EXPLORING THE POWER OF
Browser Storage

ConFoo.CA
DEVELOPER CONFERENCE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

EXPLORING THE POWER OF BROWSER STORAGE

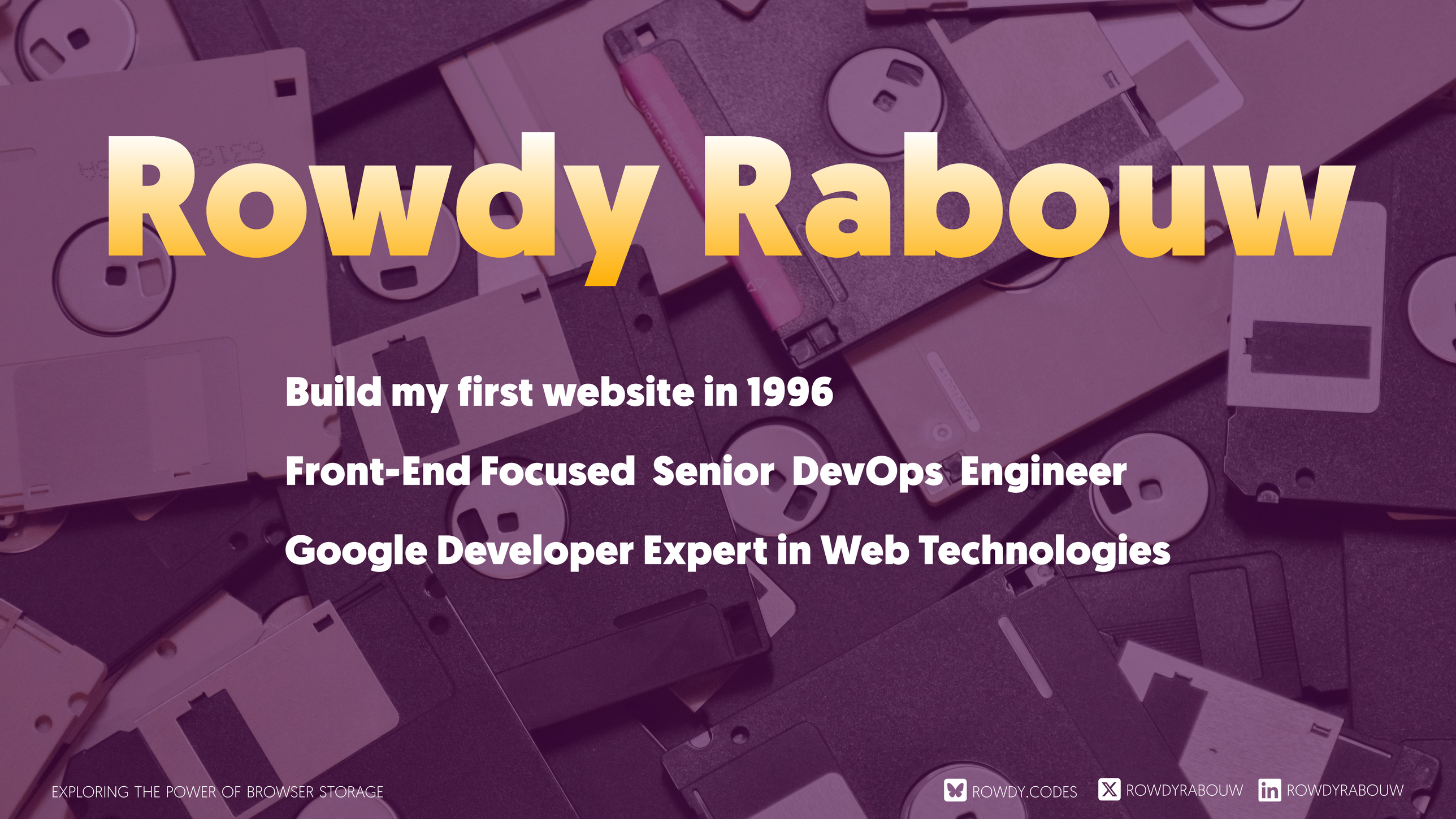
ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

Boeing 747-200 (February 1987)

Geosky Georgia


UNITED STATES OF AMERICA
28000


I'M OLD,
NOT
OBSOLETE



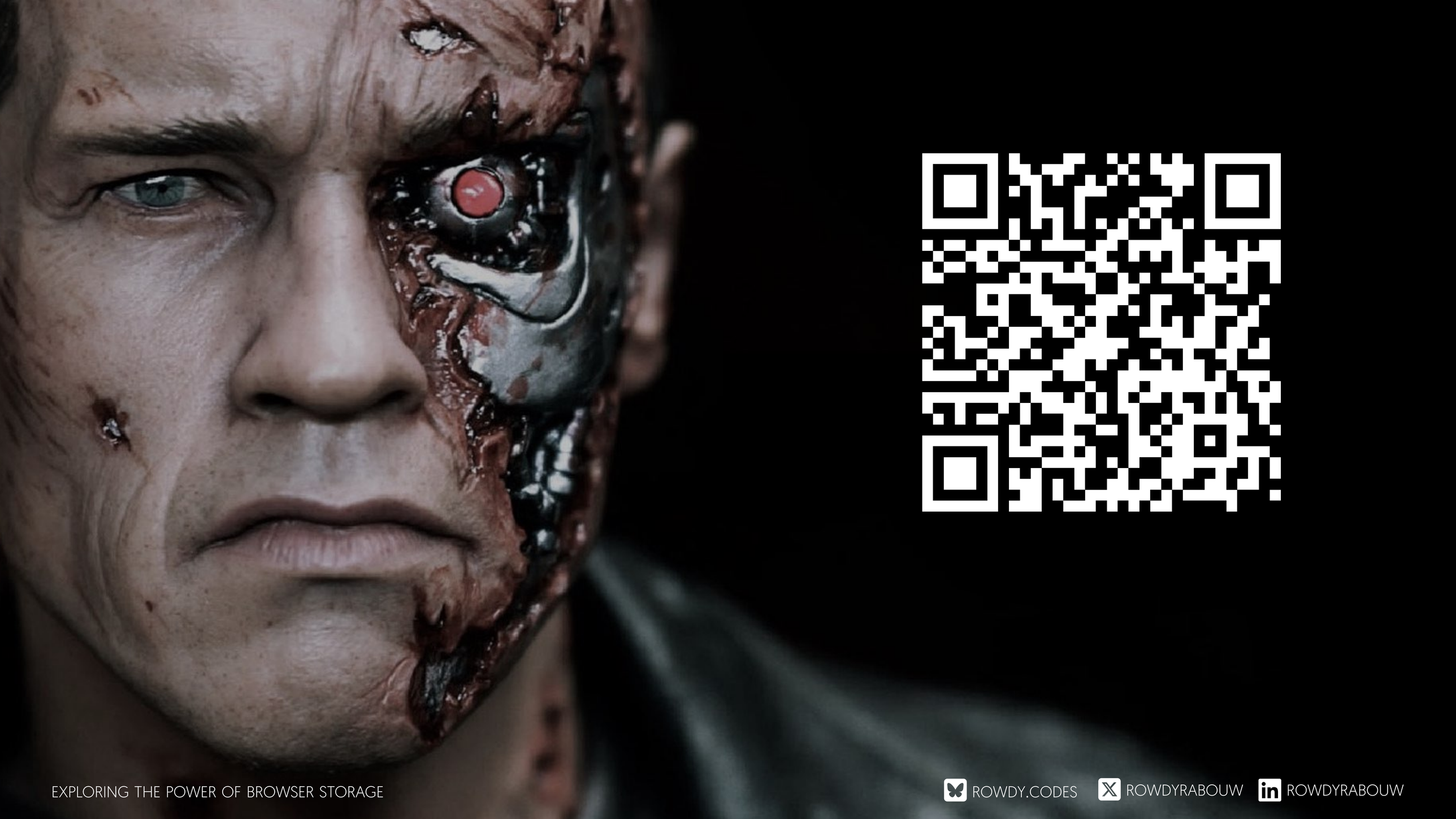
Rowdy Rabouw

Build my first website in 1996

Front-End Focused Senior DevOps Engineer

Google Developer Expert in Web Technologies

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW



EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

EXPLORING THE POWER OF Browser Storage

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

Cookies

IndexedDB

Local Storage

File System API

Session Storage

AppCache

WebSQL

Cache Storage

Cookies

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

Cookies

Formally known as HTTP Cookies

Synchronous, blocking, and single-threaded

Tracking

Login

Analytics

```
document.cookie = 'Favorite=Chocolate; path=/; max-age=31536000;';
```

path: defaults to the current path

domain: defaults to the current hostname

expires: date in GMT format

max-age: in seconds

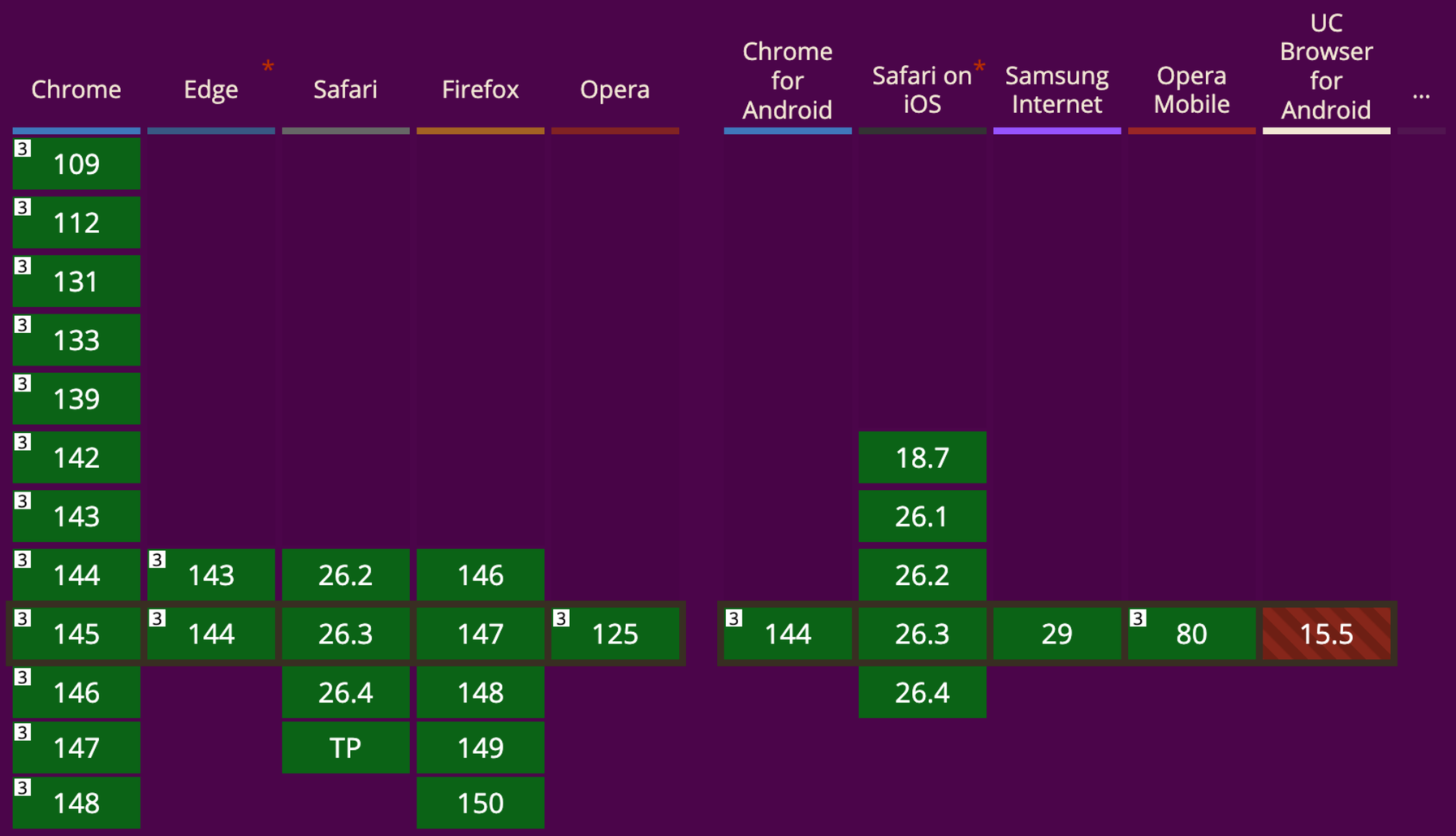
same-site:

lax = same-site requests and navigation GET requests

strict = not sent to any external sites

none = no restrictions

Cookies


Chrome
Edge *
Safari
Firefox
Opera
Chrome for Android
Safari on iOS *
Samsung Internet
Opera Mobile
UC Browser for Android
...
3 109
3 112
3 131
3 133
3 139
3 142
3 143
3 144
3 145
3 146
3 147
3 148
3 143
3 144
26.2
26.3
26.4
TP
146
147
148
149
150
3 125
3 144
18.7
26.1
26.2
26.3
26.4
29
3 80
15.5


DEMO

Cookie Store

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

Cookie Store

Asynchronous and non-blocking

Works in Service Workers

```
cookieStore.set('Favorite', 'Chocolate');

cookieStore.get('Favorite');

cookieStore.getAll('Favorite');

cookieStore.delete('Favorite');
```

```
const day = 24 * 60 * 60 * 1000;

cookieStore.set({
  name: "Favorite",
  value: "Chocolate",
  expires: Date.now() + day,
  path: "/",
});
```

path: defaults to the current path

domain: defaults to the current hostname

expires: date in GMT format

same-site:

lax = same-site requests and navigation GET requests

strict = not sent to any external sites

none = no restrictions

Cookie Store

Chrome	Edge *	Safari	Firefox	Opera	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mobile	UC Browser for Android	...
109										
112										
131										
133										
139										
142						18.7				
143						26.1				
144	143	26.2	146			26.2				
145	144	26.3	147	125	144	26.3	29	80	15.5	
146		26.4	148			26.4				
147		TP	149							
148			150							

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

Web Storage

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

Web Storage

Key and Value pairs

Key and Value are always a strings

Web Storage

Object

JSON.stringify / JSON.parse

Array

array.toString / string.split(“,”)

Local Storage
Session Storage

Local Storage

Not bound to a session (tab or window)

Persistent “forever”

Available on next visit

```
localStorage.setItem(key, value);

localStorage.getItem(key);

localStorage.removeItem(key);
```

Local Storage

Chrome	Edge *	Safari	Firefox	Opera	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mobile	UC Browser for Android	...
109										
112										
131										
133										
139										
142						18.7				
143						26.1				
144	143	26.2	146			26.2				
145	144	26.3	147	125	144	26.3	29	80	15.5	
146		26.4	148			26.4				
147		TP	149							
148			150							

Session Storage

Bound to Session (tab or window)

Duplicating a tab will copy Session Storage

```
sessionStorage.setItem(key, value);

sessionStorage.getItem(key);

sessionStorage.removeItem(key);
```

Session Storage

Chrome	Edge *	Safari	Firefox	Opera	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mobile	UC Browser for Android	...
109										
112										
131										
133										
139										
142						18.7				
143						26.1				
144	143	26.2	146			26.2				
145	144	26.3	147	125	144	26.3	29	80	15.5	
146		26.4	148			26.4				
147		TP	149							
148			150							


DEMO

WebSQL

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

WebSQL

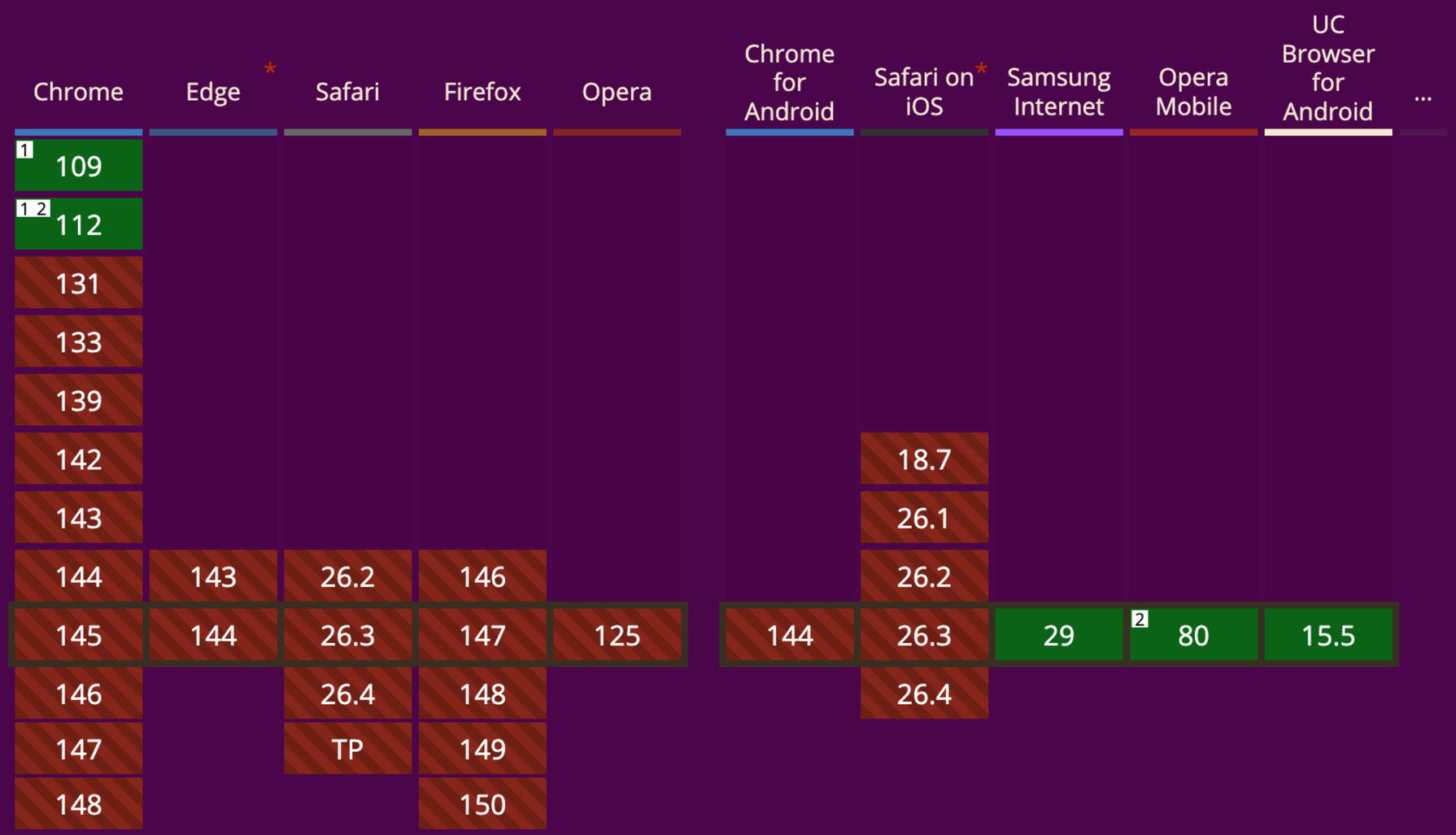
Bringing SQLite to the frontend

Never implemented in Firefox

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

```
openDatabase('mydatabase', 1, 'mydatabase', 5000000, function (db) {
  db.transaction(function (tx) {
    tx.executeSql('create table rainstorms (mood text, severity int)',
      [], function () {
        tx.executeSql('insert into rainstorms values (?, ?)',
          ['somber', 6], function () {
            tx.executeSql('select * from rainstorms where mood = ?',
              ['somber'], function (tx, res) {
                var row = res.rows.item(0);
                console.log('rainstorm severity: ' + row.severity +
                  ',  my mood: ' + row.mood);
              });
          });
      });
  }, function (err) {
    console.log('boo, transaction failed!: ' + err);
  }, function () {
    console.log('yay, transaction succeeded!');
  });
});
```

WebSQL


Chrome
Edge *
Safari
Firefox
Opera
Chrome for Android
Safari on iOS *
Samsung Internet
Opera Mobile
UC Browser for Android
...
1 109
1 2 112
131
133
139
142
143
144
145
146
147
148
143
144
26.2
26.3
26.4
TP
146
147
148
149
150
125
144
18.7
26.1
26.2
26.3
26.4
29
2 80
15.5

IndexedDB

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

IndexedDB

Replacement for WebSQL but as NoSQL

Tables are object stores

Store values by keys, like Web Storage

Supports transactions for reliability

Supports indexes

IndexedDB

Chrome	Edge *	Safari	Firefox	Opera	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mobile	UC Browser for Android	...
109										
112										
131										
133										
139										
142						18.7				
143						26.1				
144	143	26.2	146			26.2				
145	144	26.3	147	125	144	26.3	29	80	15.5	
146		26.4	148			26.4				
147		TP	149							
148			150							

```
let db;

const openDB = indexedDB.open("groceries-IndexedDB", 1);

openDB.onupgradeneeded = function (event) {
  db = openDB.result;
  const oldVer = event.oldVersion;
  if (oldVer < 1) {
    db.createObjectStore('groceryList', { keyPath: "product" });
  }
};
```

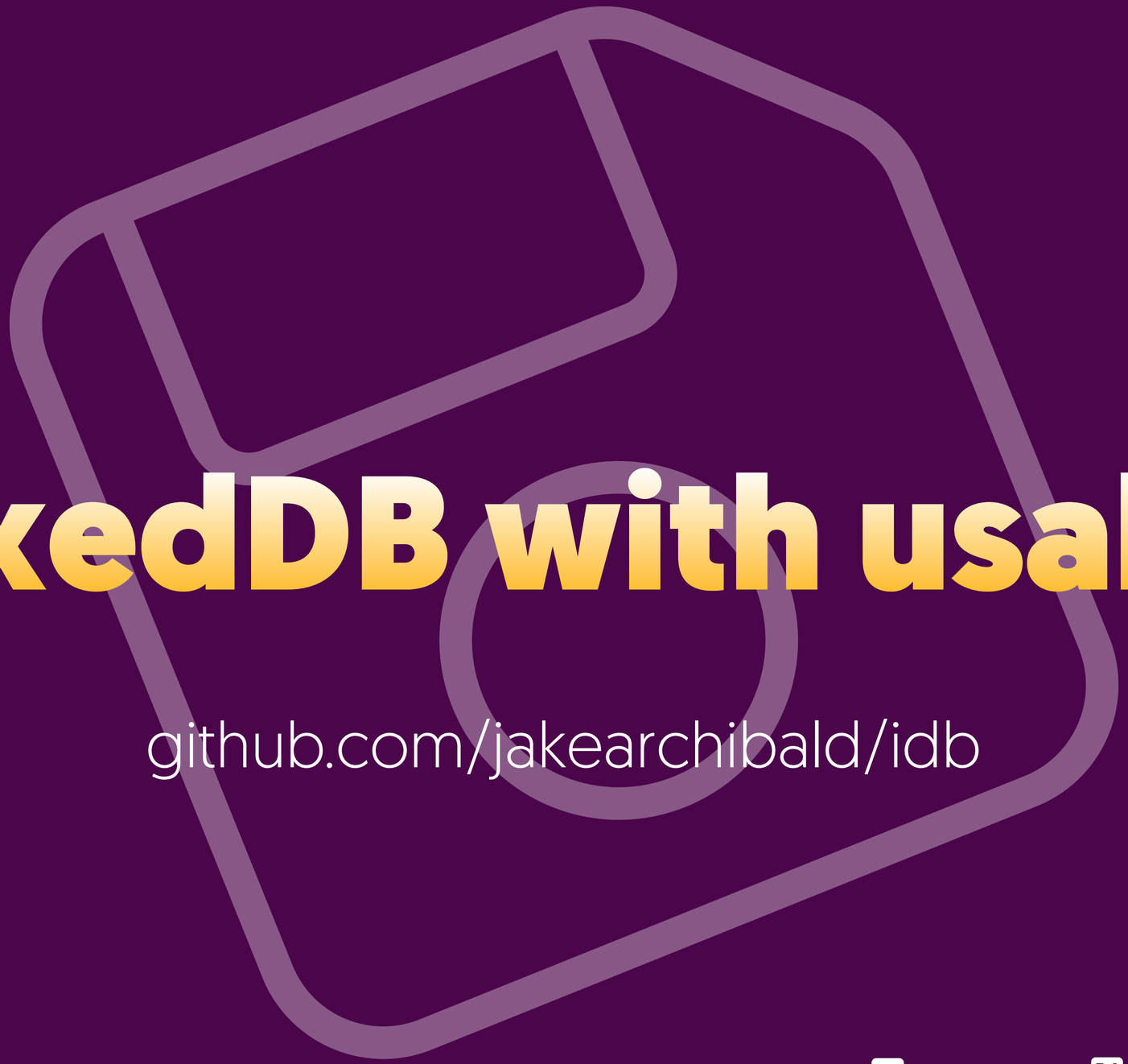
```
openDB.onsuccess = function () {
  console.info("Database opened successfully");
  db = openDB.result;
};

openDB.onerror = function () {
  console.error(openDB.error);
};
```

```
const store =
db.transaction("groceryList", "readwrite").objectStore("groceryList");

let request = store.put({ product: "Orange Juice", amount: 1 });

request.onsuccess = function () {
  console.log("Product added:", request.result);
};

request.onerror = function () {
  console.warn(request.error);
};
```


DEMO

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW



IndexedDB with usability

github.com/jakearchibald/idb

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

idb

Promises enables async / await

Shortcuts for common transactions like getAll, put, delete

```
let db;

const openDatabase = async () ⇒ {
  db = await idb.openDB('groceries-idb', 1, {
    upgrade(db, oldVersion, newVersion, transaction, event) {
     if (oldVersion < 1) {
        db.createObjectStore('groceryList', { keyPath: "product" });
      }
    }
  });
};

await db.put('groceryList', { product: "Orange Juice", amount: 1 });
```


DEMO

Data Synchronization

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

script.js

```
import './idb.js';
import { openDatabase, saveData } from "./utils.js";

async function useWorker() {
  try {
    const worker = new Worker('worker.js', { type: 'module' });
    worker.postMessage('CheckNetworkState');
  }
  catch (error) {
    console.error('Failed to create worker:', error);
  }
}
```

worker.js

```
onmessage = (e) => {
  if (e.data === 'CheckNetworkState') {
    checkNetworkState();
  }
};

const checkNetworkState = () => {
  setInterval(() => {
    if (navigator.onLine) {
      console.info('NetworkState: online')
      syncRecords();
    }
    else {
      console.info('NetworkState: offline')
    }
  }, 3000);
}
```

worker.js

```
const syncRecords = async () => {
  const db = await openDatabase();
  const records = await db.getAllFromIndex('groceryList', 'synced', 'false');
  db.close();

  records.forEach(async (record) => {
    const response = await fetch('http://127.0.0.1:3000', {
      method: "post",
      headers: {
        'Accept': 'application/json',
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(record)
    });

    if (response.status === 200) {
      await saveData([{ ...record, synced: 'true' }]);
    }
  });
}
```

utils.js

```
export const saveData = async (data) ⇒ {
  if (data.length > 0) {
    const db = await openDatabase();
    data.forEach(async (item) ⇒ {
      await db.put('groceryList', item);
    });
    db.close();
  }
}
```


DEMO

Access File System API

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

File System Access API

Interact with files on the user's local device

Adobe Photoshop Web Version

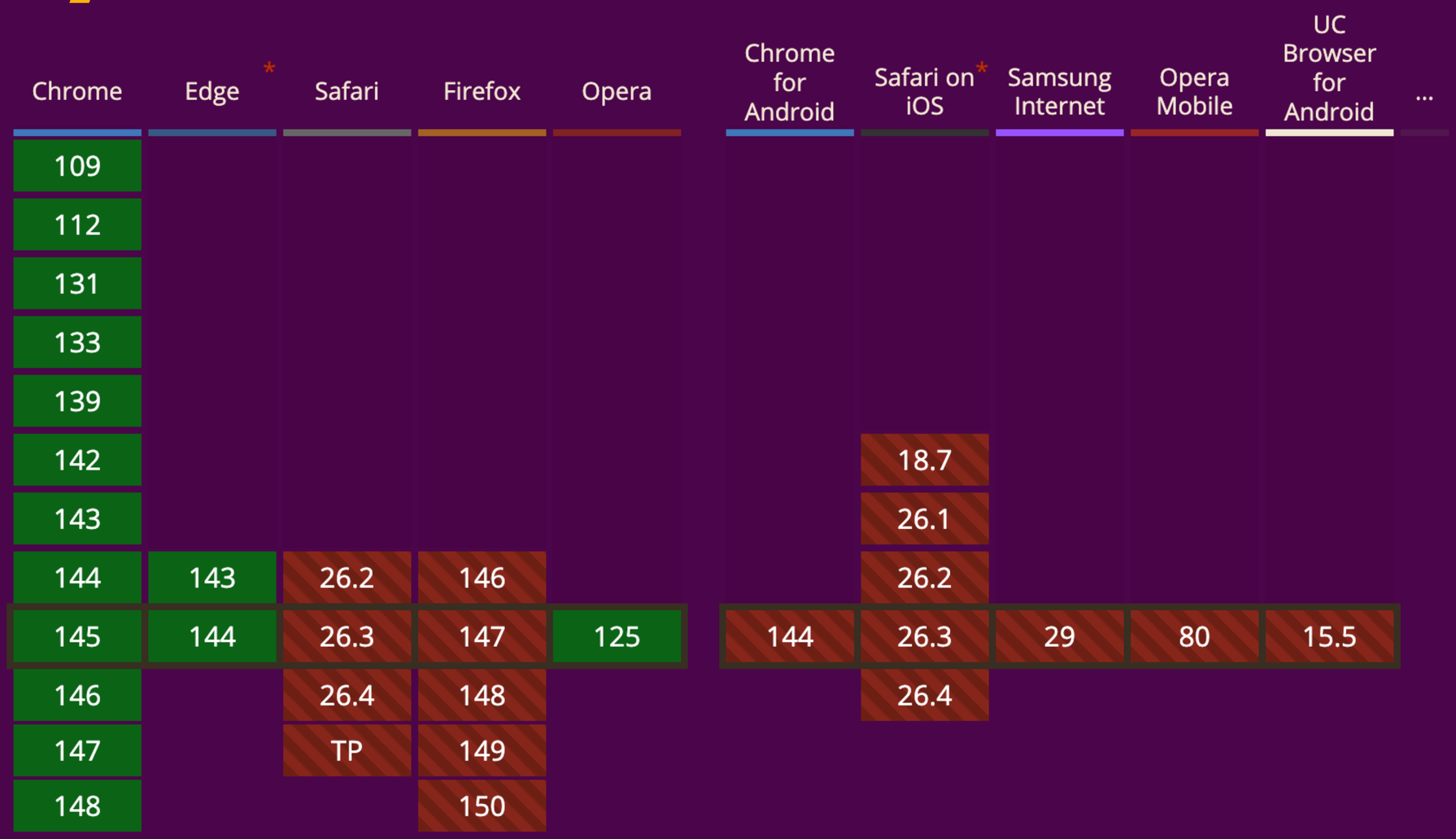
vscode.dev

```
const saveFile = async () => {
  const fileHandle = await window.showSaveFilePicker({
    types: [
      {
        accept: { 'text/plain': ['.txt'] }
      }
    ]
  });
  const writable = await fileHandle.createWritable();
  await writable.write('Hello World!');
  await writable.close();
};
```

```
const readFile = async () => {
  const fileHandle = await window.showOpenFilePicker(
    {
      types: [
        {
          accept: { 'text/plain': ['.txt'] }
        }
      ]
    });
  const file = await fileHandle[0].getFile();
  const contents = await file.text();
  console.log(contents);
};
```


DEMO

File System Access API



ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

File API

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

File API

Origin Private File System

Virtual drive within the browser sandbox

No access to the users' file system

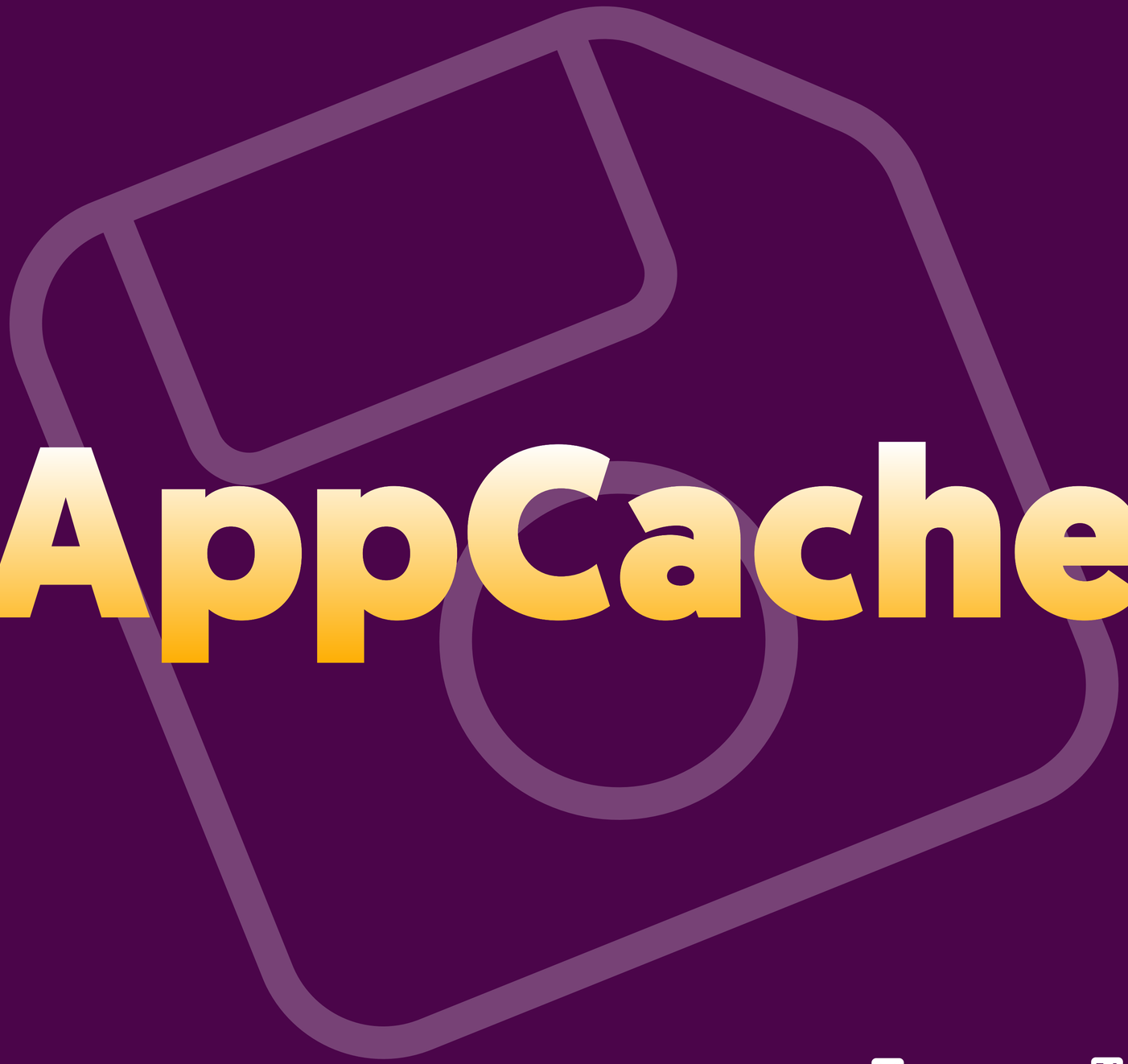
input type = “file” or via drag and drop

Read-only

```
const fileInput = document.querySelector("input[type=file]");
const output = document.querySelector(".output");

fileInput.addEventListener("change", async () ⇒ {
  const [file] = fileInput.files;

  if (file) {
    output.innerText = await file.text();
  }
});
```

File API

Chrome	Edge *	Safari	Firefox	Opera	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mobile	UC Browser for Android	...
109										
112										
131										
133										
139										
142						18.7				
143						26.1				
144	143	26.2	146			26.2				
145	144	26.3	147	125	144	26.3	29	80	15.5	
146		26.4	148			26.4				
147		TP	149							
148			150							



AppCache

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

```html
<!DOCTYPE html>
<html lang="en" manifest="offline.appcache">
```

```
CACHE MANIFEST
# v2

favicon.ico
js/script.js
css/styles.css
https://code.jquery.com/jquery-3.7.1.min.js

NETWORK:
img/hero.jpg
```

AppCache

Files are only cached if all files in the manifest are available

HTML file that has manifest is cached as well

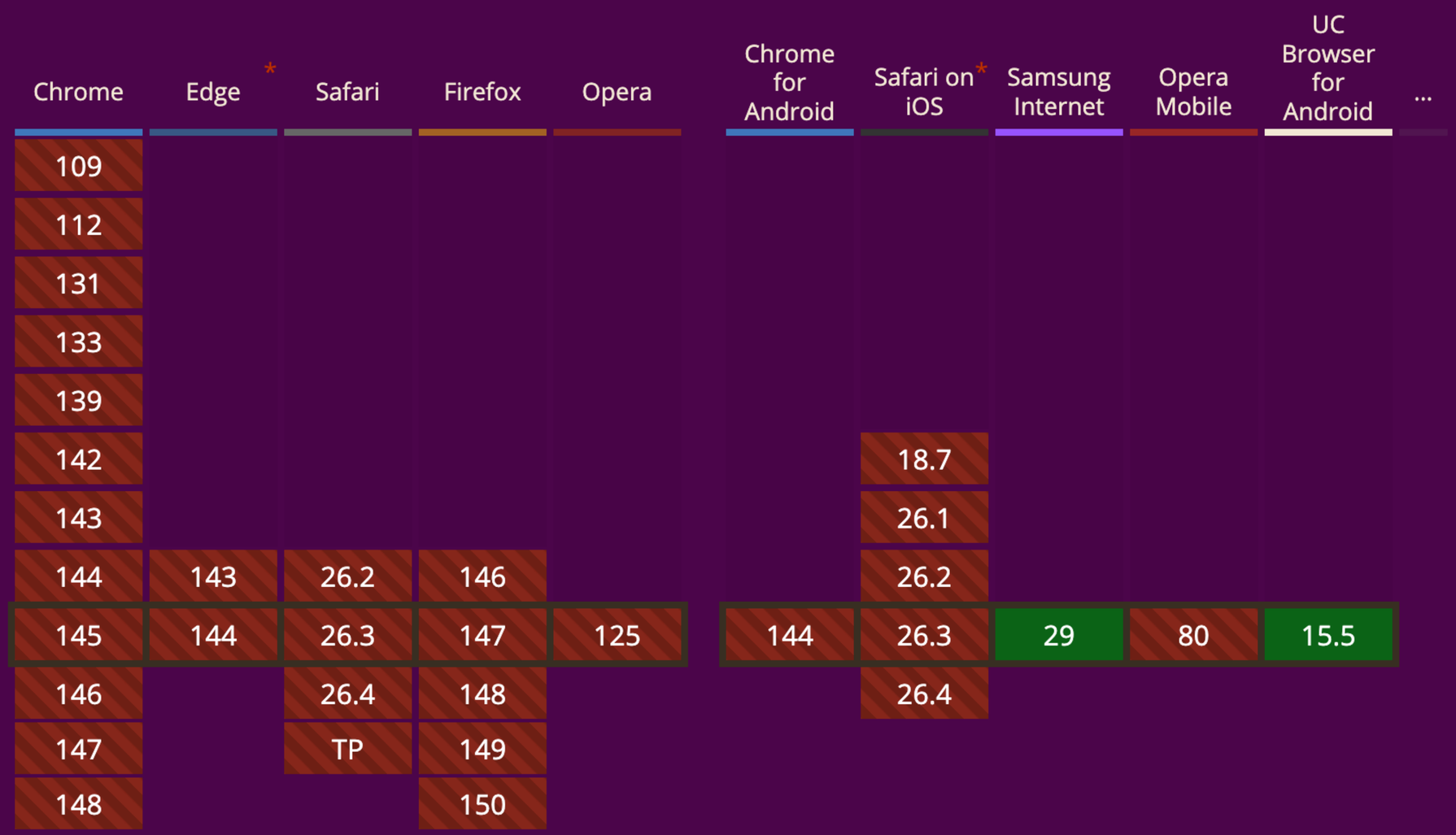
Cache files are always served from appcache

HTML updates require update manifest

Online only assets (like analytics) must be in NETWORK

mobilism
2012

AppCache


Chrome
Edge *
Safari
Firefox
Opera
Chrome for Android
Safari on iOS *
Samsung Internet
Opera Mobile
UC Browser for Android
...
109
112
131
133
139
142
143
144
145
146
147
148
143
144
26.2
26.3
26.4
TP
146
147
148
149
150
125
144
18.7
26.1
26.2
26.3
26.4
29
80
15.5

Cache Storage

EXPLORING THE POWER OF BROWSER STORAGE

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW

Cache Storage

Typically used with Service Workers

Pre-cache static assets

Cache assets on the fly

Serve assets from the cache for performance and offline access

```
const STATIC_CACHE = "static";
const STATIC_FILES = [
  "index-3.html",
  "script-3.js",
  "../assets/svg/floppy.svg",
  "../assets/css/style.css",
];

self.addEventListener("install", async () => {
  const cache = await caches.keys();
  if (!cache.length) {
    caches.open(STATIC_CACHE).then((cache) => {
      cache.addAll(STATIC_FILES);
    });
  }
  self.skipWaiting();
});
```

```
DYNAMIC_CACHE = "dynamic";

self.addEventListener("fetch", (event) => {
  event.respondWith(getFromNetworkOrCache(event.request));
});

const getFromNetworkOrCache = async (request) => {
  const cacheResponse = await caches.match(request);
  if (cacheResponse) {
    return cacheResponse;
  }
  else {
    const networkResponse = await fetch(request);
    const cache = await caches.open(DYNAMIC_CACHE);
    cache.put(request, networkResponse.clone());
    return networkResponse;
  }
}
```


DEMO

Cache Storage

Chrome	Edge *	Safari	Firefox	Opera	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mobile	UC Browser for Android	...
109										
112										
131										
133										
139										
142						18.7				
143						26.1				
144	143	26.2	146			26.2				
145	144	26.3	147	125	144	26.3	29	80	15.5	
146		26.4	148			26.4				
147		TP	149							
148			150							

Merci beaucoup!

rowdy.codes/storage

ROWDY.CODES ROWDYRABOUW ROWDYRABOUW