



Browsers are the new Operating Systems

Diyorbek Sadullaev

Sr. Software Engineer, ex-Pinterest

Mentor @ 42.uz



Google
Developer
Groups



Browsers are the new Application Platforms

Diyorbek Sadullaev

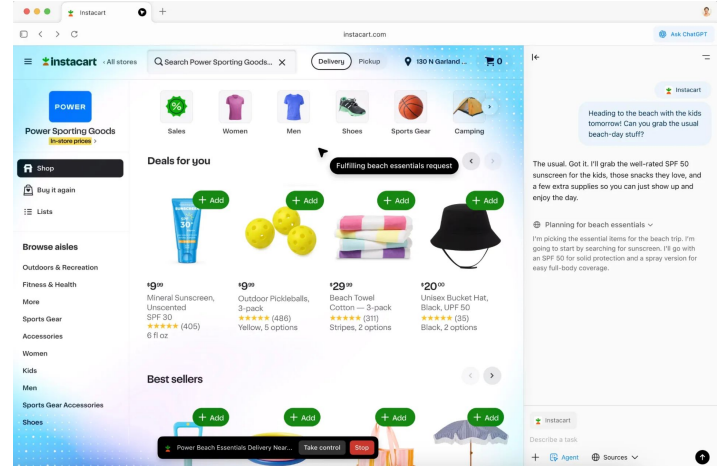
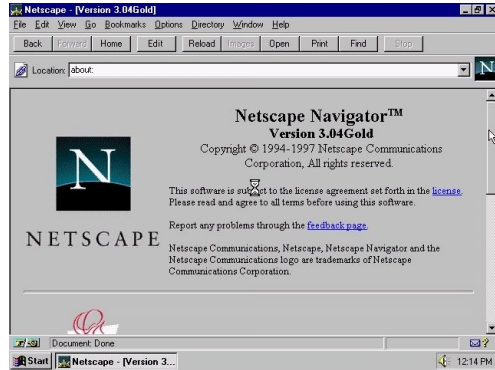
Sr. Software Engineer, ex-Pinterest

Mentor @ 42.uz



Google
Developer
Groups

Evolution of Browsers



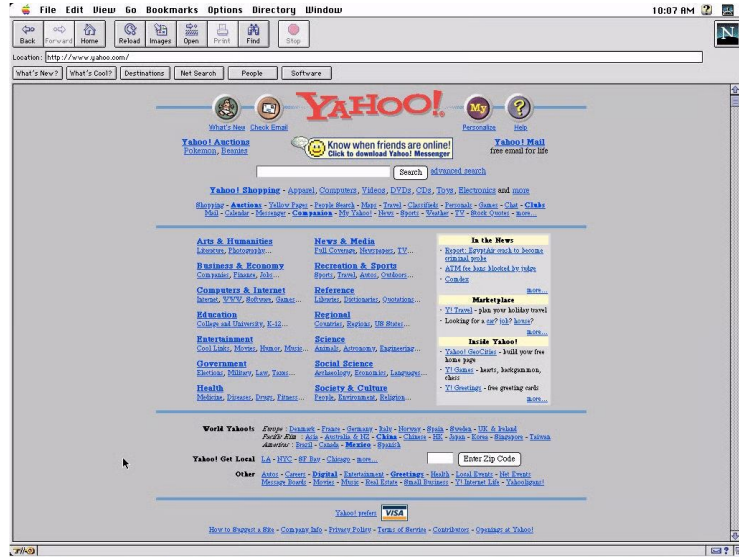
Google
Developer
Groups

1990s



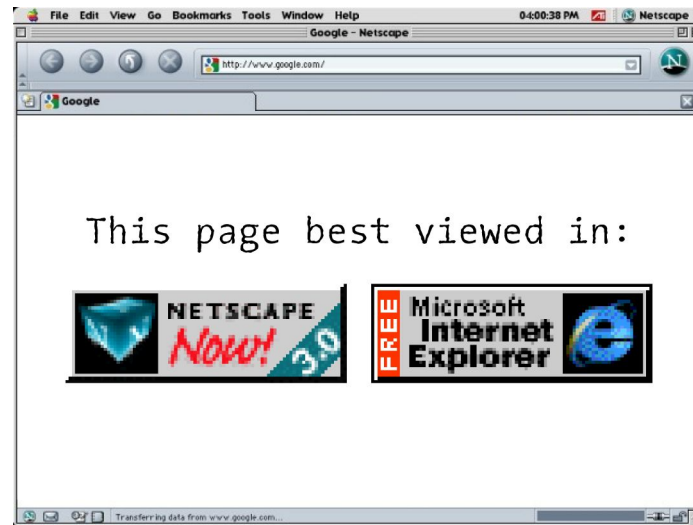
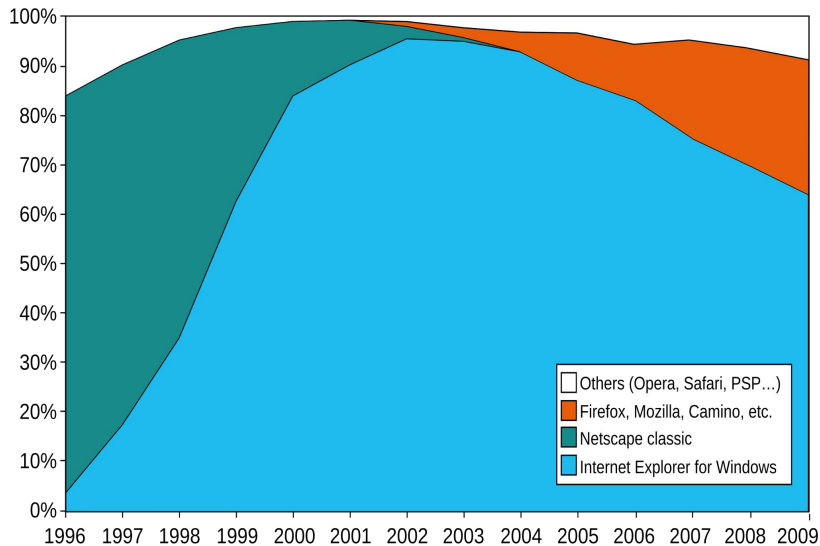
Google
Developer
Groups

Early browsers



Google
Developer
Groups

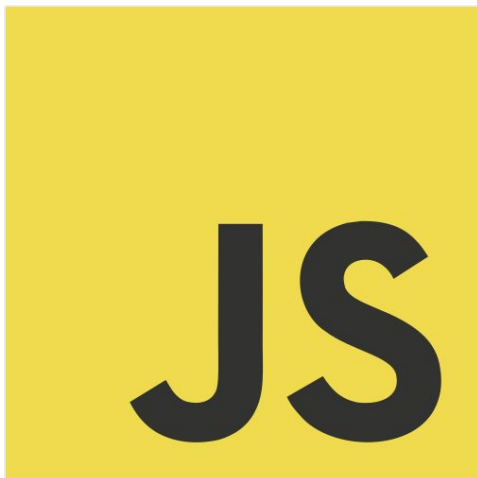
Browser Wars



Google
Developer
Groups



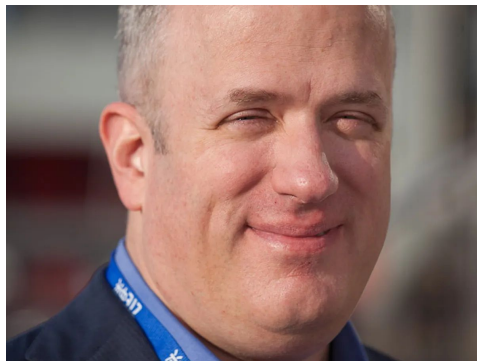
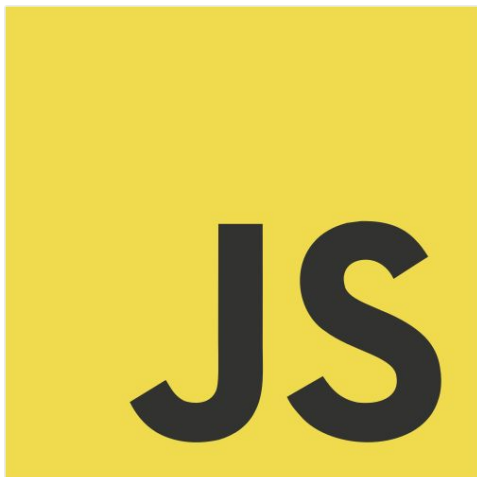
Netscape



Google
Developer
Groups



Netscape



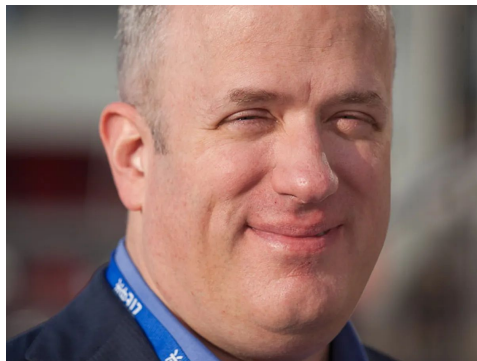
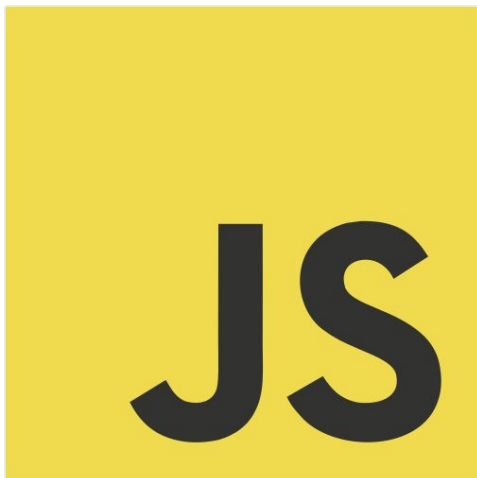
Brendan Eich



Google
Developer
Groups



Netscape



Brendan Eich

in 10 days!



Google
Developer
Groups

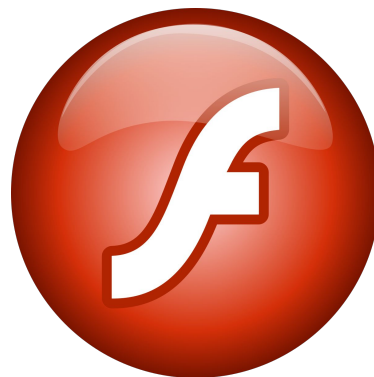


JavaScript

JS



The Blind Alley: **Plugins**



Google
Developer
Groups

ActiveX, Flash, Java Applets **failed** because:

- Closed source & Proprietary
- Security nightmares
- Resource hogs
- Poor portability
- Blocked by mobile (iOS)



Google
Developer
Groups

The lesson was clear:
**You can't build the future of the web on
closed, proprietary technology.**



Google
Developer
Groups

Rise of Open Standards

The logo for AJAX, featuring the letters 'A', 'J', 'A', and 'X' in a blue, sans-serif font. The letter 'J' is stylized with a blue arrow pointing downwards from its top, and the letter 'A' has a blue arrow pointing upwards from its bottom.

Google
Developer
Groups

Rise of Open Standards

- **HTML5** (semantic elements, video/audio)
- **CSS3** (advanced layouts and animations)
- **V8** from Chrome (making JavaScript faster)
- **WebGL** (GPU-accelerated 3D graphics)
- **Web Workers** (parallel execution)



Rise of The Mobile

Now companies faced a problem:
They needed to build for

- **iOS**
- **Android**
- **Web**
- **AND desktop.**



Google
Developer
Groups

Rise of The Mobile

- Four separate codebases
- Different languages/teams/bugs
 - maintenance nightmare!
- Different release cycles
- 4x the engineering cost



Google
Developer
Groups

Workarounds

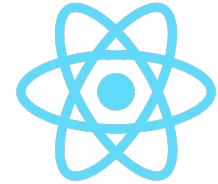
- **Cordova:** Web apps in native shells
- **Electron:** Run Chromium + Node.js as desktop apps
- **React Native:** Use web principles to build native mobile



APACHE
CORDOVA™



ELECTRON








React Native



Google
Developer
Groups

The feedback loop kicked in:

1. Better browser APIs shipped 
2. More sophisticated apps 
3. Users expected MORE web apps to be capable 
4. Demand pushed browsers to add more capabilities 
5. Repeat 



The web stack won the technology war

Now, the browser itself is becoming the most capable delivery mechanism.



Google
Developer
Groups

WHAT AN OS PROVIDES

BROWSER EQUIVALENT

Access to hardware, CPU, GPU

Web Workers, WebGPU, MediaStream

Resource management & isolation

Service Workers, Origin-based security

File system & Persistence

File System Access API, IndexedDB

Installation and updates

Progressive Web Apps (PWAs)

System engagement

Push API, Notifications API, Badging API



Google
Developer
Groups

Execution & Performance

- **Web Workers:**
True parallel threads. No more UI freezing.
- **WebGPU:**
Direct access to the GPU for 3D rendering and compute tasks.
- **WebAssembly:**
Run C++, Rust, or Go at nearnative speeds.



Google
Developer
Groups

Storage & Persistence

- **LocalStorage:**
Small, not perfect, but already made many things possible.
- **File System Access API:**
Open, edit, and save files directly to the user's disk.
- **IndexedDB:**
Store gigabytes of structured data client-side.



Deep Hardware and Device Access

- **Camera, microphone:**
MediaStream API—video calls, content creation, AR experiences
- **Sensors:**
Accelerometer, gyroscope, ambient light
- **Bluetooth, USB:**
Directly communicate with external devices. IoT applications.



Google
Developer
Groups

Security & Permissions Model

- **Permission prompts:**
Users grant or deny camera, location, notifications
- **Origin-based isolation:**
Different websites cannot access each other's data.
This is process isolation at the web level.



Google
Developer
Groups

Background Execution

Service Workers run independently of your web page. They survive when you close the tab. They can:

- Intercept network requests
- Sync data when connectivity returns (Background Sync)
- Run periodic tasks (Periodic Sync)
- Receive push notifications from servers



Google
Developer
Groups

What Browsers Still Lack

- Performance Ceiling
- Hardware Access is Limited
- Battery and Resource Efficiency
- System Integration



Why Are These Hard to Solve?

- Security > Raw power
- Cross-platform > Optimization
- User consent > Convenience
- Open standards evolve slower



Why should we care?



Google
Developer
Groups

Architecture Decisions

Where Should Computation Happen?



Google
Developer
Groups

When Client-Side Makes Sense

- **Real-time interactions:**
Image editing, 3D apps, anything that needs instant feedback with zero latency.
- **Privacy-sensitive operations:**
Processing user data locally means it never leaves their device.
- **Reducing server costs:**
Offloading work to the client can dramatically reduce infrastructure bills.



Google
Developer
Groups

When Cloud/Server-Side Makes Sense

- **Consistency and Security:**
Complex business logic, API keys, and proprietary algorithms should remain server-side.
- **Data aggregation:**
When you need to process data from multiple users.
- **Heavy computation:**
Training ML models, processing terabytes of data.
Don't drain the user's battery!



Network Awareness

- **Optimistic updates:**
Update the UI immediately, sync in the background
- **Conflict resolution:**
What happens when offline changes conflict with server state?
- **Background sync strategies**



Accessibility: **Native Standards**

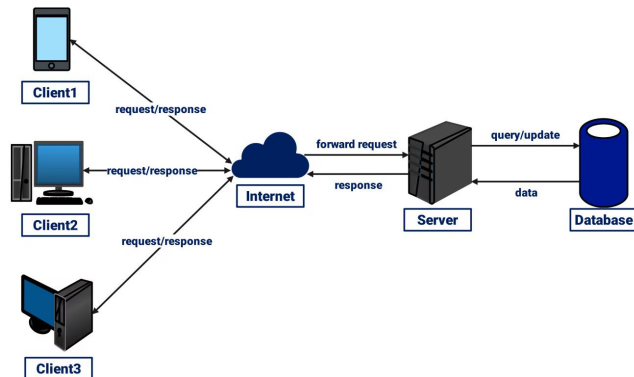
- **Keyboard navigation** must be complete and intuitive
- **Screen readers** must understand your complex UIs
- **Users with motor impairments** must be able to interact efficiently
- **Color contrast, text size, animations**—all need to consider various disabilities
- **Respect the host system's accessibility settings**



Google
Developer
Groups

Security: Hostile Environment

- **XSS:** Sanitize all inputs.
- **Dependencies:** Audit npm packages.
- **HTTPS:** Mandatory for all sensitive APIs.
- **Validation:** Have validation on **server-side**.



Web Model:

"We control nothing. Assume compromise."



Google
Developer
Groups

Developer Skillset

- **Performance Profiling:**
Learn to use browser DevTools deeply.
- **Systems Thinking:**
Understand how browsers work under the hood
- **Debugging Complex Applications:**
Learn to debug Service Workers, Web Workers, and complex async operations.



Google
Developer
Groups

Reality Check:
Always be **pragmatic!**



Google
Developer
Groups

Choose the Right Tool

 Use all these new capabilities

 Understand what's possible



Google
Developer
Groups

Thank you! 😊



diyorbek.blog