

Web Browsers

Jeszenszky Péter

Faculty of Informatics, University of Debrecen

November 5, 2024

Terms of Use

This work is licensed under a [Creative Commons](#) “[Attribution 4.0 International](#)” license.



History (1)

The first web browser:

- WorldWideWeb (Tim Berners-Lee, December 25, 1990)
- See:
 - Tim Berners-Lee. *The WorldWideWeb browser*.
<https://www.w3.org/People/Berners-Lee/WorldWideWeb.html>
 - Jay Hoffmann. *The Web's First (And Second) Browser*. September 25, 2017. <https://thehistoryoftheweb.com/web-first-and-second-browser/>

History (2)

Further information:

- Jay Hoffmann. *The History of the Web*.
<https://thehistoryoftheweb.com/>
 - Chapter 2: Browsers
 - Chapter 10: Browser Wars

Recommended Reading about How Browsers Work (1)

- Tali Garsiel, Paul Irish. *How Browsers Work: Behind the scenes of modern web browsers*. August 5, 2011.
<https://web.dev/howbrowserswork/>
- Anton Paras. *Notes on “How Browsers Work”*. December 11, 2017.
<https://codeburst.io/how-browsers-work-6350a4234634>
- MDN Web Docs – Populating the page: how browsers work

Recommended Reading about How Browsers Work (2)

- Eric Lawrence (ericlaw). *Demystifying Browsers*. October 29, 2024. <https://textslashplain.com/2020/02/09/demystifying-browsers/>
- Mariko Kosaka. *Inside look at modern web browser (part 1-4)*. 2018.
 - <https://developer.chrome.com/blog/inside-browser-part1>
 - <https://developer.chrome.com/blog/inside-browser-part2>
 - <https://developer.chrome.com/blog/inside-browser-part3>
 - <https://developer.chrome.com/blog/inside-browser-part4>
- Pavel Panchekha, Chris Harrelson. *Web Browser Engineering*. Oxford University Press, 2024. <https://browser.engineering/>
 - Repository: <https://github.com/browserengineering/book>

Structure of a Web Browser

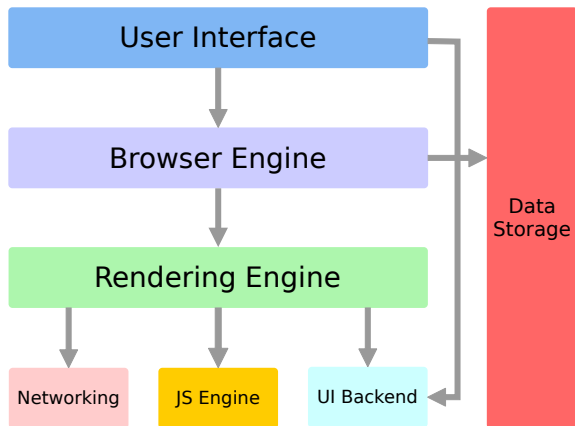


Figure 1: Source: <https://web.dev/howbrowserswork/>

Browser Engine vs Rendering Engine

- The terms “browser engine”, “rendering engine”, and “layout engine” are often used interchangeably, although it is technically inaccurate.
- Gecko is a special case, since it consists of a browser engine and a rendering engine.

Components of Web Browsers: User Interface

- The user interface includes the address bar, buttons (e.g., Back, Forward, Refresh), menus, etc.
- It includes every visible part of the web browser except the area in which the web page is displayed.

Components of Web Browsers: Browser Engine

- Provides a high-level interface for querying and manipulating the rendering engine.
- Acts as a middleman between the user interface and the rendering engine.
- Communicates with the data storage component.

Components of Web Browsers: Rendering Engine

- A core component of web browsers.
- Its function is to create a visual representation of web content (e.g, a HTML or XML document) to be displayed on an output device.
- The rendering of content happens according to the formatting rules specified.

Components of Web Browsers: Networking Component (1)

- The networking component is responsible for handling network communication, including:
 - HTTP requests
 - WebSocket API
 - WebRTC
- It provides a platform-independent interface, behind which platform specific APIs are used.

Components of Web Browsers: Networking Component (2)

Further information:

- WebSockets: [WebSockets Living Standard \(WHATWG\)](#)
- WebRTC:
 - <https://webrtc.org/>
 - [WebRTC: Real-Time Communication in Browsers \(W3C Recommendation, 8 October 2024\)](#)

Components of Web Browsers: JavaScript Engine (1)

The JavaScript engine is responsible for executing JavaScript code.

Components of Web Browsers: JavaScript Engine (2)

Major JavaScript engines:

- SpiderMonkey (written in: C++/Rust/JavaScript; license: Mozilla Public License 2.0) <https://spidermonkey.dev/>
 - The JavaScript engine of the Mozilla Project.
- V8 (written in: C++; license: New BSD License) <https://v8.dev/>
<https://chromium.googlesource.com/v8/v8.git>
 - The JavaScript engine of Chromium.
- JavaScriptCore (written in: C++; license: LGPLv2) <https://developer.apple.com/documentation/javascriptcore>
<https://trac.webkit.org/wiki/JavaScriptCore>
 - The JavaScript engine of Apple.

Components of Web Browsers: JavaScript Engine (3)

Other JavaScript engines:

- GraalVM JavaScript (written in: C/C++; license: Universal Permissive License v1.0)
<https://www.graalvm.org/latest/reference-manual/js/>
<https://github.com/oracle/graaljs>
 - JavaScript engine to execute JavaScript and Node.js applications in GraalVM.

Components of Web Browsers: UI Back-end

- The UI back-end is responsible for drawing in the web browser.
- It is used both for displaying user interface elements and the web page.
- It provides a platform-independent interface, behind which platform specific APIs are used.

Components of Web Browsers: Data Storage (1)

Data storage is responsible for persisting data locally, it is used for:

- HTTP cookies
- HTTP caching
- Indexed Database API (IndexedDB)
- Web Storage API

Components of Web Browsers: Data Storage (2)

Further information:

- HTTP cookies:
 - Adam Barth. *RFC 6265: HTTP State Management Mechanism*. April 2011. <https://www.rfc-editor.org/rfc/rfc6265>
- HTTP caching:
 - Roy T. Fielding (ed.), Mark Nottingham (ed.), Julian F. Reschke (ed.). *RFC 9111: HTTP Caching*. June 2022. <https://www.rfc-editor.org/rfc/rfc9111>
- Indexed Database API (IndexedDB):
 - [Indexed Database API 3.0 \(W3C Working Draft, 15 October 2024\)](#)
- Web Storage API:
 - [HTML Living Standard – Web storage](#)

Major Rendering Engines

- WebKit
- Blink
- Gecko

Rendering Engines: WebKit

- Website: <https://webkit.org/>
- Repository: <https://github.com/WebKit/WebKit>
- Developer: Apple
- License: LGPLv2/Simplified BSD License
- Written in: C++
- WebKit-based software:
 - Safari
 - Google Chrome for iOS
 - Firefox for iOS
 - GNOME Web

Rendering Engines: Blink

- Website: <https://www.chromium.org/blink/>
- Repository: <https://chromium.googlesource.com/chromium/blink/>
- Developer: Chromium Project (Google)
- License: Simplified BSD License/New BSD License/LGPLv2.1
- Written in: C++
- Originally, Blink was born as a fork of the WebCore component of WebKit.
- Blink-based software:
 - Chromium
 - Google Chrome, Google Chrome for Android
 - Chromium-based Microsoft Edge
 - Opera
 - Vivaldi

Rendering Engines: Gecko

- Website: https://wiki.mozilla.org/Gecko:Home_Page
- Repository: <https://hg.mozilla.org/mozilla-central/>
- Developer: Mozilla Project
- License: Mozilla Public License 2.0
- Written in: C, C++, JavaScript, Rust
- Gecko-based software:
 - Firefox
 - Firefox for Android
 - SeaMonkey

Market Share of Browsers

- All platforms: <https://gs.statcounter.com/browser-market-share>
- Desktop: <https://gs.statcounter.com/browser-market-share/desktop>
- Mobile: <https://gs.statcounter.com/browser-market-share/mobile>

Major Desktop Browsers

- Chromium, Google Chrome
- Firefox
- Opera
- Safari
- Microsoft Edge

Comparison of Web Browsers

- [Browser comparison \(Can I use...\)](#)
- [Comparison of Web Browsers \(eylenburg.github.io\)](#)
- [PrivacyTests.org](#)
- [Compare Firefox with other browsers \(Mozilla\)](#)

Technical Documentation for Developers

- Chromium, Google Chrome: [For Developers](#)
- Firefox:
 - <https://developer.mozilla.org/en-US/docs/Mozilla/Firefox>
 - <https://firefox-source-docs.mozilla.org/>
- Microsoft Edge: [Microsoft Edge for Developers](#)

Major Desktop Browsers: Chromium

- Website: <https://www.chromium.org/Home/>
- Repository: <https://chromium.googlesource.com/chromium/src.git>
- Developer: Chromium Project (Google)
- License: New BSD License
 - See: <https://chromium.googlesource.com/chromium/src/+master/LICENSE>
- Written in: C, C++, JavaScript
- Platform: Linux, macOS, Windows
- Rendering engine: Blink
- JavaScript engine: V8
- Technical information: `chrome://system`

Major Desktop Browsers: Google Chrome

- Website: <https://www.google.com/chrome/>
- Developer: Google
- License: proprietary (see: `chrome://terms`)
- Written in: C, C++, JavaScript
- Platform: Linux, macOS, Windows
- Rendering engine: Blink
- JavaScript engine: V8
- Technical information: `chrome://system`
- Developer edition: [Google Chrome for developers](#)

Major Desktop Browsers: Chromium and Google Chrome (1)

Google Chrome is based on Chromium.

Major Desktop Browsers: Chromium and Google Chrome

(2)

- Differences between the two browsers:
 - Different logos.
 - They are equipped with a different set of audio and video codecs.
 - User data are stored stored in different directories.
 - Chromium: `~/.cache/chromium`, `~/.config/chromium`
 - Google Chrome: `~/.cache/google-chrome`,
`~/.config/googlechrome`
 - Operating system distributions may modify the source code of Chromium, packaging also depends on the distribution.
- See: [The Difference between Google Chrome and Chromium on Linux](#)

Major Desktop Browsers: Firefox

- Website: <https://www.mozilla.org/firefox/>
- Repository: <https://hg.mozilla.org/mozilla-central/>
<https://searchfox.org/mozilla-central/source>
- Developer: Mozilla Project
- License: Mozilla Public License 2.0 (see: `about:license`)
- Written in: C, C++, JavaScript, Rust
- Platform: Linux, macOS, Windows
- Rendering engine: Gecko
- JavaScript engine: SpiderMonkey
- Technical information: `about:support`
- Developer edition: [Firefox Developer Edition](#)

Major Desktop Browsers: Opera

- Website: <https://www.opera.com/>
- Developer: Opera Software
- License: proprietary
 - See: <https://legal.opera.com/eula/computers/>
- Written in: C++
- Platform: Linux, macOS, Windows
- Rendering engine: Blink
- JavaScript engine: V8
- Technical information: `opera://system`
- Developer edition: [Opera Developer](#)

Major Desktop Browsers: Safari

- Website: <https://www.apple.com/safari/>
- Developer: Apple
- License: proprietary
- Written in: C++, Objective-C
- Platform: macOS
- Rendering engine: WebKit
- JavaScript engine: JavaScriptCore
- Developer edition: [Safari Technology Preview](#)

Major Desktop Browsers: Microsoft Edge (1)

- Website: <https://www.microsoft.com/edge>
- Developer: Microsoft
- License: proprietary (see: `edge://terms`)
- Platform: Linux (2020-), macOS (2019-), Windows
- Rendering engine: EdgeHTML (-2019), Blink (2019-)
- JavaScript engine: Chakra (-2019), V8 (2019-)

Major Desktop Browsers: Microsoft Edge (2)

Chromium-based Edge:

- Announcement:
 - Joe Belfiore. *Introducing Microsoft Edge Beta: Be one of the first to try it now.* August 20, 2019.
<https://blogs.windows.com/windowsexperience/2019/08/20/introducing-microsoft-edge-beta-be-one-of-the-first-to-try-it-now/>
- Written in: C, C++, JavaScript

Web Developer Tools

- Chromium, Google Chrome, Opera: [Chrome DevTools](#)
- Firefox: [Firefox DevTools User Docs](#)
- Safari: [Safari Developer Help](#)
- Chromium-based Edge: [Microsoft Edge DevTools documentation](#)

Browser Add-ons

- A browser add-on is a software module for customizing the appearance or behavior of a web browser.
- Types of add-ons: extensions, language packs, themes, plugins

Browser Extensions

A browser extension is a browser add-on that alters the behavior of a web browser, i.e., adds new functionality or modifies existing functionality.

Available Browser Add-ons

- Chromium, Google Chrome: [Chrome Web Store](#)
- Firefox: [Firefox Add-ons](#)
- Opera: [Opera add-ons](#)
- Safari: Mac App Store
- Chromium-based Edge: [Microsoft Edge Addons](#)

Standardization Efforts on Browser Extensions

WebExtensions Community Group (W3C)

- Webhely: <https://www.w3.org/community/webextensions/>
- Tároló: <https://github.com/w3c/webextensions/>
- Initiators of the group: Apple, Google, Microsoft, Mozilla
- The goal of the community group is to specify and standardize a common browser extension platform that enable developers to not have to completely rewrite their extensions to work in different browsers.

Developing Browser Extensions (1)

- Chromium, Google Chrome, Chromium-based Edge:
 - See: <https://developer.chrome.com/docs/extensions>
- Firefox:
 - Extensions for Firefox are built using the WebExtensions API, a cross-browser system for developing extensions.
 - This technology is, to a large extent, compatible with the extension API supported by Chromium-based browsers. In most cases, extensions written for Chromium-based browsers run in Firefox with just a few changes.
 - See: [Browser Extensions \(MDN\)](#)

Developing Browser Extensions (2)

- Opera:
 - See: [Extensions Documentation](#)
- Safari:
 - Extensions for the Safari browser must be created with the Xcode IDE.
 - See: [Safari Extensions](#)

Developing Browser Extensions (3)

- Extensions are built on web technologies, such as HTML, CSS, and JavaScript.
- In the case of the major browsers discussed, they are bundles of HTML, CSS, JavaScript, and other files (e.g., images).
- Chromium, Google Chrome, Firefox, Opera, Microsoft Edge:
 - Every extension has a manifest file (`manifest.json`) that contains metadata and ties together the components.

Developing Browser Extensions (4)

Further information:

- [Firefox Extension Workshop](#)

Headless Browsers (1)

- A headless browser is a web browser without a graphical user interface.
- It can be controlled programmatically.
- Possible applications:
 - Automated testing of web applications
 - Automating interaction with web pages
 - Taking screenshots of web pages
 - Web scraping
 - ...
- See: https://en.wikipedia.org/wiki/Headless_browser

Headless Browsers (2)

The Chromium, Google Chrome, and Firefox browsers can run in headless mode.

- Chromium, Google Chrome:
 - Mathias Bynens, Peter Kvittek. *Chrome Headless mode*.
<https://developer.chrome.com/docs/chromium/headless>
 - <https://chromium.googlesource.com/chromium/src/+/lkgr/headless/README.md>
- Firefox:
 - [Using Headless Mode in Firefox](#)

Headless Browsers (3)

Taking a screenshot where output is written to the file `screenshot.png`:

- Google Chrome:

```
google-chrome --headless --screenshot \  
--window-size=1920,1080 https://www.w3.org/
```

- Firefox:

```
firefox -screenshot --window-size=1920,1080 \  
https://www.w3.org/
```

Headless Browsers (4)

Free and open source headless browsers:

- **HtmlUnit** (written in: Java; license: Apache License 2.0)
<https://www.htmlunit.org/> <https://github.com/HtmlUnit/htmlunit>
 - Provides a way to simulate a browser for testing purposes and is intended to be used within another testing framework such as JUnit.
- **Puppeteer** (written in: JavaScript; license: Apache License 2.0)
<https://pptr.dev/> <https://github.com/puppeteer/puppeteer>
 - JavaScript library which provides a high-level API to control Chrome or Firefox.

See: [dhamaniasad/HeadlessBrowsers: A list of \(almost\) all headless web browsers in existence](#)

Text-based Web Browsers

- Lynx (platform: Linux, macOS, Windows; license: GPLv2)
<https://lynx.invisible-island.net/>
 - The oldest web browser that is still in general use and is being actively developed.
- Links (platform: Linux, macOS, Windows; license: GPLv2)
<http://links.twibright.com/>
 - It can also run in graphics mode, see the `-g` command line option.
- Browsh (platform: Linux, macOS, Windows; license: LGPLv2.1)
<https://www.brow.sh/> <https://github.com/browsh-org/browsh>
 - Creates a purely text-based version of web pages using Firefox running in headless mode.

Major Mobile Browsers

- Google Chrome (Android, iOS)
- Safari for iOS
- Samsung Internet for Android
- UC Browser (Android, iOS)

Mobile Browsers for iOS

- Web browsers for iOS must use the WebKit rendering engine and the JavaScriptCore engine.
- See: [App Store Review Guidelines – Software Requirements](#)
Apps that browse the web must use the appropriate WebKit framework and WebKit Javascript.

Major Mobile Browsers: Chrome

- Website: <https://www.google.com/chrome/>
- Developer: Google
- License: proprietary
- Platform: Android, iOS
- Rendering engine: Blink (Android)/iOS WebKit (iOS)
- JavaScript engine: V8 (Android)/JavaScriptCore (iOS)

Major Mobile Browsers: Safari for iOS

- Website: <https://www.apple.com/safari/>
- Developer: Apple
- License: proprietary
- Rendering engine: iOS WebKit
- JavaScript engine: JavaScriptCore

Major Mobile Browsers: Samsung Internet for Android

- Website: <https://www.samsung.com/us/support/owners/app/samsung-internet>
- Developer: Samsung
- License: proprietary
- Rendering engine: Blink
- JavaScript engine: V8
- Further information for developers:
<https://developer.samsung.com/internet>

Major Mobile Browsers: UC Browser

- Website: <https://www.ucweb.com/>
- Developer: UCWeb, a subsidiary of Alibaba
- License: proprietary
- Platform: Android, iOS
- Rendering engine: Blink (Android)/iOS WebKit (iOS)
- JavaScript engine: V8 (Android)/JavaScriptCore (iOS)

Major Mobile Browsers: Opera

- Opera for Android/iOS, Opera Mini (platform: Android), Opera GX (platform: Android, iOS)
- Website: <https://www.opera.com/>
- Developer: Opera Software
- License: proprietary
- Rendering engine: Blink (Android)/iOS WebKit (iOS)
- JavaScript engine: V8 (Android)/JavaScriptCore (iOS)

The about URI Scheme (1)

- The about URI scheme is widely used by Web browsers to designate access to their internal resources, such as settings, application information, or hidden built-in functionality (e.g., Easter eggs).
 - See: S. Moonesamy (ed.). *RFC 6694: The “about” URI Scheme*. August 2012. <https://www.rfc-editor.org/rfc/rfc6694>
- Example:
 - The `about:blank` URI references a blank page.

The about URI Scheme (2)

- Some browsers map about URIs to their own equivalents replacing the scheme name (i.e., about) with their name (such as chrome, opera, or edge).
 - For example, Chromium and Google Chrome uses the URI `chrome://about` instead of `about:about`.
- An exception is `about:blank`, that is left unchanged.

The about URI Scheme (3)

Browser support:

- Chromium, Google Chrome:
 - `about:` is mapped to `chrome://`
 - The list of Chrome URIs: `chrome://about`
 - Examples: `chrome://bookmarks`, `chrome://dino`, `chrome://flags/`, `chrome://history`, ...
 - For debugging purposes: `chrome://crash`, `chrome://quit`, `chrome://restart`, ...
- Firefox:
 - The list of about URIs: `about:about`
 - Examples: `about:config`, `about:downloads`, `about:rights`, ...

The about URI Scheme (4)

Browser support: (continued)

- Opera: Supports a subset of chrome URIs and a few additional ones, but the scheme name is replaced with opera.
 - Examples: `opera://downloads`, `opera://emoji-picker`, `opera://flags`, `opera://history`, ...
- Safari: Only `about:blank` is recognized.

The about URI Scheme (5)

Browser support: (continued)

- Chromium-based Edge:
 - `about:` is mapped to `edge://`
 - The list of Edge URIs: `edge://about`
 - Examples: `edge://downloads`, `edge://flags`,
`edge://history/all`, `edge://history/today`,
`edge://history/yesterday`, ...

Support for Web Technologies in Browsers

- Current support:
 - Can I use... Support tables for HTML5, CSS3, etc
 - Chrome Platform Status
- Roadmap:
 - Mozilla Specification Positions
 - WebKit Standards Positions