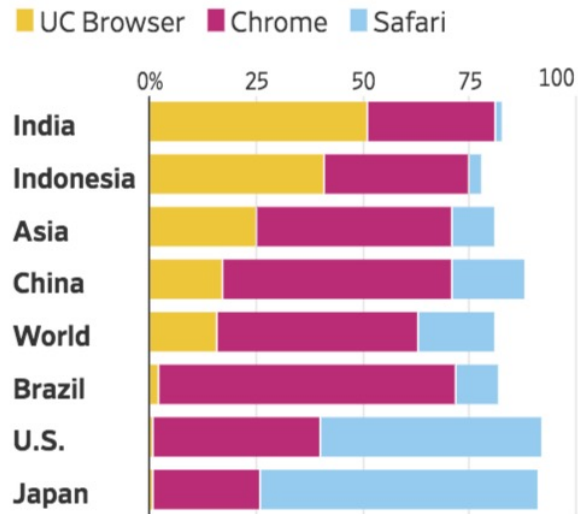
Chrome has the lead, followed by Safari. Samsung Internet, Opera and UC Browser from UCWeb of Alibaba Group of China have small percentages.

The Browser Wars Comparison (cont'd)

Browser for the Next Billion

Alibaba's mobile browser, UC Browser, has a larger market share than Google's Chrome in India and Indonesia, where many of the world's 'next billion users' are getting online for the first time.

Mobile web browser market share

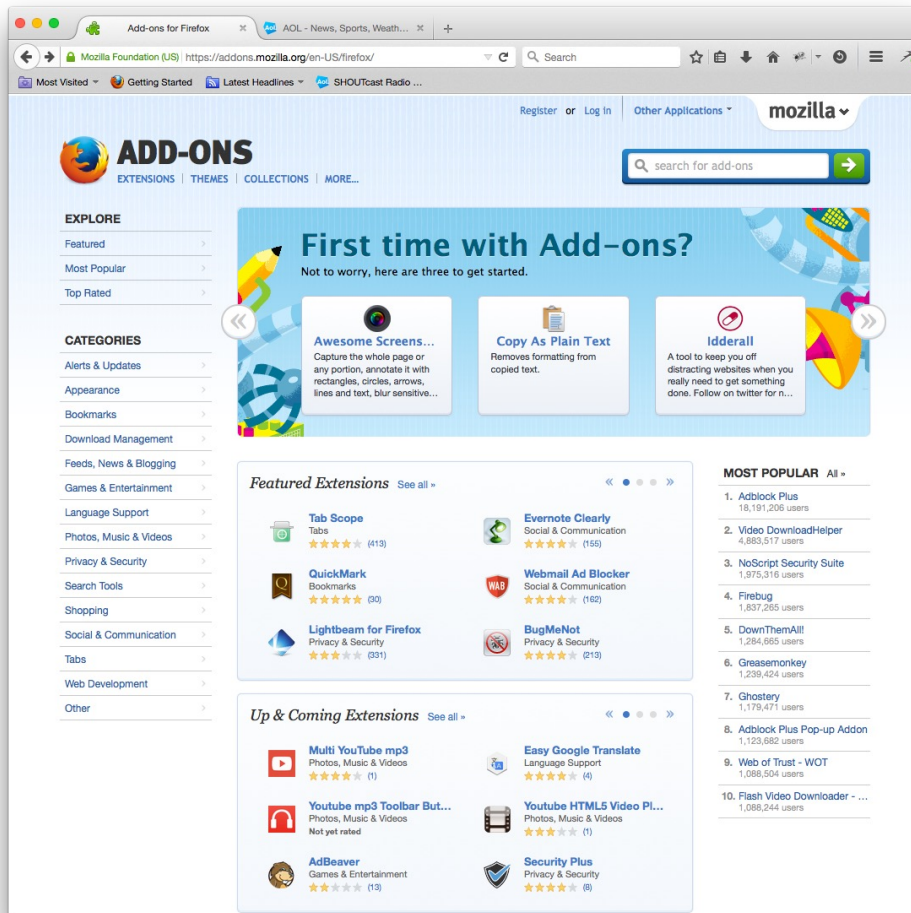


Note: Data Oct.-Dec. 2016 through Oct.-Dec. 2017.

Source: StatCounter

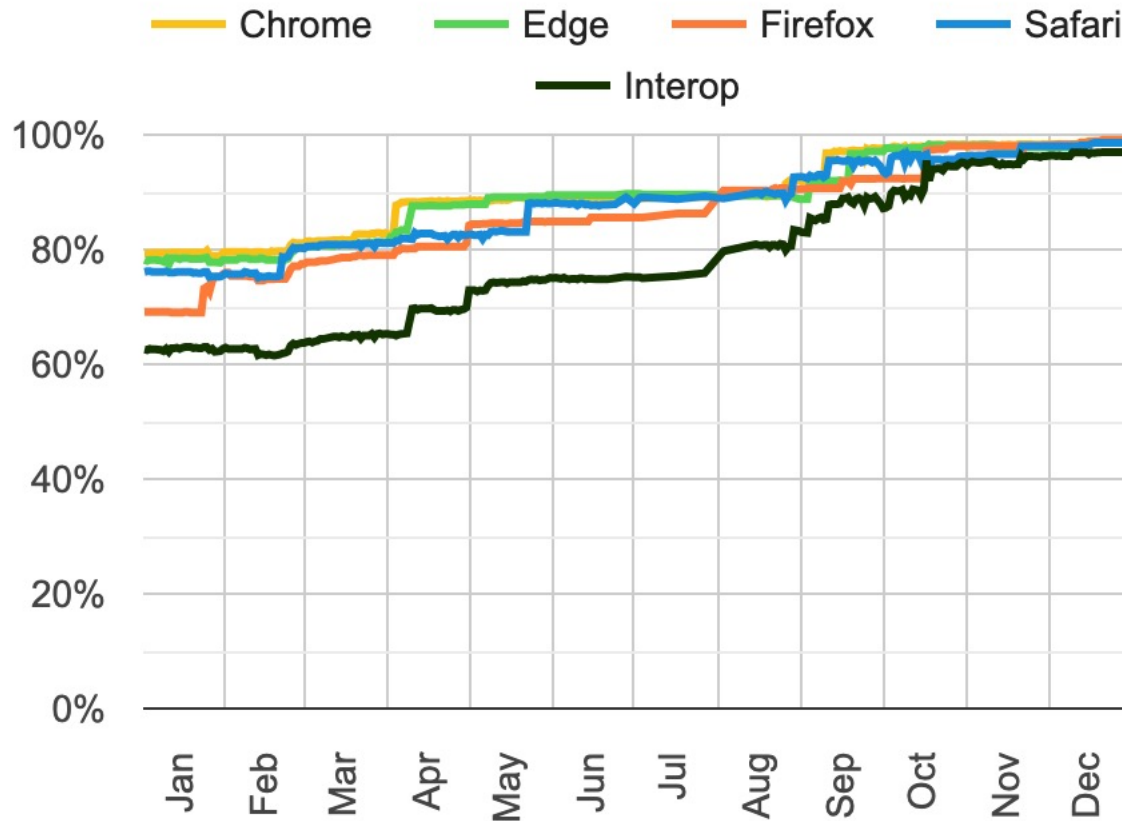
StatCounter Global Stats, Oct.-Dec. 2016 through Oct.-Dec. 2017, See <https://www.wsj.com/articles/a-browser-youve-never-heard-of-is-dethroning-google-in-asia-1514808002>

Browsers Have Many Plugins Available



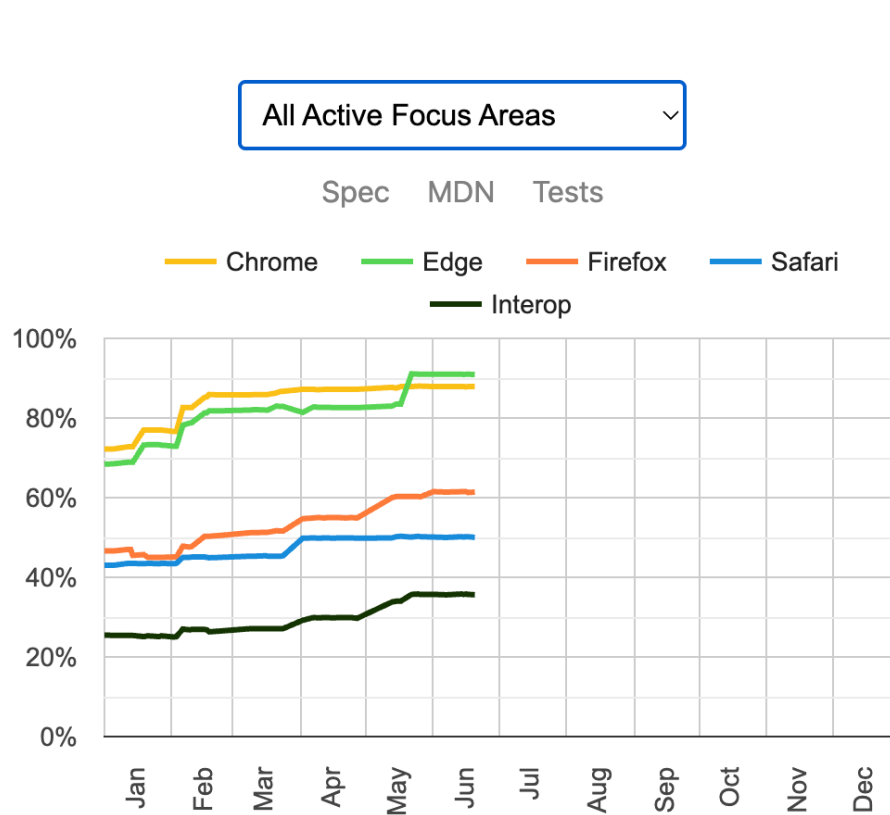
- Plug-ins are now called Browser “Extensions”
- Firefox plug-ins that will be especially useful in this course are:
 - HTTP Header Live
 - <https://addons.mozilla.org/en-US/firefox/addon/http-header-live/>
- HTML5 does away with most video / audio plug-ins

Browsers Interoperability 2024



<https://wpt.fyi/interop-2024>

Browsers Interoperability 2025



Active Focus Areas					INTEROP
backdrop-filter	95.6%	84.0%	84.0%	85.5%	75.3%
Core Web Vitals	74.4%	99.1%	59.8%	4.6%	4.6%
CSS anchor positioning	98.2%	98.2%	14.7%	14.6%	14.6%
<details> element	100%	100%	73.6%	72.4%	51.4%
Layout	99.9%	99.9%	96.0%	94.4%	91.5%
Modules	94.3%	94.3%	98.6%	84.9%	78.7%
Navigation API	98.7%	98.0%	0.4%	0.4%	0.4%
Pointer and mouse events	92.8%	92.8%	90.4%	77.2%	74.8%
Remove mutation events	100%	100%	40.0%	30.0%	30.0%
@scope	100%	96.8%	14.6%	79.1%	11.5%
scrollend event	100%	100%	92.8%	1.7%	1.7%
Storage Access API	69.6%	63.7%	81.2%	33.0%	28.4%
text-decoration	91.1%	91.1%	100%	73.8%	73.8%
URLPattern	97.7%	97.7%	0.1%	0.1%	0.1%
View Transition API	99.4%	97.5%	4.5%	90.2%	3.8%
WebAssembly	50.0%	50.0%	50.0%	0.0%	0.0%
Web compat	96.3%	91.4%	86.1%	63.7%	60.0%
WebRTC	13.2%	73.5%	80.6%	72.8%	5.5%
Writing modes	100%	100%	100%	71.9%	71.9%
TOTAL	87.9%	90.9%	61.4%	50.0%	35.6%

<https://wpt.fyi/interop-2024>

Evolution of Web Sites

Client-centric Static	Server Applications Databases Dynamic web pages	Web services Multiple layers Business and service Integration	Service Oriented Arch. (SOA) Client-centric	Multi-platform (desktop, tablet, phone) Client-centric	IoT, Wearables, Cloud computing, Serverless Arch. (BaaS, FaaS) Docker	Web Assembly, WebGPU, SQLite, File System Access API, Passkeys
HTML Scripts CGI	ODBC, JDBC ASP Applets, ActiveX	XML, WML, SQL, .NET COM+, Beans	Ajax, Web 2.0, JSON	HTML5, CSS3, JS gestures	JS Frameworks AWS, GCP, Azure Microservices containers	Web Transitions API, Badging API

1 st gen	2 nd gen	3 rd gen	4 th gen	5 th gen	6 th gen	7 th gen
1991	1997	2000	2005	2008	2014	2020