Technical analysis of browser fingerprinting techniques based on FingerprintJS

James Bergfeld
Technical University Munich
Munich, Germany
j.bergfeld@tum.de

Samuel Scheit
Technical University Munich
Munich, Germany
tum@samuelscheit.com

1. Introduction

In an age of progressive digitalization and data collection, the commercial value of web users' data is continuously increasing. Personal content delivery, bot detection, user authentication and personalized advertising rely heavily on the ability to reliably identify users across the web. Most data vendors currently use so-called "third-party cookies" to identify and discern between users. However, users and browser developers are becoming more aware of the threats that cookies pose to digital privacy. Most major browser vendors, therefore, have announced the deprecation of third-party cookies in order to reduce the privacy impact on users, with Google's Chrome browser being the most recent example. These latest changes in the world of web browsing and privacy protection have created a rising demand for other methods of web tracking and traffic analysis, which has lead to increasing usage of browser fingerprinting technology on the internet. [1]

1. Definition

Websites use browser fingerprinting to create a unique identifier of each website visitor by collecting data about the visitor's device and browser configuration and combining them to create a unique "fingerprint". The browser properties considered when creating a fingerprint are herein referred to as parameters of the fingerprint. A parameter may describe browser settings, installed extensions, system configuration, available fonts, audio devices, or others. They may also be derived from the hardware, software or network stack by analyzing the results of tests, such as image rendering, that can yield distinct results depending on the devices attached to and drivers installed on the system.

Because of their statelessness, fingerprints can even identify users across different websites, visits and between regular and private browsing mode (or incognito mode) without the reliance on login- and tracking-cookies stored on a user's browser, which makes fingerprinting a suitable alternative to other tracking methods.

2. Related Academic Work

There are a number of studies concerning browser fingerprinting technology, browser configurations and privacy implications of fingerprinting. Publications such as Olejnik et al. [2] investigate individual features of a browser, e.g. the battery API, to determine how information derived from them can be used to uniquely identify a device. Other works are mainly concerned with devising custom browser fingerprinting algorithms on parameter sets of varying sizes to demonstrate the effectiveness of browser fingerprinting. In 2010, Eckersly [3] was able to successfully identify 83.6% fingerprints from a sample of 470,161 using a simple algorithm with a manually chosen set of eight parameters. Expanding on the former, Pugliese et al. [4] collected user fingerprints over a span of three years to evaluate the identifiability of possible parameter combinations from a total of 305 collected parameters. Out of 43,025 fingerprints collected from 652 users, 94.5% were deemed to be trackable using feature sets determined by an algorithm optimized based on a dataset of similar size.

3. FingerprintJS

Despite their successful statistical demonstrations, the aforementioned algorithms are not a suitable basis for conclusions about the practical implementation of modern browser fingerprinting. This is due to user sampling biases and the usage of deprecated features such as the Adobe Flash suite for fingerprint generation. By conducting an analysis of a widely used fingerprinting library, direct insight into modern standards of browser fingerprinting technology may be gained. FingerprintJS is currently the most widely used browser fingerprinting library [5]. This is evidenced by npm download statistics, which show FPJS as the most popular JavaScript browser fingerprinting library [6]. As such, it's a suitable subject to investigate the implementation of browser fingerprinting in practice. FPJS offers two fingerprinting solutions: FingerprintJS, an open-source library with moderate coverage of different browser types and configurations as well as FingerprintJS Pro, a subscription-based closed-source library that uses a greater set of parameters and claims a 99.5% rate of (re-)identification. The analysis within this paper focuses solely on the latter because of its higher accuracy that is more likely to match the industry standards that are to be examined.

FPJS Pro is a JavaScript library that runs on a user's browser, collecting a total of 117 parameters from browser API calls and statistical properties. The parameters are sent to the FPJS server and parsed using an algorithm that isn't publicly available. The algorithm returns an ID or fingerprint based on the data that uniquely identifies the browser being fingerprinted.

Since the source code of FPJS Pro isn't available, conducting an analysis on the full implementation of FingerprintJS Pro isn't possible. Conclusive results can instead be achieved by reverse-engineering a public instance of the FingerprintJS Pro library hosted on FingerprintJS' demonstration website[1].

FPJS is the most popular JavaScript browser fingerprinting library according to npm downloads. [6]

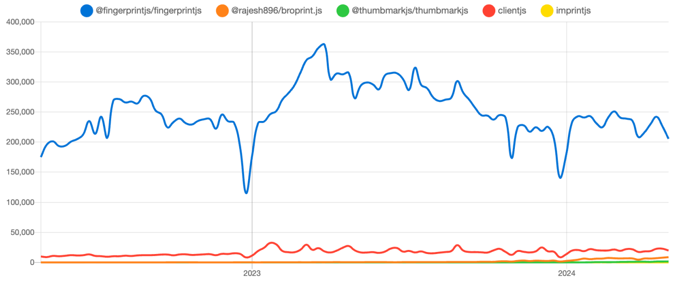


Figure 1: NPM downloads per day, comparison of different JS fingerprinting libraries (as of May 2024). From [6].

2. Methodology

1. Reverse-engineering FingerprintJS Pro

Since the source code of FPJS Pro isn't freely available, a technical analysis necessitates reverse-engineering a publicly hosted instance of FPJS Pro, such as their demonstration website.

A snapshot of the demonstration page is captured to ensure a consistent and stable environment for analysis and prevent changes to the code base caused by updates to the FPJS Pro library. The snapshot used for this paper was created on May 8, 2024, capturing version 3.9.4 of FPJS Pro. It is assumed that library files are static and not modified per device and are the same versions for mobile and desktop clients. The obfuscated JavaScript library is retrieved from the site by visiting it with an up-to-date Chrome browser and extracting the code using Chrome's built-in developer tools.

Using a JavaScript Proxy object[2] a list of API- and function calls can be captured to give an overview of the general behavior of the library. This aids in identifying entry points for later reverse engineering.

Analysis of the network calls made when running FPJS reveals that fingerprints are generated by collecting data about the browser setup using client-side JavaScript and sending it to a remote server. The server parses the data and computes a fingerprint. The client payload is serialized before being sent and can't be read in plaintext. Therefore the request needs to be traced back to the caller function that generates the payload to decipher its contents. The browser's internal JavaScript debugger can be used to inspect the payload in plaintext format before it is serialized. The plaintext payload can be used to identify the functions used to measure each parameter value in the user's browser. These functions are then analyzed in detail, with particular regard to the browser APIs used and their processing.

The parameters collected by FingerprintJS Pro and the results of their analysis are described in Section 3.1.

2. Implementing a custom library

The actual fingerprints are generated with API calls to a closed-source server. It is therefore infeasible to reverse-engineer the actual fingerprint generation algorithm. A data-driven approach to developing an algorithm that tracks users across visiting using fingerprints is presented in Section 3.2.

A representative dataset of fingerprints is necessary to assess the identifiability of browsers using the individual parameters collected by FingerprintJS. To create such a dataset, fingerprints are gathered from real browsers by deploying a custom fingerprinting library on a web server. The website displays a consent notice and a brief definition of browser fingerprinting. Upon accepting the data usage policy, the library gathers the same parameters as FingerprintJS and sends them back to the server, which in turn generates a fingerprint based on this data. Data was collected between July 9 and July 26, 2024.

To identify and evaluate fingerprints from multiple visits of the same browser, a unique random ID is generated and stored on the client's `localstorage`, cookies, `indexedDB` and the `filesystem` API. The ID persists across visits and is attached to every recorded fingerprint sent back to the server. This establishes a ground truth for measurements pertaining to the stability of fingerprinting parameters as defined in Section 3.2.3. It is assumed that participants are of good faith and don't send arbitrary requests with invalid visitor IDs or spoofed values aside from those spoofed by privacy-enhancing settings, such as Firefox Enhanced Tracking protection.

[1] https://fingerprint.com/demo/

[2] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Proxy

3. Devising a fingerprinting algorithm

After the custom library has collected the visitor's browser parameters, this data must be evaluated to determine whether the visitor is a known or a new visitor.

The collected visitor parameters must be compared with the database of all previous visitors. A probability must be calculated that indicates how closely the visitor matches an existing user in the database.

For this, the algorithm must implement a probability weight system to make sure that the fingerprint still matches the same user if the browser configuration slightly changes.

3. Results

1. Parameters

Reverse engineering the FPJS library revealed that 117 parameters are considered for browser fingerprint generation. Further investigation into the obfuscated JavaScript used to collect these parameters showed that the library retrieves a large number of static browser properties such as screen geometry, enabled languages, browser vendor and version. See Section 6.3 for a list of all browser properties and API accesses made by FPJS.

1) Statistical properties

FPJS enumerates properties of the global `navigator` and `window` objects and sends them to the server without further processing. The navigator interface consists of properties and functions used to describe the web browser (navigator) which are suitable for fingerprinting for obvious reasons. The global window object "is home to a variety of functions, namespaces, objects, and constructors which are not necessarily directly associated with the concept of a user interface window" [7]. The values of these properties have a variety of implications some of which (such as the screen's color depth) may also be used to identify a user. Checks are made for global JavaScript objects specific to major browsers, e.g. the `chrome` object for Google Chrome and other Chromium-based browsers or the `ApplePaySession` object that is only present on devices supporting Apple Pay. None of the parameters retrieved by FPJS use the deprecated Adobe Flash API, as opposed to scientific studies such as [4] which rely on values retrieved using flash scripts.

2) Audio

The Web Audio API[3] is a browser API that can be used to artificially generate sounds and audio data.

However it can also be used to create a unique audio profile of the browser by:

1. Generating a series of tone signals with predefined properties such as frequency, volume and distortion.
2. Playing and recording the sound at the same time via the Web Audio API.
3. Analyzing the recorded audio data and encoding it as a hash to create a unique audio fingerprint.

FPJS first creates a triangle oscillator[4] tone signal with a frequency of `10.000` Hz. Then a compressor[5] is created with the following parameters:[6]

Value	Description
`-50db`	"value above which the compression will start taking effect"
`40db`	"value representing the range above the threshold where the curve smoothly transitions to the compressed portion."
`12db`	"amount of change needed in the input for a 1 dB change in the output."
`0s`	"the amount of time required to reduce the gain by 10 dB."
`0.25s`	"the amount of time required to increase the gain by 10 dB."

The open-source FPJS version uses a square oscillator with a base frequency of `1.000` Hz and an additional bi-quad filter[7]. This can be visualized by the following diagram that compares the audio values of different browser implementations:

[3] https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API

[4] https://developer.mozilla.org/en-US/docs/Web/API/BaseAudioContext/createOscillator

[5] https://developer.mozilla.org/en-US/docs/Web/API/BaseAudioContext/createDynamicsCompressor

[6] https://developer.mozilla.org/en-US/docs/Web/API/DynamicsCompressorNode/

[7] https://developer.mozilla.org/en-US/docs/Web/API/BiquadFilterNode

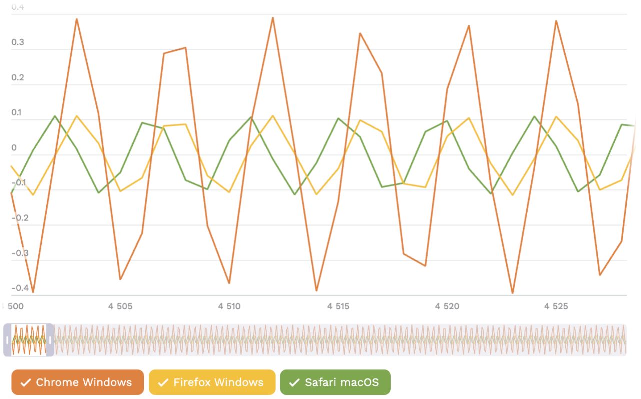


Figure 2: Audio API browser comparison[8]

Due to subtle differences in audio processing and playback of different browsers and systems, the recorded data will vary slightly from the original.

To prevent fingerprinting Firefox has the ability to disable the audio API[9].

3) WebRTC

Web Real-Time Communication (WebRTC[10]) is a browser API used to transmit video-/audio data in realtime over a (optionally peer-to-peer) connection.

a) IP Address

Interactive Connectivity Establishment (ICE)[11] is used in WebRTC to establish connections between clients that may be behind different network configurations or firewalls. This is achieved by connecting to a STUN (Session Traversal Utilities for NAT)[12] server which resolves possible ICE candidates (public IP address and port of the device). Additionally, the browser exposes the local IP address of the device's local area network (LAN)[13] to enable local connections in intranets.

This information can be retrieved by creating a new `RTCPeerConnection`[14] with a specified ICE server and a unique username to correlate the STUN connection with the current browser session.

By adding an `icecandidate`[15] event listener, the ICE candidates can be retrieved. The following string is an example candidate:

```
candidate:2079771436 1 udp 2122260223 123.234.1.250
50012 typ host generation 0 ufrag qRGm network-id 3
```

The candidate includes the IP address, port, network transport protocol, a unique identifier and other key-value parameters.

Specifically the local IP address can be used to recognize a device even if the public IP address changes e.g. when using a Virtual Private Network (VPN).

For this reason the TOR Browser has disabled the WebRTC protocol[16] and the Brave Browser has the ability to disable the usage of LAN IP addresses for WebRTC[17].

However, it should be noted local IP addresses are not unique and different LAN subnets have a limited address room. Specifically, 17.891.328 IPv4 addresses are reserved for LAN networks[18] and similar subnets and IP addresses are reused on many different networks and therefore can only be used for fingerprinting in conjunction with other parameters.

b) Codecs

Additionally the supported audio and video codecs can further help to fingerprint a device as different Browser and Device configurations support different codecs. The `RTCPeerConnection` created in the previous step can be queried via `connection.getStats()`[19] and returns a `RTCStatsReport`[20], which contains statistics of used audio and video codecs for the connection. For example the VP8 video codec is represented as the following object:

```
id: "HjD6dszXj",
type: "codec",
clockRate: 90000,
mimeType: "video/VP8",
direction: "sendrecv",
uri: "urn:ietf:params:rtp-hdrext:toffset",
```

and contains various information about the supported audio and video codecs e.g. support for CPU acceleration, forward error correction, stereo audio, bit-rate, codec version, frame size and other codec specific parameters[21].

These parameters are partially stable as browser updates might add support for different codecs, but processor specific codec acceleration does not change without a hardware modification.

c) Media devices

[8] https://fingerprint.com/blog/audio-fingerprinting/

[9] https://bugzilla.mozilla.org/show_bug.cgi?id=1708593

[10] https://webrtc.org/

[11] https://www.rfc-editor.org/rfc/rfc8445.html

[12] https://datatracker.ietf.org/doc/html/rfc5389

[13] https://www.ieee802.org/

[14] https://developer.mozilla.org/en-US/docs/Web/API/RTCPeerConnection

[15] https://developer.mozilla.org/en-US/docs/Web/API/RTCPeerConnection/icecandidate_event

[16] https://bugzilla.mozilla.org/show_bug.cgi?id=1432983

[17] https://support.brave.com/hc/en-us/articles/360017989132-How-do-I-change-my-Privacy-Settings#webrtc

[18] https://datatracker.ietf.org/doc/html/rfc1918#section-3

[19] https://developer.mozilla.org/en-US/docs/Web/API/RTCPeerConnection/getStats

[20] https://developer.mozilla.org/en-US/docs/Web/API/RTCStatsReport

[21] https://datatracker.ietf.org/doc/html/draft-ietf-payload-rtp-opus-04#section-6.1

WebRTC media devices are audio and video sources of the browser as well as audio playback and video display devices. These can be microphones, cameras, speakers and screens. WebRTC allows websites to access these devices via the `MediaDevices`[22] API.

The `navigator.mediaDevices.enumerateDevices()` API returns a “list of the currently available media input and output devices”.[23] Each media device[24] contains the following properties:

- `deviceId` (unique and persistent device identifier)
- `groupId` (optional identifier that groups multiple ids of the same physical device)
- `kind` (`"videoinput"`,`"audioinput"`,`"audiooutput"`)
- `label` (optional human readable name for the device)

Note that all device properties except `kind` are `null` if the website has never requested a media stream before[25]. FPJS uses this to determine the amount of audio and video devices the user has connected. As most websites don’t use the media stream, the devices don’t have a unique identifier and the media devices are a weak indicator for a unique fingerprint.

4) Speech synthesis

SpeechSynthesis[26] is part of the Web Speech Browser API[27] that allows websites to convert text to audio data (so-called Text-to-speech or TTS). The browser exposes the function `SpeechSynthesis.getVoices()`[28] that lists all locally and remotely available voices that can be used for TTS. Each voice contains the following properties:

- `voiceURI` (unique voice identifier)
- `name` (human-readable name of the voice)
- `lang` (ISO language code of the voice)
- `localService` (boolean indicating if the voice is locally available or a remote service)
- `default` (boolean indicating if the voice is set as default)

FPJS converts this list of voices to a string with `JSON.stringify`[29] and then hashes it with `Murmurhash3_128_x64` [8]. Additionally FPJS also sends a boolean indicating if any `"Google"` voices are installed on the system. As browsers return the list in order this hash is stable and only changes when the browser or the user adds a new voice to their system. However this hash only identifies specific browser versions and operating systems and is not unique. Firefox prevents this when `resistFingerprinting` is enabled by returning an empty list.

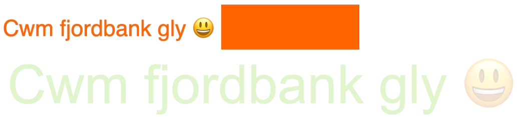
5) Canvas

Canvas is a browser API that allows websites to display dynamic 2D graphics. However it can also be used to create a unique identifier for the user’s graphic engine.

Canvas fingerprinting works by using the Canvas API to draw text, shapes, and images onto a canvas element and then extracting the pixel data to create a unique identifier. This identifier is based on subtle differences in the way browsers and devices render the same graphics instructions.

1. **Text Rendering:** By rendering specific text onto a hidden canvas element, the browser’s font rendering and antialiasing techniques contribute to the uniqueness of the fingerprint.
2. **Shape Drawing:** Drawing shapes and applying transformations (scaling, rotation, etc.) can reveal details about the graphics rendering engine and hardware acceleration capabilities.
3. **Image Manipulation:** Using images and manipulating them at a pixel level can reveal information about image processing algorithms and rendering accuracy.

FPJS uses the canvas API to render the following text, emojis and geometry:



The pixel data is then retrieved by calling `canvas.toDataURL()`[30] and hashed using `Murmurhash3_128_x64` [8]. However browsers such as Brave[31] or Firefox[32] add noise to the retrieved canvas data. To verify if canvas noise is added FPJS calls `toDataURL()` twice and compares the resulting buffers. Additionally FPJS uses an embedded image to check if the PNG[33] image data returned by `toDataURL()` matches the data of the embedded image.
If one of the checks fails, it can be concluded that noise was

[22] https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices

[23] https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/enumerateDevices

[24] https://developer.mozilla.org/en-US/docs/Web/API/MediaDeviceInfo

[25] https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getDisplayMedia

[26] https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis

[27] https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API

[28] https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis/getVoices

[29] https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify

[30] https://developer.mozilla.org/en-US/docs/Web/API/HTMLCanvasElement/toDataURL?retiredLocale=de

[31] https://github.com/brave/brave-browser/issues/9186

[32] https://support.mozilla.org/en-US/kb/firefox-protection-against-fingerprinting

[33] https://datatracker.ietf.org/doc/html/rfc2083

added to the canvas and the resulting hash is always unique per session and therefore unusable for identification without any further parameters.

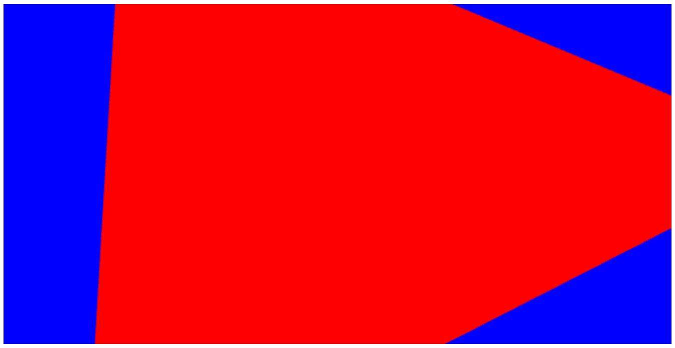
6) **WebGL**

The WebGL (Web Graphics Library)[34] is an additional API on top of the canvas element that allows websites to render 3D graphics, shaders and can also be used to create a unique identifier of the graphics engine.

a) **Rendering**

By rendering specific shaders and geometric shapes the GPU capabilities for texturing and rendering complexity can uniquely be identified.

FPJS uses WebGL fingerprinting by rendering the following graphic:



with the following shaders:

```
attribute vec2 p;
uniform float t;
void main() {
  float s = sin(t);
  float c = cos(t);
  gl_Position = vec4(p * mat2(c, s, -s, c), 1, 1);
}
```

```
void main() { gl_FragColor = vec4(1, 0, 0, 1); }
```

The data is retrieved, hashed and verified in the same way as with the Canvas API.

b) **Extensions**

Additionally the GPU capabilities can be queried by calling `context.getSupportedExtensions()`[35], `context.getContextAttributes()`[36], `context.getParameter()`[37] and `context.getExtension()`[38] functions of the `WebGLRenderingContext`[39]-API.

The list of all queried WebGL extensions and parameters by FPJS are available as an attachment in Section 6.4.
FPJS then concatenates the result of the queries and creates a hash over the following categories of WebGL parameters:

```
contextAttributes:
"6b1ed336830d2bc96442a9d76373252a",
parameters: "57a2cddb99538d50a0138430ed0720c5",
parameters2: "7bd4d913de3e22461894a997d864dcb8",
shaderPrecisions:
"f223dfbcd580cf142da156d93790eb83",
extensions: "57233d7b10f89fcd1ff95e3837ccd72d",
extensionParameters:
"fa430f89faf2af23f701c2c6909bcaad",
extensionParameters2:
"86a8abb36f0cb30b5946dec0c761d042",
```

and extracts the following plaintext parameters:

```
version: "WebGL 1.0 (OpenGL ES 2.0 Chromium)",
vendor: "WebKit",
vendorUnmasked: "Google Inc. (Apple)",
renderer: "WebKit WebGL",
rendererUnmasked: "ANGLE (Apple, ANGLE Metal
Renderer: Apple M1 Ultra, Unspecified Version)",
shadingLanguageVersion: "WebGL GLSL ES 1.0 (OpenGL ES
GLSL ES 1.0 Chromium)",
```

2. **Algorithm**

After all browser parameters of the visiting user have been collected, this data must be evaluated.

1) **Goal**

The algorithm must decide whether the website visitor is a known visitor or a new visitor.

2) **Approach**

The collected parameters of the visiting user must be compared with the database of all previous visitors.
A probability must be calculated that indicates how closely the visiting user matches the existing user in the database.

Calculation

To calculate the probability, all parameters have to be compared with each entry in the database, to see if they match or not.

Naive

A naive approach to calculating the probability would be a value between 0 and 1, starting at 0 and increasing evenly for each matching parameter value in the database.

The issue with this approach is that all parameters influence the probability equally, which reduces the accuracy of the identification. A less stable parameter, such as screen resolution, could distort the identification if it is not weighted accordingly.

Weighting

[34] https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API

[35] https://developer.mozilla.org/en-US/docs/Web/API/WebGLRenderingContext/getSupportedExtensions

[36] https://developer.mozilla.org/en-US/docs/Web/API/WebGLRenderingContext/getContextAttributes

[37] https://developer.mozilla.org/en-US/docs/Web/API/WebGLRenderingContext/getParameter

[38] https://developer.mozilla.org/en-US/docs/Web/API/WebGLRenderingContext/getExtension

[39] https://developer.mozilla.org/en-US/docs/Web/API/WebGLRenderingContext

Weighting is the process of giving a parameter type an adjusted influence on the resulting probability.
Therefore, the parameters need to be weighted in a way that ensures that a small change in the browser configuration does not have a great impact on the probability.
This way, the user can continue to be reliably identified.

3) Formula

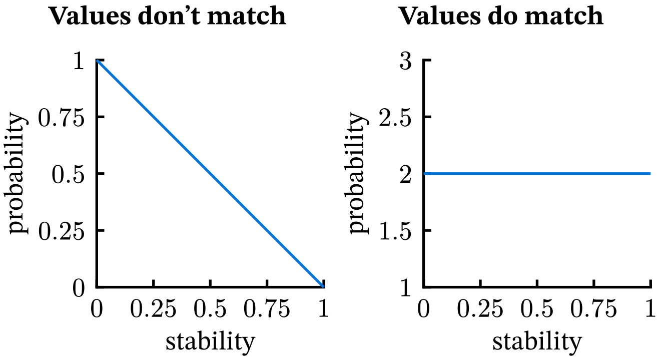
The implementation of parameter weights is done by evaluating the stability and uniqueness of each parameter.

$$x = \begin{cases} 0 \text{ parameter does not match with the compared value} \\ 1 \text{ parameter does match with the compared value} \end{cases}$$

1. **Stability**: Indicates how likely it is that a parameter value remains unchanged across multiple website visits from the same user (user session).

   The following stability function determines the probability of a fingerprint match based on the parameter stability.
   $\text{stability}(s, x) = (1 - s) \cdot (x + 1) + 2 \cdot s \cdot x$

   Where s is the stability of the parameter. A stable parameter that does not match (s close to 1) decreases the probability of a match, while an unstable parameter (s close to 0) has less influence. A stable parameter that does match increases the probability of a match and it is further influenced by the uniqueness of the parameter.


Values don't match

Values do match

Determining the stability for each parameter

First, the parameters of different users must be saved in a database. It must be noted that the user can be identified by a unique cookie parameter. After a certain time interval, all parameters must be read out from the user's browser and saved again. The data can then be analyzed by identifying and dividing the user entries from the database based on their unique cookie identifier into so-called "user sessions". This makes it possible to analyze how individual parameters of a user change over time and which ones remain the same.

To calculate the stability value for each parameter, the following formula must be calculated:

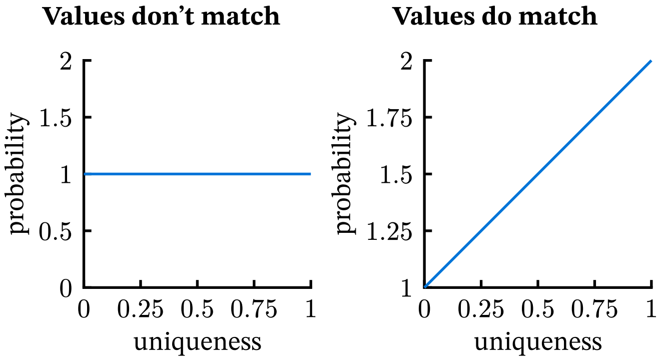
$\# = \text{count}$

$$\forall p \in \text{Parameter} \mid p_{\text{stability}} = \text{average user session stability} = \frac{\sum_{u \in \text{ user sessions}} (u_{\text{stability}})}{\text{session count}}$$

$$u_{\text{stability}} = \sum_{\text{v } \in \text{ user parameters p}} \left(\frac{\#\text{parameters with value v}}{\#\text{parameters for user}} \right)$$

2. **Uniqueness**: Measures how unique a parameter value is compared to other users.
   $\text{uniqueness}(u, x) = (1 - x) + (1 + u) \cdot x$

   Here u is the uniqueness of the parameter. A parameter that does match with high uniqueness value (u close to 1) increases the probability. The uniqueness of a parameter that does not match has no influence on the probability, as the stability is the deciding factor for that case.


Values don't match

Values do match

Determining the uniqueness of each parameter

The parameters of many different users must be saved in a database, possibly with a unique browser configuration. To calculate the uniqueness value for each parameter, the following formula must be calculated:

$\lambda = \#\text{all parameter values}$

$$\mu_v = \left(\frac{\#\text{parameters with value v}}{\lambda} \right)$$

$$\forall p \in P \mid p_{\text{uniqueness}} = - \sum_{\text{v } \in \text{ parameters}} \mu_v \cdot \log_\lambda(\mu_v)$$

3. **Algorithm to calculate the probability of a fingerprint match**:

The following compare function c is used to calculate the variable x. This function determines if a parameter value matches the compared value from the database.

$$c(a, b) := \begin{cases} 0 \text{ if a } \neq b \\ 1 \text{ otherwise} \end{cases}$$

The following formula combines stability and uniqueness into a parameter_weight function that indicates the influence of the parameter for identifying a user.

$$\text{parameter\_weight}(s, u, x) = \text{stability}(s, x) \cdot \text{uniqueness}(u, x)$$

The final fingerprint matching algorithm then multiplies each parameter_weight to a final probability value.

$$\text{match (db, user parameters)} = \prod_{\text{p} \in \text{parameters}} \text{parameter\_weight}\left(p_{\text{stability}}, p_{\text{uniqueness}}, c(p, \text{db}_p)\right)$$

3. Data analysis

We deployed our custom fingerprint library on our website https://fingerprint.samuelscheit.com/ and over the span of 17 days we collected fingerprints from 229 visits of 115 unique visitors. Each visitor was identified based on their unique cookie ID and IP address. On average each unique visitor visited the website 2 times, with a mean visit count of 1 and a standard deviation of 2.392.

Furthermore by analyzing the uniqueness values of the parameters that are available in Section 6.2 the following abnormal uniqueness values can be seen:

```
"navigator_webdriver": 0.005145600506297009,
```

The parameter `navigator_webdriver` indicates if the browser is being automated, e.g. by Chrome's dev tool protocol, which is used by web scrapers.

One might expect that only "human" visitors visited the website and clicked the consent button to start fingerprinting. However, since the value of `navigator_webdriver` was `true` instead of `null` for one data entry, this strongly suggests that an automated browser (web scraper) visited the website. By analyzing the visitor's IP address and user agent, it can be determined that the WebArchive[40] scraper also visited the website.

In addition, when analyzing the stability values in Section 6.1 the following abnormalities can be seen:

```
"browser_gecko": 0.9608695652173913,
"browser_webkit": 0.9956521739130435,
...
```

These parameters indicate if the browser uses the Gecko or WebKit browser engine. Normally these values should have the value `1` indicating that the browser type never changes for the same user. However by analyzing the dataset one can see that the same user has spoofed their user agent and browser type resulting in the slightly decreased stability for the browser engine parameters.

4. Conclusion

The analysis of the technical implementation of FPJS and its parameters revealed that the library uses a large set of 117 parameters to identify web browsers. As opposed to comparable algorithms proposed by academic works, FPJS checks for the existence of browser-specific global objects as well as individual fingerprints for the audio, WebGL and canvas APIs. These parameters are sent to the FPJS server which generates a corresponding unique fingerprint for that user agent. The discrepancies between the parameters used in research and their practical application can be explained by the timeliness and constant updating of the FPJS library and the lack of constant research in this area.

As posts about the dataset gathering were made in a forum about digital privacy, a greater than average number of spoofed parameter values in the dataset should be expected due to privacy enhancing technologies used by the visitors.

A total of 229 valid fingerprints were recorded over the course of 17 days using a custom fingerprinting library.

The size of this dataset was insufficient to provide statistically significant insights into the stability of the parameters used by FPJS.

A larger dataset of fingerprints over a longer time span is required to draw conclusions about the accuracy of FingerprintJS' claims of 99.5% fingerprint stability over 120 days.

Further research is needed to investigate the accuracy of the proposed algorithm. Conducting a study similar to [4] on two datasets, one to weigh parameters based on stability and uniqueness and another to determine the effectiveness of the algorithm may provide more conclusive findings.

5. References

[1] U. Iqbal, S. Englehardt, and Z. Shafiq, "Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors," in *2021 IEEE Symposium on Security and Privacy (SP)*, 2021, pp. 1143–1161. doi: 10.1109/SP40001.2021.00017.

[2] Ł. Olejnik, G. Acar, C. Castelluccia, and C. Diaz, "The Leaking Battery," in *Data Privacy Management, and Security Assurance*, J. Garcia-Alfaro, G. Navarro-Arribas, A. Aldini, F. Martinelli, and N. Suri, Eds., Cham: Springer International Publishing, 2016, pp. 254–263.

[3] P. Eckersley, "How Unique Is Your Web Browser?," in *Privacy Enhancing Technologies*, M. J. Atallah and N. J. Hopper, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–18.

[4] G. Pugliese, C. Riess, F. Gassmann, and Z. Benenson, "Long-Term Observation on Browser Fingerprinting: Users' Trackability and Perspective," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, pp. 558–577, 2020, doi: 10.2478/popets-2020-0041.

[40] https://archive.org/

[5] A. K. Junhua Su, “Automatic Discovery of Emerging Browser Fingerprinting Techniques.” [Online]. Available: https://www.kapravelos.com/publications/fptechniques-www23.pdf

[6] “@fingerprintjs/fingerprintjs vs @rajesh896/broprint.js vs @thumbmarkjs/thumbmarkjs vs clientjs vs imprintjs | npm trends.” Accessed: May 02, 2024. [Online]. Available: https://npmtrends.com/@fingerprintjs/fingerprintjs-vs-@rajesh896/broprint.js-vs-@thumbmarkjs/thumbmarkjs-vs-clientjs-vs-imprintjs

[7] “Window - Web APIs | MDN.” Accessed: Jul. 27, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Window

[8] [Online]. Available: https://github.com/aappleby/smhasher

[9] N. N. J. P. Konstantinos Solomos Panagiotis Ilia, “Escaping the Confines of Time: Continuous Browser Extension Fingerprinting Through Ephemeral Modifications.” [Online]. Available: https://www.cs.uic.edu/~polakis/papers/solomos-ccs22.pdf

[10] [Online]. Available: https://fingerprint.com/

[11] [Online]. Available: https://github.com/fingerprintjs/fingerprintjs

6. Attachments

1. Stability

```
  "tls_ja4": 0.8418840579710145,
"audio1": 0.9044927536231884,
"audio2": 0.8957971014492755,
"canvas_geometry": 0.91231884057971,
"canvas_text": 0.9101449275362318,
"canvas_winding": 1,
"webgl_geometry": 0.8967325428194993,
"webgl_version": 0.9579710144927537,
"webgl_vendor": 0.955072463768116,
"webgl_shading_language_version":
0.9579710144927537,
"webgl_vendor_unmasked": 0.9536231884057972,
"webgl_renderer": 0.9536231884057972,
"webgl_renderer_unmasked": 0.9536231884057972,
"webgl_context_attributes": 0.9572463768115943,
"webgl_shader_precisions": 0.9739130434782609,
"webgl_extensions": 0.9528985507246378,
"webgl_parameters1": 0.9579710144927537,
"webgl_parameters2": 0.9471014492753623,
"webgl_extensions_parameters1": 0.9528985507246378,
"webgl_extensions_parameters2": 0.9528985507246378,
"http_headers": 0.9202898550724637,
"user_agent": 0.922463768115942,
"math": 0.9579710144927537,
"buffer": 1,
"navigator_pdfViewerEnabled": 0.9869565217391304,
"navigator_language": 0.9739130434782609,
"navigator_languages": 0.9384057971014493,
"navigator_webdriver": 1,
"navigator_userAgentData": 0.9543478260869566,
"navigator_appVersion": 0.9246376811594202,
"navigator_connection_rtt": 0.9681159420289855,
"navigator_plugins": 0.9428985507246377,
"navigator_hardwareConcurrency": 0.9478260869565217,
"navigator_deviceMemory": 0.9481159420289855,
"navigator_platform": 0.9855072463768115,
"navigator_vendor": 0.955072463768116,
"navigator_productSub": 0.9565217391304348,
"navigator_vendorSub": 1,
"navigator_onLine": 1,
"navigator_media_devices": 0.8825465838509315,
"navigator_getHighEntropyValues":
0.9543478260869566,
"navigator_doNotTrack": 0.963768115942029,
"navigator_oscpu": 0.95,
"navigator_maxTouchPoints": 0.991304347826087,
"navigator_prototype": 0.9347826086956522,
"window_TouchEvent": 0.9956521739130435,
"window_ontouchstart": 0.9956521739130435,
"storage_estimate": 0.7773913043478261,
"storage_getDirectory": 0.9956521739130435,
"speechSynthesis_voices": 0.9376811594202898,
"webrtc_candidates": 0.8380400276052449,
"webrtc_stats": 0.8267893899840969,
"apple_pay": 0.9956521739130435,
"screen_safeArea": 0.9565217391304348,
```

"screen_width": 0.938695652173913,
"screen_height": 0.9242028985507246,
"screen_colorDepth": 0.991304347826087,
"screen_outerWidth": 0.941304347826087,
"screen_outerHeight": 0.8969565217391303,
"screen_innerWidth": 0.9391304347826087,
"screen_innerHeight": 0.9002898550724637,
"screen_highRes": 0.9869565217391304,
"window_devicePixelRatio": 0.9710144927536233,
"dom_blocker": 0.9782608695652174,
"font_list": 0.9543478260869566,
"font_widths": 1,
"font_emoji": 0.8630434782608696,
"font_math": 0.8586956521739131,
"location_href": 1,
"document_referrer": 0.9159420289855073,
"window_webkitRequestFileSystem":
0.9608695652173913,
"window_openDatabase": 0.9956521739130435,
"window_sessionStorage": 1,
"window_localStorage": 1,
"window_indexedDB": 1,
"window_permissions": 1,
"window_process": 1,
"window_globals": 0.9405797101449276,
"document_cookie": 0.8492753623188405,
"cookies_enabled": 1,
"webassembly_validate": 0.9826086956521739,
"media_dark_mode": 0.9739130434782609,
"media_inverted_colors": 1,
"media_forced_colors": 1,
"media_max_monochrome": 1,
"media_contrast": 1,
"media_reduced_motion": 1,
"media_dynamic_range": 0.9869565217391304,
"media_transparency": 1,
"media_backdrop_blur": 0.9956521739130435,
"time_zone_offset": 0.9695652173913043,
"performance_now": 0.9434782608695652,
"performance_memory": 0.9594202898550726,
"browser_chromium": 0.9608695652173913,
"browser_chromium86OrNewer": 1,
"browser_trident": 1,
"browser_gecko": 0.9608695652173913,
"browser_webkit": 0.9956521739130435,
"browser_ipad": 0.9608695652173913,
"browser_android": 1,
"browser_webkit606OrNewer": 1,
"browser_webkit616OrNewer": 0.9608695652173913,
"browser_safari_webkit": 0.9782608695652174,
"browser_webkit_desktop": 0.9739130434782609,
"browser_edgeHTML": 1,
"a_attributionSourceId": 1,
"a_attributeNames": 1,
"eval_toString": 0.9608695652173913,
"drm": 0.9492753623188405,
"error_stack": 0.9594202898550726,
"error_toSource": 1,
"error_undefined": 1,
"external_toString": 0.9956521739130435,
"window_close_toString": 0.9608695652173913,
"function_bind_toString": 0.9608695652173913,
"keyboard_layout": 0.9695652173913043

2. Uniqueness

  "tls_ja4": 0.558658346053923,
"audio1": 0.5536408036909787,
"audio2": 0.5597586878789822,
"canvas_geometry": 0.567201332727466,
"canvas_text": 0.5911014186366464,
"canvas_winding": 0,
"webgl_geometry": 0.5929631750497047,
"webgl_version": 0.19025775379537763,
"webgl_vendor": 0.169998758900446,
"webgl_shading_language_version":
0.24239469535254213,
"webgl_vendor_unmasked": 0.3586756910748313,
"webgl_renderer": 0.2504210831624503,
"webgl_renderer_unmasked": 0.42137563566439074,
"webgl_context_attributes": 0.2338376361058555,
"webgl_shader_precisions": 0.24730203521117708,
"webgl_extensions": 0.4951542027154874,
"webgl_parameters1": 0.19018199767319904,
"webgl_parameters2": 0.5319140808245012,
"webgl_extensions_parameters1": 0.4951542027154874,
"webgl_extensions_parameters2": 0.4951542027154874,
"http_headers": 0.43981488623946885,
"user_agent": 0.6143899539626935,
"math": 0.2254932933107505,
"buffer": 0.12024776284680094,
"navigator_pdfViewerEnabled": 0.09089894245072039,
"navigator_language": 0.28264574382627755,
"navigator_languages": 0.43938860386786216,
"navigator_webdriver": 0.005145600506297009,
"navigator_userAgentData": 0.30044966195931827,
"navigator_appVersion": 0.5386222800956254,
"navigator_connection_rtt": 0.11621433806482255,
"navigator_plugins": 0.26183111798311565,
"navigator_hardwareConcurrency": 0.3718775259784102,
"navigator_deviceMemory": 0.21457322090589404,
"navigator_platform": 0.32724574689293334,
"navigator_vendor": 0.1968445648757399,
"navigator_productSub": 0.11930351928958445,
"navigator_vendorSub": 0,
"navigator_onLine": 0,
"navigator_media_devices": 0.5196661102782442,
"navigator_getHighEntropyValues":
0.35736828432236006,
"navigator_doNotTrack": 0.1715794662121956,
"navigator_oscpu": 0.21631864261631562,
"navigator_maxTouchPoints": 0.14017459088308656,
"navigator_prototype": 0.47216321610759127,
"window_TouchEvent": 0.12743395320937465,
"window_ontouchstart": 0.12743395320937465,
"storage_estimate": 0.6862541822097243,
"storage_getDirectory": 0.005145600506297009,
"speechSynthesis_voices": 0.26798945330952995,
"webrtc_candidates": 0.6813077547097491,
"webrtc_stats": 0.7283061767429343,
"apple_pay": 0.10322943378082484,

```
"screen_safeArea": 0.2661681551109322,
"screen_width": 0.5907721897190661,
"screen_height": 0.6592835792225026,
"screen_colorDepth": 0.052425666393813374,
"screen_outerWidth": 0.6542369236315339,
"screen_outerHeight": 0.7681318453450321,
"screen_innerWidth": 0.6337798391096926,
"screen_innerHeight": 0.7644282244187757,
"screen_highRes": 0.12494043720753048,
"window_devicePixelRatio": 0.31016654685445166,
"dom_blocker": 0.23908706392983922,
"font_list": 0.4463400108628246,
"font_widths": 0,
"font_emoji": 0.8392869503455953,
"font_math": 0.8371809112265248,
"location_href": 0,
"document_referrer": 0.22047091991467876,
"window_webkitRequestFileSystem":
0.12342764907489093,
"window_openDatabase": 0.027765821406086657,
"window_sessionStorage": 0,
"window_localStorage": 0,
"window_indexedDB": 0,
"window_permissions": 0,
"window_process": 0,
"window_globals": 0.17771721380932312,
"document_cookie": 0.2414423372854151,
"cookies_enabled": 0,
"webassembly_validate": 0.14830216984893713,
"media_dark_mode": 0.12272112592574969,
"media_inverted_colors": 0,
"media_forced_colors": 0,
"media_max_monochrome": 0.009179346505798587,
"media_contrast": 0.027765821406086657,
"media_reduced_motion": 0.032887514608951327,
"media_dynamic_range": 0.11978321736543193,
"media_transparency": 0,
"media_backdrop_blur": 0.08859371946450163,
"time_zone_offset": 0.30998753207516005,
"performance_now": 0.23630876622094138,
"performance_memory": 0.2271445824933998,
"browser_chromium": 0.12466665111069306,
"browser_chromium86OrNewer": 0,
"browser_trident": 0,
"browser_gecko": 0.11978321736543193,
"browser_webkit": 0.09926320122747831,
"browser_ipad": 0.1273991859899874,
"browser_android": 0.10796313962517398,
"browser_webkit606OrNewer": 0,
"browser_webkit616OrNewer": 0.12494043720753048,
"browser_safari_webkit": 0.0811148537710179,
"browser_webkit_desktop": 0.1002175442839273,
"browser_edgeHTML": 0,
"a_attributionSourceId": 0,
"a_attributeNames": 0,
"eval_toString": 0.12466665111069306,
"drm": 0.21257774657617215,
"error_stack": 0.19697130793963233,
"error_toSource": 0,
"error_undefined": 0,
"external_toString": 0.09926320122747831,
"window_close_toString": 0.12466665111069306,
"function_bind_toString": 0.12466665111069306,
"keyboard_layout": 0.18513415894394583
```

3. Parameters

```
  Math
Array.isArray
ArrayBuffer
atob
Blob
Boolean
btoa
CompressionStream
crypto
crypto.getRandomValues
CSS
CSS.supports
Date.getTimezoneOffset
Date.now
Date.toISOString
Date.toUTCString
document.cookie
document.createElement().attributionsourceid
document.createElement().attributionSourceId
window.console.debug
window.devicePixelRatio
window.document
window.document.body
window.document.body.append
window.document.body.appendChild
window.document.body.style
window.document.createElement().getBoundingClientRect().width
window.document.createElement().offsetHeight
window.document.createElement().offsetWidth
window.document.createElement().style
window.document.readyState
window.Error
window.getComputedStyle
window.getComputedStyle().color
window.getComputedStyle().getPropertyValue
window.navigator
window.navigator.webdriver
window.screen
window.screen.height
window.screen.width
window.setTimeout
canvasContext.ACTIVE_ATTRIBUTES
canvasContext.ACTIVE_TEXTURE
canvasContext.ACTIVE_UNIFORMS
canvasContext.ALIASED_LINE_WIDTH_RANGE
canvasContext.ALIASED_POINT_SIZE_RANGE
canvasContext.ALPHA
canvasContext.ALPHA_BITS
canvasContext.ALWAYS
canvasContext.arc
canvasContext.ARRAY_BUFFER
canvasContext.ARRAY_BUFFER_BINDING
canvasContext.ATTACHED_SHADERS
canvasContext.attachShader
canvasContext.BACK
canvasContext.beginPath
canvasContext.bindBuffer
canvasContext.BLEND
canvasContext.BLEND_COLOR
canvasContext.BLEND_DST_ALPHA
canvasContext.BLEND_DST_RGB
canvasContext.BLEND_EQUATION
canvasContext.BLEND_EQUATION_ALPHA
canvasContext.BLEND_EQUATION_RGB
canvasContext.BLEND_SRC_ALPHA
canvasContext.BLEND_SRC_RGB
canvasContext.BLUE_BITS
canvasContext.BOOL
canvasContext.BOOL_VEC2
canvasContext.BOOL_VEC3
canvasContext.BOOL_VEC4
canvasContext.BROWSER_DEFAULT_WEBGL
canvasContext.BUFFER_SIZE
canvasContext.BUFFER_USAGE
canvasContext.bufferData
canvasContext.BYTE
canvasContext.canvas
canvasContext.canvas.toDataURL
canvasContext.CCW
canvasContext.CLAMP_TO_EDGE
canvasContext.clear
canvasContext.clearColor
canvasContext.closePath
canvasContext.COLOR_ATTACHMENT0
canvasContext.COLOR_BUFFER_BIT
canvasContext.COLOR_CLEAR_VALUE
canvasContext.COLOR_WRITEMASK
canvasContext.COMPILE_STATUS
canvasContext.compileShader
canvasContext.COMPRESSED_TEXTURE_FORMATS
canvasContext.CONSTANT_ALPHA
canvasContext.CONSTANT_COLOR
canvasContext.CONTEXT_LOST_WEBGL
canvasContext.createBuffer
canvasContext.createProgram
canvasContext.createShader
canvasContext.CULL_FACE
canvasContext.CULL_FACE_MODE
canvasContext.CURRENT_PROGRAM
canvasContext.CURRENT_VERTEX_ATTRIB
canvasContext.CW
canvasContext.DECR
canvasContext.DECR_WRAP
canvasContext.DELETE_STATUS
canvasContext.DEPTH_ATTACHMENT
canvasContext.DEPTH_BITS
canvasContext.DEPTH_BUFFER_BIT
canvasContext.DEPTH_CLEAR_VALUE
canvasContext.DEPTH_COMPONENT
canvasContext.DEPTH_COMPONENT16
canvasContext.DEPTH_FUNC
canvasContext.DEPTH_RANGE
canvasContext.DEPTH_STENCIL
canvasContext.DEPTH_STENCIL_ATTACHMENT
canvasContext.DEPTH_TEST
canvasContext.DEPTH_WRITEMASK
canvasContext.DITHER
canvasContext.DONT_CARE
canvasContext.drawArrays
canvasContext.DST_ALPHA
canvasContext.DST_COLOR
canvasContext.DYNAMIC_DRAW
canvasContext.ELEMENT_ARRAY_BUFFER
canvasContext.ELEMENT_ARRAY_BUFFER_BINDING
canvasContext.enableVertexAttribArray
canvasContext.EQUAL
canvasContext.FASTEST
canvasContext.fill
canvasContext.fillRect
canvasContext.fillText
canvasContext.FLOAT
canvasContext.FLOAT_MAT2
canvasContext.FLOAT_MAT3
canvasContext.FLOAT_MAT4
canvasContext.FLOAT_VEC2
canvasContext.FLOAT_VEC3
canvasContext.FLOAT_VEC4
canvasContext.FRAGMENT_SHADER
canvasContext.FRAMEBUFFER
canvasContext.FRAMEBUFFER_ATTACHMENT_OBJECT_NAME
canvasContext.FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE
canvasContext.FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE
canvasContext.FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL
canvasContext.FRAMEBUFFER_BINDING
canvasContext.FRAMEBUFFER_COMPLETE
canvasContext.FRAMEBUFFER_INCOMPLETE_ATTACHMENT
canvasContext.FRAMEBUFFER_INCOMPLETE_DIMENSIONS
canvasContext.FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT
canvasContext.FRAMEBUFFER_UNSUPPORTED
canvasContext.FRONT
canvasContext.FRONT_AND_BACK
canvasContext.FRONT_FACE
canvasContext.FUNC_ADD
canvasContext.FUNC_REVERSE_SUBTRACT
canvasContext.FUNC_SUBTRACT
canvasContext.GENERATE_MIPMAP_HINT
canvasContext.GEQUAL
canvasContext.getContextAttributes()
canvasContext.getExtension
canvasContext.getParameter
canvasContext.getShaderPrecisionFormat
canvasContext.getShaderPrecisionFormat().precision
canvasContext.getShaderPrecisionFormat().rangeMax
canvasContext.getShaderPrecisionFormat().rangeMin
canvasContext.getSupportedExtensions
canvasContext.getUniformLocation
canvasContext.GREATER
canvasContext.GREEN_BITS
canvasContext.HIGH_FLOAT
canvasContext.HIGH_INT
canvasContext.IMPLEMENTATION_COLOR_READ_FORMAT
canvasContext.IMPLEMENTATION_COLOR_READ_TYPE
canvasContext.INCR
canvasContext.INCR_WRAP
canvasContext.INT
canvasContext.INT_VEC2
canvasContext.INT_VEC3
canvasContext.INT_VEC4
canvasContext.INVALID_ENUM
canvasContext.INVALID_FRAMEBUFFER_OPERATION
canvasContext.INVALID_OPERATION
canvasContext.INVALID_VALUE
canvasContext.INVERT
canvasContext.isPointInPath
canvasContext.KEEP
canvasContext.LEQUAL
canvasContext.LESS
canvasContext.LINE_LOOP
canvasContext.LINE_STRIP
canvasContext.LINE_WIDTH
canvasContext.LINEAR
canvasContext.LINEAR_MIPMAP_LINEAR
canvasContext.LINEAR_MIPMAP_NEAREST
canvasContext.LINES
canvasContext.LINK_STATUS
canvasContext.linkProgram
canvasContext.LOW_FLOAT
canvasContext.LOW_INT
canvasContext.LUMINANCE
canvasContext.LUMINANCE_ALPHA
canvasContext.MAX_COMBINED_TEXTURE_IMAGE_UNITS
canvasContext.MAX_CUBE_MAP_TEXTURE_SIZE
canvasContext.MAX_FRAGMENT_UNIFORM_VECTORS
canvasContext.MAX_RENDERBUFFER_SIZE
canvasContext.MAX_TEXTURE_IMAGE_UNITS
canvasContext.MAX_TEXTURE_SIZE
canvasContext.MAX_VARYING_VECTORS
canvasContext.MAX_VERTEX_ATTRIBS
canvasContext.MAX_VERTEX_TEXTURE_IMAGE_UNITS
canvasContext.MAX_VERTEX_UNIFORM_VECTORS
canvasContext.MAX_VIEWPORT_DIMS
canvasContext.MEDIUM_FLOAT
canvasContext.MEDIUM_INT
canvasContext.MIRRORED_REPEAT
canvasContext.NEAREST
canvasContext.NEAREST_MIPMAP_LINEAR
canvasContext.NEAREST_MIPMAP_NEAREST
canvasContext.NEVER
canvasContext.NICEST
canvasContext.NO_ERROR
canvasContext.NONE
canvasContext.NOTEQUAL
canvasContext.ONE
canvasContext.ONE_MINUS_CONSTANT_ALPHA
canvasContext.ONE_MINUS_CONSTANT_COLOR
canvasContext.ONE_MINUS_DST_ALPHA
canvasContext.ONE_MINUS_DST_COLOR
canvasContext.ONE_MINUS_SRC_ALPHA
canvasContext.ONE_MINUS_SRC_COLOR
canvasContext.OUT_OF_MEMORY
canvasContext.PACK_ALIGNMENT
canvasContext.POINTS
canvasContext.POLYGON_OFFSET_FACTOR
canvasContext.POLYGON_OFFSET_FILL
canvasContext.POLYGON_OFFSET_UNITS
canvasContext.rect
canvasContext.RED_BITS
canvasContext.RENDERBUFFER
canvasContext.RENDERBUFFER_ALPHA_SIZE
canvasContext.RENDERBUFFER_BINDING
canvasContext.RENDERBUFFER_BLUE_SIZE
canvasContext.RENDERBUFFER_DEPTH_SIZE
canvasContext.RENDERBUFFER_GREEN_SIZE
canvasContext.RENDERBUFFER_HEIGHT
canvasContext.RENDERBUFFER_INTERNAL_FORMAT
canvasContext.RENDERBUFFER_RED_SIZE
canvasContext.RENDERBUFFER_STENCIL_SIZE
canvasContext.RENDERBUFFER_WIDTH
canvasContext.RENDERER
canvasContext.REPEAT
canvasContext.REPLACE
canvasContext.RGB
canvasContext.RGB5_A1
canvasContext.RGB565
canvasContext.RGB8
canvasContext.RGBA
canvasContext.RGBA4
canvasContext.RGBA8
canvasContext.SAMPLE_ALPHA_TO_COVERAGE
canvasContext.SAMPLE_BUFFERS
canvasContext.SAMPLE_COVERAGE
canvasContext.SAMPLE_COVERAGE_INVERT
canvasContext.SAMPLE_COVERAGE_VALUE
canvasContext.SAMPLER_2D
canvasContext.SAMPLER_CUBE
canvasContext.SAMPLES
canvasContext.SCISSOR_BOX
canvasContext.SCISSOR_TEST
canvasContext.SHADER_TYPE
canvasContext.shaderSource
canvasContext.SHADING_LANGUAGE_VERSION
canvasContext.SHORT
canvasContext.SRC_ALPHA
canvasContext.SRC_ALPHA_SATURATE
canvasContext.SRC_COLOR
canvasContext.STATIC_DRAW
canvasContext.STENCIL_ATTACHMENT
canvasContext.STENCIL_BACK_FAIL
canvasContext.STENCIL_BACK_FUNC
canvasContext.STENCIL_BACK_PASS_DEPTH_FAIL
canvasContext.STENCIL_BACK_PASS_DEPTH_PASS
canvasContext.STENCIL_BACK_REF
canvasContext.STENCIL_BACK_VALUE_MASK
canvasContext.STENCIL_BACK_WRITEMASK
canvasContext.STENCIL_BITS
canvasContext.STENCIL_BUFFER_BIT
canvasContext.STENCIL_CLEAR_VALUE
canvasContext.STENCIL_FAIL
canvasContext.STENCIL_FUNC
canvasContext.STENCIL_INDEX8
canvasContext.STENCIL_PASS_DEPTH_FAIL
canvasContext.STENCIL_PASS_DEPTH_PASS
canvasContext.STENCIL_REF
canvasContext.STENCIL_TEST
canvasContext.STENCIL_VALUE_MASK
canvasContext.STENCIL_WRITEMASK
canvasContext.STREAM_DRAW
canvasContext.SUBPIXEL_BITS
canvasContext.TEXTURE
canvasContext.TEXTURE_2D
canvasContext.TEXTURE_BINDING_2D
canvasContext.TEXTURE_BINDING_CUBE_MAP
canvasContext.TEXTURE_CUBE_MAP
canvasContext.TEXTURE_CUBE_MAP_NEGATIVE_X
canvasContext.TEXTURE_CUBE_MAP_NEGATIVE_Y
canvasContext.TEXTURE_CUBE_MAP_NEGATIVE_Z
canvasContext.TEXTURE_CUBE_MAP_POSITIVE_X
canvasContext.TEXTURE_CUBE_MAP_POSITIVE_Y
canvasContext.TEXTURE_CUBE_MAP_POSITIVE_Z
canvasContext.TEXTURE_MAG_FILTER
canvasContext.TEXTURE_MIN_FILTER
canvasContext.TEXTURE_WRAP_S
canvasContext.TEXTURE_WRAP_T
canvasContext.TEXTURE0
canvasContext.TEXTURE1
canvasContext.TEXTURE10
canvasContext.TEXTURE11
canvasContext.TEXTURE12
canvasContext.TEXTURE13
canvasContext.TEXTURE14
canvasContext.TEXTURE15
canvasContext.TEXTURE16
canvasContext.TEXTURE17
canvasContext.TEXTURE18
canvasContext.TEXTURE19
canvasContext.TEXTURE2
canvasContext.TEXTURE20
canvasContext.TEXTURE21
canvasContext.TEXTURE22
canvasContext.TEXTURE23
canvasContext.TEXTURE24
canvasContext.TEXTURE25
canvasContext.TEXTURE26
canvasContext.TEXTURE27
canvasContext.TEXTURE28
canvasContext.TEXTURE29
canvasContext.TEXTURE3
canvasContext.TEXTURE30
canvasContext.TEXTURE31
canvasContext.TEXTURE4
canvasContext.TEXTURE5
canvasContext.TEXTURE6
canvasContext.TEXTURE7
canvasContext.TEXTURE8
canvasContext.TEXTURE9
canvasContext.TRIANGLE_FAN
canvasContext.TRIANGLE_STRIP
canvasContext.TRIANGLES
canvasContext.uniform1f
canvasContext.UNPACK_ALIGNMENT
canvasContext.UNPACK_COLORSPACE_CONVERSION_WEBGL
canvasContext.UNPACK_FLIP_Y_WEBGL
canvasContext.UNPACK_PREMULTIPLY_ALPHA_WEBGL
canvasContext.UNSIGNED_BYTE
canvasContext.UNSIGNED_INT
canvasContext.UNSIGNED_SHORT
canvasContext.UNSIGNED_SHORT_4_4_4_4
canvasContext.UNSIGNED_SHORT_5_5_5_1
canvasContext.UNSIGNED_SHORT_5_6_5
canvasContext.useProgram
canvasContext.VALIDATE_STATUS
canvasContext.VENDOR
canvasContext.VERSION
canvasContext.VERTEX_ATTRIB_ARRAY_BUFFER_BINDING
canvasContext.VERTEX_ATTRIB_ARRAY_ENABLED
canvasContext.VERTEX_ATTRIB_ARRAY_NORMALIZED
canvasContext.VERTEX_ATTRIB_ARRAY_POINTER
canvasContext.VERTEX_ATTRIB_ARRAY_SIZE
canvasContext.VERTEX_ATTRIB_ARRAY_STRIDE
canvasContext.VERTEX_ATTRIB_ARRAY_TYPE
canvasContext.VERTEX_SHADER
canvasContext.vertexAttribPointer
canvasContext.VIEWPORT
canvasContext.ZERO
document.createElement().offsetHeight
document.createElement().parentNode
document.createElement().parentNode.removeChild
document.createElement().style
document.createElement().style.setProperty
document.createElement().toDataURL
document.createEvent
document.documentElement
document.documentElement.getAttributeNames
document.documentElement.style
document.hidden
document.referrer
document.removeEventListener
encodeURIComponent
Error
Error.call
Error.fileName
Error.prototype
Error.sourceURL
Error.stack
Error().state
eval
eval.toString
Float32Array
Float32Array.0
Float32Array.buffer
Float32Array.prototype
Function
```

```
Function.prototype
globalThis
Image
Image.prototype
Image.style
isNaN
localStorage
localStorage.getItem
localStorage.getItem.call
localStorage.removeItem
localStorage.removeItem.call
localStorage.setItem
localStorage.setItem.call
location
location.hash
location.hostname
location.href
Map
Map.delete
Map.get
Map.prototype
Map.set
matchMedia
matchMedia().matches
Math.abs
Math.acos
Math.acosh
Math.asin
Math.asinh
Math.atan
Math.atanh
Math.cos
Math.cosh
Math.exp
Math.expm1
Math.floor
Math.log
Math.log1p
Math.max
Math.min
Math.PI
Math.pow
Math.random
Math.round
Math.sin
Math.sinh
Math.sqrt
Math.tan
Math.tanh
MimeTypeArray
MimeTypeArray.prototype
navigator
navigator.appVersion
navigator.connection
navigator.connection.rtt
navigator.cpuClass
navigator.deviceMemory
navigator.doNotTrack
navigator.hardwareConcurrency
navigator.language
navigator.languages
navigator.languages.0
navigator.languages.1
navigator.languages.length
navigator.languages.toJSON
navigator.maxTouchPoints
navigator.mediaDevices
navigator.mediaDevices.enumerateDevices
navigator.mimeTypes
navigator.mimeTypes.length
navigator.onLine
navigator.oscpu
navigator.pdfViewerEnabled
navigator.permissions
navigator.permissions.query
navigator.platform
navigator.plugins
navigator.productSub
navigator.storage
navigator.storage.getDirectory
navigator.userAgent
navigator.userAgentData
```

```
navigator.vendor
navigator.webdriver
navigator.webkitTemporaryStorage
navigator.webkitTemporaryStorage.queryUsageAndQuota
Number
parseFloat
parseInt
performance
performance.getEntriesByName
performance.now
performance.timeOrigin
PluginArray
PluginArray.prototype
Promise
Promise.all
Promise.prototype
Promise.race
Promise.resolve
RegExp.global
RegExp.unicode
RegExp.unicodeSets
screen
screen.availHeight
screen.availLeft
screen.availTop
screen.availWidth
screen.height
screen.width
Set
Set.forEach
Set.has
Set.prototype
Set.size
setTimeout
SourceBuffer
SourceBufferList
String
String.fromCharCode
String.fromCodePoint
Symbol
Symbol.iterator
TextDecoder
TextEncoder
TextEncoder.encode
TextEncoder.encode().buffer
TextEncoder.encode().byteLength
TextEncoder.encode().byteOffset
TextEncoder.encode().length
TextEncoder.prototype
Uint32Array
Uint8Array
URL
URL.prototype
URL.toString
window
window.__crWeb
window.__edgeTrackingPreventionStatistics
window.__firefox__
window.__fpjs_pvid
window.__gCrWeb
window.__yb
window.__ybro
window.ApplePaySession
window.Audio
window.Audio.prototype
window.chrome
window.clearInterval
window.close
window.close.toString
window.devicePixelRatio
window.document
window.Document
window.Document.prototype
window.document.referrer
window.external
window.external.toString
window.Function
window.Function.prototype
window.HTMLElement
window.HTMLElement.prototype
window.indexedDB
window.innerHeight
window.innerWidth
window.Intl
window.Intl.DateTimeFormat
window.Intl.DateTimeFormat.prototype
window.Intl.DateTimeFormat.resolvedOptions
window.Intl.DateTimeFormat.resolvedOptions().timeZone
window.Intl.toString
window.Intl.valueOf
window.localStorage
window.location
window.location.ancestorOrigins
window.location.ancestorOrigins.length
window.location.href
window.location.origin
window.matchMedia
window.matchMedia().matches
window.navigator
window.navigator.adAuctionComponents
window.navigator.adAuctionComponents.name
window.navigator.appCodeName
window.navigator.appName
window.navigator.appVersion
window.navigator.bluetooth
window.navigator.canLoadAdAuctionFencedFrame
window.navigator.canLoadAdAuctionFencedFrame.name
window.navigator.clearAppBadge
window.navigator.clearAppBadge.name
window.navigator.clearOriginJoinedAdInterestGroups
window.navigator.clearOriginJoinedAdInterestGroups.name
window.navigator.clipboard
window.navigator.connection
window.navigator.constructor
window.navigator.constructor.name
window.navigator.cookieEnabled
window.navigator.createAuctionNonce
window.navigator.createAuctionNonce.name
window.navigator.credentials
window.navigator.deprecatedReplaceInURN
window.navigator.deprecatedReplaceInURN.name
window.navigator.deprecatedRunAdAuctionEnforcesKAnonymity
window.navigator.deprecatedURNToURL
window.navigator.deprecatedURNToURL.name
window.navigator.deviceMemory
window.navigator.doNotTrack
window.navigator.geolocation
window.navigator.getBattery
window.navigator.getBattery.name
window.navigator.getGamepads
window.navigator.getGamepads.name
window.navigator.getInstalledRelatedApps
window.navigator.getInstalledRelatedApps.name
window.navigator.getUserMedia
window.navigator.getUserMedia.name
window.navigator.gpu
window.navigator.hardwareConcurrency
window.navigator.hid
window.navigator.ink
window.navigator.javaEnabled
window.navigator.javaEnabled.name
window.navigator.joinAdInterestGroup
window.navigator.joinAdInterestGroup.name
window.navigator.keyboard
window.navigator.language
window.navigator.languages
window.navigator.leaveAdInterestGroup
window.navigator.leaveAdInterestGroup.name
window.navigator.locks
window.navigator.login
window.navigator.managed
window.navigator.maxTouchPoints
window.navigator.mediaCapabilities
window.navigator.mediaDevices
window.navigator.mediaSession
window.navigator.mimeTypes
window.navigator.onLine
window.navigator.pdfViewerEnabled
window.navigator.permissions
window.navigator.platform
window.navigator.plugins
window.navigator.presentation
window.navigator.product
window.navigator.productSub
window.navigator.registerProtocolHandler
window.navigator.registerProtocolHandler.name
window.navigator.requestMediaKeySystemAccess
window.navigator.requestMediaKeySystemAccess.name
window.navigator.requestMIDIAccess
window.navigator.requestMIDIAccess.name
window.navigator.runAdAuction
window.navigator.runAdAuction.name
window.navigator.scheduling
window.navigator.sendBeacon
window.navigator.sendBeacon.name
window.navigator.serial
window.navigator.serviceWorker
window.navigator.setAppBadge
window.navigator.setAppBadge.name
window.navigator.storage
window.navigator.storageBuckets
window.navigator.unregisterProtocolHandler
window.navigator.unregisterProtocolHandler.name
window.navigator.updateAdInterestGroups
window.navigator.updateAdInterestGroups.name
window.navigator.usb
window.navigator.userActivation
window.navigator.userAgent
window.navigator.userAgentData
window.navigator.vendor
window.navigator.vendorSub
window.navigator.vibrate
window.navigator.vibrate.name
window.navigator.virtualKeyboard
window.navigator.wakeLock
window.navigator.webdriver
window.navigator.webkitGetUserMedia
window.navigator.webkitGetUserMedia.name
window.navigator.webkitTemporaryStorage
window.navigator.windowControlsOverlay
window.navigator.xr
window.Notification
window.Notification.permission
window.OfflineAudioContext
window.OfflineAudioContext.createBiquadFilter
window.OfflineAudioContext.createBiquadFilter().connect
window.OfflineAudioContext.createBiquadFilter().frequency
window.OfflineAudioContext.createBiquadFilter().Q
window.OfflineAudioContext.createBufferSource
window.OfflineAudioContext.createBufferSource().connect
window.OfflineAudioContext.createBufferSource().start
window.OfflineAudioContext.createDynamicsCompressor
window.OfflineAudioContext.createDynamicsCompressor().attack
window.OfflineAudioContext.createDynamicsCompressor().connect
window.OfflineAudioContext.createDynamicsCompressor().knee
window.OfflineAudioContext.createDynamicsCompressor().ratio
window.OfflineAudioContext.createDynamicsCompressor().release
window.OfflineAudioContext.createDynamicsCompressor().threshold
window.OfflineAudioContext.createOscillator
window.OfflineAudioContext.createOscillator().connect
window.OfflineAudioContext.createOscillator().frequency
window.OfflineAudioContext.createOscillator().start
window.OfflineAudioContext.destination
window.OfflineAudioContext.prototype
window.OfflineAudioContext.startRendering
window.OfflineAudioContext.state
window.openDatabase
window.oprt
window.origin
window.outerHeight
window.outerWidth
window.parent
window.performance
window.performance.memory
window.performance.memory.jsHeapSizeLimit
window.performance.now
window.process
window.puffinDevice
window.Reflect
window.Reflect.toString
window.Reflect.valueOf
window.RTCPeerConnection
window.RTCPeerConnection.close
window.RTCPeerConnection.createDataChannel
window.RTCPeerConnection.createOffer
window.RTCPeerConnection.getStats
window.RTCPeerConnection.iceGatheringState
window.RTCPeerConnection.localDescription
window.RTCPeerConnection.localDescription.sdp
window.RTCPeerConnection.prototype
window.RTCPeerConnection.setLocalDescription
window.RTCPeerConnection.setRemoteDescription
window.RTCRtpSender
window.RTCRtpSender.getCapabilities
window.RTCSessionDescription
window.safari
window.samsungAr
window.screen
window.screen.colorDepth
window.sessionStorage
window.setInterval
window.SharedArrayBuffer
window.speechSynthesis
window.speechSynthesis.addEventListener
window.speechSynthesis.getVoices
window.speechSynthesis.removeEventListener
window.UCShellJava
window.ucweb
window.URL
window.URL.createObjectURL
window.URL.revokeObjectURL
window.WebAssembly
window.WebAssembly.validate
window.webkit
window.yandex
XMLHttpRequest
XMLHttpRequest.getAllResponseHeaders
XMLHttpRequest.open
XMLHttpRequest.prototype
XMLHttpRequest.response
XMLHttpRequest.send
XMLHttpRequest.setRequestHeader
XMLHttpRequest.status
XMLHttpRequest.statusText
```

4. WebGL Attributes

```
  contextAttributes: [
alpha=true
antialias=true
depth=true
desynchronized=false
failIfMajorPerformanceCaveat=false
powerPreference=default
premultipliedAlpha=true
preserveDrawingBuffer=false
stencil=false
xrCompatible=false
]
parameters: [
ACTIVE_ATTRIBUTES=35721
ACTIVE_TEXTURE=34016=33984
ACTIVE_UNIFORMS=35718
ALIASED_LINE_WIDTH_RANGE=33902=11
ALIASED_POINT_SIZE_RANGE=33901=1511
ALPHA=6406
ALPHA_BITS=3413=8
ALWAYS=519
ARRAY_BUFFER=34962
ARRAY_BUFFER_BINDING=34964
ATTACHED_SHADERS=35717
BACK=1029
BLEND=3042=false
BLEND_COLOR=32773=0000
BLEND_DST_ALPHA=32970=0
BLEND_DST_RGB=32968=0
BLEND_EQUATION=32777=32774
BLEND_EQUATION_ALPHA=34877=32774
BLEND_EQUATION_RGB=32777=32774
BLEND_SRC_ALPHA=32971=1
BLEND_SRC_RGB=32969=1
BLUE_BITS=3412=8
BOOL=35670
BOOL_VEC2=35671
BOOL_VEC3=35672
BOOL_VEC4=35673
BROWSER_DEFAULT_WEBGL=37444
BUFFER_SIZE=34660
BUFFER_USAGE=34661
BYTE=5120
CCW=2305
CLAMP_TO_EDGE=33071
COLOR_ATTACHMENT0=36064
COLOR_BUFFER_BIT=16384
COLOR_CLEAR_VALUE=3106=0000
COLOR_WRITEMASK=3107=truetruetruetrue
COMPILE_STATUS=35713
COMPRESSED_TEXTURE_FORMATS=34467=
CONSTANT_ALPHA=32771
CONSTANT_COLOR=32769
CONTEXT_LOST_WEBGL=37442
CULL_FACE=2884=false
CULL_FACE_MODE=2885=1029
CURRENT_PROGRAM=35725
CURRENT_VERTEX_ATTRIB=34342
CW=2304
DECR=7683
DECR_WRAP=34056
DELETE_STATUS=35712
DEPTH_ATTACHMENT=36096
DEPTH_BITS=3414=24
DEPTH_BUFFER_BIT=256
DEPTH_CLEAR_VALUE=2931=1
DEPTH_COMPONENT16=33189
DEPTH_COMPONENT=6402
DEPTH_FUNC=2932=513
DEPTH_RANGE=2928=01
DEPTH_STENCIL=34041
DEPTH_STENCIL_ATTACHMENT=33306
DEPTH_TEST=2929=false
DEPTH_WRITEMASK=2930=true
DITHER=3024=true
DONT_CARE=4352
DST_ALPHA=772
DST_COLOR=774
DYNAMIC_DRAW=35048
ELEMENT_ARRAY_BUFFER=34963
ELEMENT_ARRAY_BUFFER_BINDING=34965
EQUAL=514
FASTEST=4353
FLOAT=5126
FLOAT_MAT2=35674
FLOAT_MAT3=35675
FLOAT_MAT4=35676
FLOAT_VEC2=35664
FLOAT_VEC3=35665
FLOAT_VEC4=35666
FRAGMENT_SHADER=35632
FRAMEBUFFER=36160
FRAMEBUFFER_ATTACHMENT_OBJECT_NAME=36049
FRAMEBUFFER_ATTACHMENT_OBJECT_TYPE=36048
FRAMEBUFFER_ATTACHMENT_TEXTURE_CUBE_MAP_FACE=36051
FRAMEBUFFER_ATTACHMENT_TEXTURE_LEVEL=36050
FRAMEBUFFER_BINDING=36006
FRAMEBUFFER_COMPLETE=36053
FRAMEBUFFER_INCOMPLETE_ATTACHMENT=36054
FRAMEBUFFER_INCOMPLETE_DIMENSIONS=36057
FRAMEBUFFER_INCOMPLETE_MISSING_ATTACHMENT=36055
FRAMEBUFFER_UNSUPPORTED=36061
FRONT=1028
FRONT_AND_BACK=1032
FRONT_FACE=2886=2305
FUNC_ADD=32774
FUNC_REVERSE_SUBTRACT=32779
FUNC_SUBTRACT=32778
GENERATE_MIPMAP_HINT=33170=4352
GEQUAL=518
GREATER=516
GREEN_BITS=3411=8
HIGH_FLOAT=36338
HIGH_INT=36341
IMPLEMENTATION_COLOR_READ_FORMAT=35739=6408
IMPLEMENTATION_COLOR_READ_TYPE=35738=5121
INCR=7682
INCR_WRAP=34055
INT=5124
INT_VEC2=35667
INT_VEC3=35668
INT_VEC4=35669
INVALID_ENUM=1280
INVALID_FRAMEBUFFER_OPERATION=1286
```

```
INVALID_OPERATION=1282
INVALID_VALUE=1281
INVERT=5386
KEEP=7680
LEQUAL=515
LESS=513
LINEAR=9729
LINEAR_MIPMAP_LINEAR=9987
LINEAR_MIPMAP_NEAREST=9985
LINES=1
LINE_LOOP=2
LINE_STRIP=3
LINE_WIDTH=2849=1
LINK_STATUS=35714
LOW_FLOAT=36336
LOW_INT=36339
LUMINANCE=6409
LUMINANCE_ALPHA=6410
MAX_COMBINED_TEXTURE_IMAGE_UNITS=35661=32
MAX_CUBE_MAP_TEXTURE_SIZE=34076=16384
MAX_FRAGMENT_UNIFORM_VECTORS=36349=1024
MAX_RENDERBUFFER_SIZE=34024=16384
MAX_TEXTURE_IMAGE_UNITS=34930=16
MAX_TEXTURE_SIZE=3379=16384
MAX_VARYING_VECTORS=36348=30
MAX_VERTEX_ATTRIBS=34921=16
MAX_VERTEX_TEXTURE_IMAGE_UNITS=35660=16
MAX_VERTEX_UNIFORM_VECTORS=36347=1024
MAX_VIEWPORT_DIMS=3386=1638416384
MEDIUM_FLOAT=36337
MEDIUM_INT=36340
MIRRORED_REPEAT=33648
NEAREST=9728
NEAREST_MIPMAP_LINEAR=9986
NEAREST_MIPMAP_NEAREST=9984
NEVER=512
NICEST=4354
NONE=0
NOTEQUAL=517
NO_ERROR=0
ONE=1
ONE_MINUS_CONSTANT_ALPHA=32772
ONE_MINUS_CONSTANT_COLOR=32770
ONE_MINUS_DST_ALPHA=773
ONE_MINUS_DST_COLOR=775
ONE_MINUS_SRC_ALPHA=771
ONE_MINUS_SRC_COLOR=769
OUT_OF_MEMORY=1285
PACK_ALIGNMENT=3333=4
POINTS=0
POLYGON_OFFSET_FACTOR=32824=0
POLYGON_OFFSET_FILL=32823=false
POLYGON_OFFSET_UNITS=10752=0
RED_BITS=3410=8
RENDERBUFFER=36161
RENDERBUFFER_ALPHA_SIZE=36179
RENDERBUFFER_BINDING=36007
RENDERBUFFER_BLUE_SIZE=36178
RENDERBUFFER_DEPTH_SIZE=36180
RENDERBUFFER_GREEN_SIZE=36177
RENDERBUFFER_HEIGHT=36163
RENDERBUFFER_INTERNAL_FORMAT=36164
RENDERBUFFER_RED_SIZE=36176
RENDERBUFFER_STENCIL_SIZE=36181
RENDERBUFFER_WIDTH=36162
RENDERER=7937=WebKit WebGL
REPEAT=10497
REPLACE=7681
RGB565=36194
RGB5_A1=32855
RGB8=32849
RGB=6407
RGBA4=32854
RGBA8=32856
RGBA=6408
SAMPLER_2D=35678
SAMPLER_CUBE=35680
SAMPLES=32937=4
SAMPLE_ALPHA_TO_COVERAGE=32926
SAMPLE_BUFFERS=32936=1
SAMPLE_COVERAGE=32928
SAMPLE_COVERAGE_INVERT=32939=false
SAMPLE_COVERAGE_VALUE=32938=1
SCISSOR_BOX=3088=00300150
SCISSOR_TEST=3089=false
SHADER_TYPE=35663
SHADING_LANGUAGE_VERSION=35724=WebGL GLSL ES 1.0 (OpenGL ES GLSL ES 1.0
Chromium)
SHORT=5122
SRC_ALPHA=770
SRC_ALPHA_SATURATE=776
SRC_COLOR=768
STATIC_DRAW=35044
STENCIL_ATTACHMENT=36128
STENCIL_BACK_FAIL=34817=7680
STENCIL_BACK_FUNC=34816=519
STENCIL_BACK_PASS_DEPTH_FAIL=34818=7680
STENCIL_BACK_PASS_DEPTH_PASS=34819=7680
STENCIL_BACK_REF=36003=0
STENCIL_BACK_VALUE_MASK=36004=2147483647
STENCIL_BACK_WRITEMASK=36005=2147483647
STENCIL_BITS=3415=0
STENCIL_BUFFER_BIT=1024
STENCIL_CLEAR_VALUE=2961=0
STENCIL_FAIL=2964=7680
STENCIL_FUNC=2962=519
STENCIL_INDEX8=36168
STENCIL_PASS_DEPTH_FAIL=2965=7680
STENCIL_PASS_DEPTH_PASS=2966=7680
STENCIL_REF=2967=0
STENCIL_TEST=2960=false
STENCIL_VALUE_MASK=2963=2147483647
STENCIL_WRITEMASK=2968=2147483647
STREAM_DRAW=35040
SUBPIXEL_BITS=3408=4
TEXTURE0=33984
TEXTURE10=33994
TEXTURE11=33995
TEXTURE12=33996
TEXTURE13=33997
TEXTURE14=33998
TEXTURE15=33999
TEXTURE16=34000
TEXTURE17=34001
TEXTURE18=34002
TEXTURE19=34003
TEXTURE1=33985
TEXTURE20=34004
TEXTURE21=34005
TEXTURE22=34006
TEXTURE23=34007
TEXTURE24=34008
TEXTURE25=34009
TEXTURE26=34010
TEXTURE27=34011
TEXTURE28=34012
TEXTURE29=34013
TEXTURE2=33986
TEXTURE30=34014
TEXTURE31=34015
TEXTURE3=33987
TEXTURE4=33988
TEXTURE5=33989
TEXTURE6=33990
TEXTURE7=33991
TEXTURE8=33992
TEXTURE9=33993
TEXTURE=5890
TEXTURE_2D=3553
TEXTURE_BINDING_2D=32873
TEXTURE_BINDING_CUBE_MAP=34068
TEXTURE_CUBE_MAP=34067
TEXTURE_CUBE_MAP_NEGATIVE_X=34070
TEXTURE_CUBE_MAP_NEGATIVE_Y=34072
TEXTURE_CUBE_MAP_NEGATIVE_Z=34074
TEXTURE_CUBE_MAP_POSITIVE_X=34069
TEXTURE_CUBE_MAP_POSITIVE_Y=34071
TEXTURE_CUBE_MAP_POSITIVE_Z=34073
TEXTURE_MAG_FILTER=10240
TEXTURE_MIN_FILTER=10241
TEXTURE_WRAP_S=10242
TEXTURE_WRAP_T=10243
TRIANGLES=4
TRIANGLE_FAN=6
```

```
TRIANGLE_STRIP=5
UNPACK_ALIGNMENT=3317=4
UNPACK_COLORSPACE_CONVERSION_WEBGL=37443=37444
UNPACK_FLIP_Y_WEBGL=37440=false
UNPACK_PREMULTIPLY_ALPHA_WEBGL=37441=false
UNSIGNED_BYTE=5121
UNSIGNED_INT=5125
UNSIGNED_SHORT=5123
UNSIGNED_SHORT_4_4_4_4=32819
UNSIGNED_SHORT_5_5_5_1=32820
UNSIGNED_SHORT_5_6_5=33635
VALIDATE_STATUS=35715
VENDOR=7936=WebKit
VERSION=7938=WebGL 1.0 (OpenGL ES 2.0 Chromium)
VERTEX_ATTRIB_ARRAY_BUFFER_BINDING=34975
VERTEX_ATTRIB_ARRAY_ENABLED=34338
VERTEX_ATTRIB_ARRAY_NORMALIZED=34922
VERTEX_ATTRIB_ARRAY_POINTER=34373
VERTEX_ATTRIB_ARRAY_SIZE=34339
VERTEX_ATTRIB_ARRAY_STRIDE=34340
VERTEX_ATTRIB_ARRAY_TYPE=34341
VERTEX_SHADER=35633
VIEWPORT=2978=00300150
ZERO=0
]
shaderPrecisions: [
FRAGMENT_SHADER.LOW_FLOAT=12712723
FRAGMENT_SHADER.MEDIUM_FLOAT=12712723
FRAGMENT_SHADER.HIGH_FLOAT=12712723
FRAGMENT_SHADER.LOW_INT=31300
FRAGMENT_SHADER.MEDIUM_INT=31300
FRAGMENT_SHADER.HIGH_INT=31300
VERTEX_SHADER.LOW_FLOAT=12712723
VERTEX_SHADER.MEDIUM_FLOAT=12712723
VERTEX_SHADER.HIGH_FLOAT=12712723
VERTEX_SHADER.LOW_INT=31300
VERTEX_SHADER.MEDIUM_INT=31300
VERTEX_SHADER.HIGH_INT=31300
]
extensions: [
ANGLE_instanced_arrays
EXT_blend_minmax
EXT_clip_control
EXT_color_buffer_half_float
EXT_depth_clamp
EXT_disjoint_timer_query
EXT_float_blend
EXT_frag_depth
EXT_polygon_offset_clamp
EXT_shader_texture_lod
EXT_texture_compression_bptc
EXT_texture_compression_rgtc
EXT_texture_filter_anisotropic
EXT_texture_mirror_clamp_to_edge
EXT_sRGB
KHR_parallel_shader_compile
OES_element_index_uint
OES_fbo_render_mipmap
OES_standard_derivatives
OES_texture_float
OES_texture_float_linear
OES_texture_half_float
OES_texture_half_float_linear
OES_vertex_array_object
WEBGL_blend_func_extended
WEBGL_color_buffer_float
WEBGL_compressed_texture_astc
WEBGL_compressed_texture_etc
WEBGL_compressed_texture_etc1
WEBGL_compressed_texture_pvrtc
WEBGL_compressed_texture_s3tc
WEBGL_compressed_texture_s3tc_srgb
WEBGL_debug_renderer_info
WEBGL_debug_shaders
WEBGL_depth_texture
WEBGL_draw_buffers
WEBGL_lose_context
WEBGL_multi_draw
WEBGL_polygon_mode
]
extensionParameters: [
CLIP_DEPTH_MODE_EXT=37725
CLIP_ORIGIN_EXT=37724
COLOR_ATTACHMENT0_WEBGL=36064
COLOR_ATTACHMENT10_WEBGL=36074
COLOR_ATTACHMENT11_WEBGL=36075
COLOR_ATTACHMENT12_WEBGL=36076
COLOR_ATTACHMENT13_WEBGL=36077
COLOR_ATTACHMENT14_WEBGL=36078
COLOR_ATTACHMENT15_WEBGL=36079
COLOR_ATTACHMENT1_WEBGL=36065
COLOR_ATTACHMENT2_WEBGL=36066
COLOR_ATTACHMENT3_WEBGL=36067
COLOR_ATTACHMENT4_WEBGL=36068
COLOR_ATTACHMENT5_WEBGL=36069
COLOR_ATTACHMENT6_WEBGL=36070
COLOR_ATTACHMENT7_WEBGL=36071
COLOR_ATTACHMENT8_WEBGL=36072
COLOR_ATTACHMENT9_WEBGL=36073
COMPLETION_STATUS_KHR=37297
COMPRESSED_R11_EAC=37488
COMPRESSED_RED_GREEN_RGTC2_EXT=36285
COMPRESSED_RED_RGTC1_EXT=36283
COMPRESSED_RG11_EAC=37490
COMPRESSED_RGB8_ETC2=37492
COMPRESSED_RGB8_PUNCHTHROUGH_ALPHA1_ETC2=37494
COMPRESSED_RGBA8_ETC2_EAC=37496
COMPRESSED_RGBA_ASTC_10x10_KHR=37819
COMPRESSED_RGBA_ASTC_10x5_KHR=37816
COMPRESSED_RGBA_ASTC_10x6_KHR=37817
COMPRESSED_RGBA_ASTC_10x8_KHR=37818
COMPRESSED_RGBA_ASTC_12x10_KHR=37820
COMPRESSED_RGBA_ASTC_12x12_KHR=37821
COMPRESSED_RGBA_ASTC_4x4_KHR=37808
COMPRESSED_RGBA_ASTC_5x4_KHR=37809
COMPRESSED_RGBA_ASTC_5x5_KHR=37810
COMPRESSED_RGBA_ASTC_6x5_KHR=37811
COMPRESSED_RGBA_ASTC_6x6_KHR=37812
COMPRESSED_RGBA_ASTC_8x5_KHR=37813
COMPRESSED_RGBA_ASTC_8x6_KHR=37814
COMPRESSED_RGBA_ASTC_8x8_KHR=37815
COMPRESSED_RGBA_BPTC_UNORM_EXT=36492
COMPRESSED_RGBA_PVRTC_2BPPV1_IMG=35843
COMPRESSED_RGBA_PVRTC_4BPPV1_IMG=35842
COMPRESSED_RGBA_S3TC_DXT1_EXT=33777
COMPRESSED_RGBA_S3TC_DXT3_EXT=33778
COMPRESSED_RGBA_S3TC_DXT5_EXT=33779
COMPRESSED_RGB_BPTC_SIGNED_FLOAT_EXT=36494
COMPRESSED_RGB_BPTC_UNSIGNED_FLOAT_EXT=36495
COMPRESSED_RGB_ETC1_WEBGL=36196
COMPRESSED_RGB_PVRTC_2BPPV1_IMG=35841
COMPRESSED_RGB_PVRTC_4BPPV1_IMG=35840
COMPRESSED_RGB_S3TC_DXT1_EXT=33776
COMPRESSED_SIGNED_R11_EAC=37489
COMPRESSED_SIGNED_RED_GREEN_RGTC2_EXT=36286
COMPRESSED_SIGNED_RED_RGTC1_EXT=36284
COMPRESSED_SIGNED_RG11_EAC=37491
COMPRESSED_SRGB8_ALPHA8_ASTC_10x10_KHR=37851
COMPRESSED_SRGB8_ALPHA8_ASTC_10x5_KHR=37848
COMPRESSED_SRGB8_ALPHA8_ASTC_10x6_KHR=37849
COMPRESSED_SRGB8_ALPHA8_ASTC_10x8_KHR=37850
COMPRESSED_SRGB8_ALPHA8_ASTC_12x10_KHR=37852
COMPRESSED_SRGB8_ALPHA8_ASTC_12x12_KHR=37853
COMPRESSED_SRGB8_ALPHA8_ASTC_4x4_KHR=37840
COMPRESSED_SRGB8_ALPHA8_ASTC_5x4_KHR=37841
COMPRESSED_SRGB8_ALPHA8_ASTC_5x5_KHR=37842
COMPRESSED_SRGB8_ALPHA8_ASTC_6x5_KHR=37843
COMPRESSED_SRGB8_ALPHA8_ASTC_6x6_KHR=37844
COMPRESSED_SRGB8_ALPHA8_ASTC_8x5_KHR=37845
COMPRESSED_SRGB8_ALPHA8_ASTC_8x6_KHR=37846
COMPRESSED_SRGB8_ALPHA8_ASTC_8x8_KHR=37847
COMPRESSED_SRGB8_ALPHA8_ETC2_EAC=37497
COMPRESSED_SRGB8_ETC2=37493
COMPRESSED_SRGB8_PUNCHTHROUGH_ALPHA1_ETC2=37495
COMPRESSED_SRGB_ALPHA_BPTC_UNORM_EXT=36493
COMPRESSED_SRGB_ALPHA_S3TC_DXT1_EXT=35917
COMPRESSED_SRGB_ALPHA_S3TC_DXT3_EXT=35918
COMPRESSED_SRGB_ALPHA_S3TC_DXT5_EXT=35919
COMPRESSED_SRGB_S3TC_DXT1_EXT=35916
CURRENT_QUERY_EXT=34917
DEPTH_CLAMP_EXT=34383
DRAW_BUFFER0_WEBGL=34853=1029
DRAW_BUFFER10_WEBGL=34863
DRAW_BUFFER11_WEBGL=34864
```

```
DRAW_BUFFER12_WEBGL=34865
DRAW_BUFFER13_WEBGL=34866
DRAW_BUFFER14_WEBGL=34867
DRAW_BUFFER15_WEBGL=34868
DRAW_BUFFER1_WEBGL=34854=1029
DRAW_BUFFER2_WEBGL=34855
DRAW_BUFFER3_WEBGL=34856
DRAW_BUFFER4_WEBGL=34857
DRAW_BUFFER5_WEBGL=34858
DRAW_BUFFER6_WEBGL=34859
DRAW_BUFFER7_WEBGL=34860
DRAW_BUFFER8_WEBGL=34861
DRAW_BUFFER9_WEBGL=34862
FRAGMENT_SHADER_DERIVATIVE_HINT_OES=35723=4352
FRAMEBUFFER_ATTACHMENT_COLOR_ENCODING_EXT=33296
FRAMEBUFFER_ATTACHMENT_COMPONENT_TYPE_EXT=33297
FRAMEBUFFER_ATTACHMENT_COMPONENT_TYPE_EXT=33297
GPU_DISJOINT_EXT=36795=false
HALF_FLOAT_OES=36193
LOWER_LEFT_EXT=36001
MAX_COLOR_ATTACHMENTS_WEBGL=36063=8
MAX_DRAW_BUFFERS_WEBGL=34852=8
MAX_DUAL_SOURCE_DRAW_BUFFERS_WEBGL=35068
MAX_EXT=32776
MAX_TEXTURE_MAX_ANISOTROPY_EXT=34047=16
MIN_EXT=32775
MIRROR_CLAMP_TO_EDGE_EXT=34627
NEGATIVE_ONE_TO_ONE_EXT=37726
ONE_MINUS_SRC1_ALPHA_WEBGL=35067
ONE_MINUS_SRC1_COLOR_WEBGL=35066
POLYGON_OFFSET_CLAMP_EXT=36379
QUERY_COUNTER_BITS_EXT=34916
QUERY_RESULT_AVAILABLE_EXT=34919
QUERY_RESULT_EXT=34918
RGB16F_EXT=34843
RGBA16F_EXT=34842
RGBA32F_EXT=34836
SRC1_ALPHA_WEBGL=34185
SRC1_COLOR_WEBGL=35065
SRGB8_ALPHA8_EXT=35907
SRGB_ALPHA_EXT=35906
SRGB_EXT=35904
TEXTURE_MAX_ANISOTROPY_EXT=34046
TIMESTAMP_EXT=36392=0
TIME_ELAPSED_EXT=35007
UNMASKED_RENDERER_WEBGL=37446
UNMASKED_VENDOR_WEBGL=37445
UNSIGNED_INT_24_8_WEBGL=34042
UNSIGNED_NORMALIZED_EXT=35863
UNSIGNED_NORMALIZED_EXT=35863
UPPER_LEFT_EXT=36002
VERTEX_ARRAY_BINDING_OES=34229=null
VERTEX_ATTRIB_ARRAY_DIVISOR_ANGLE=35070
ZERO_TO_ONE_EXT=37727
```