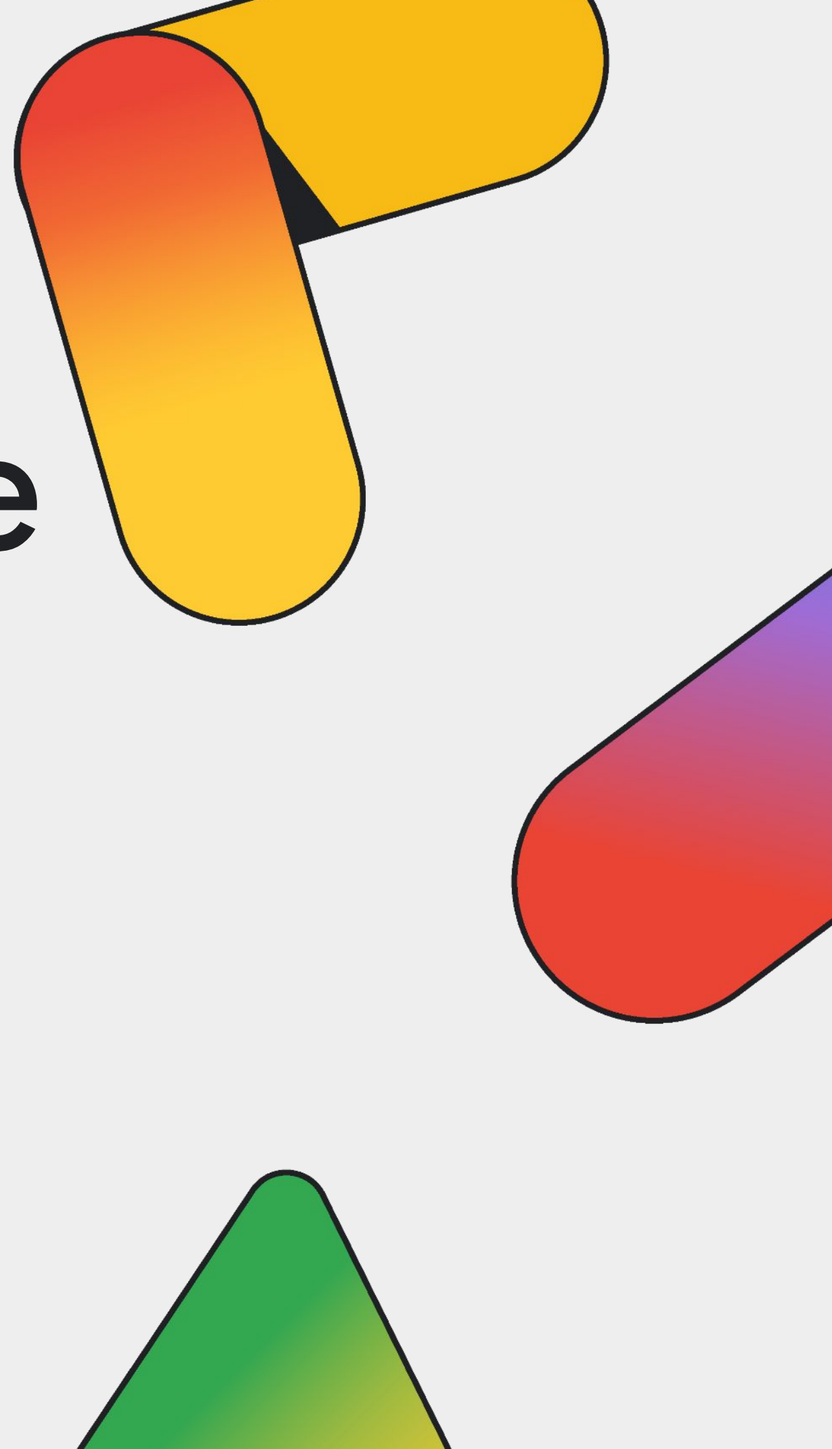
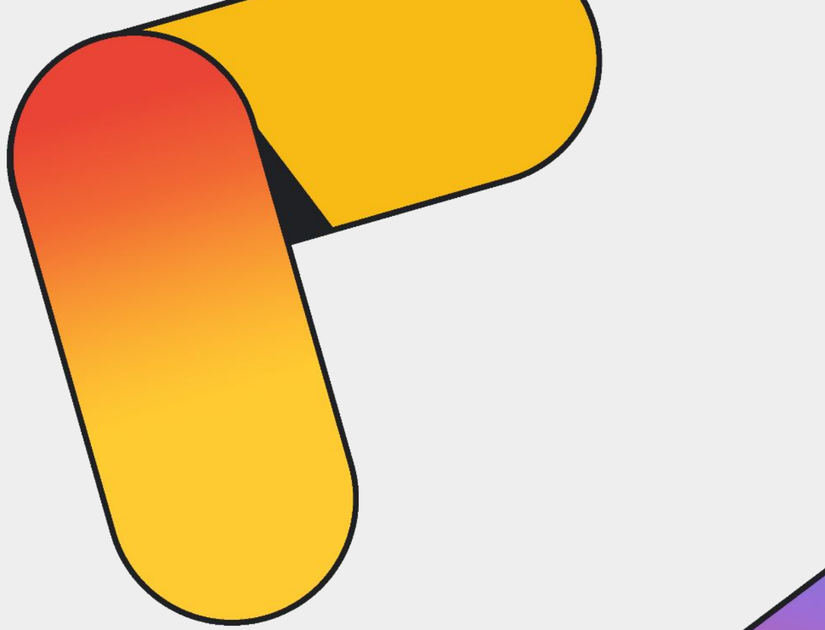
Google I/O Extended

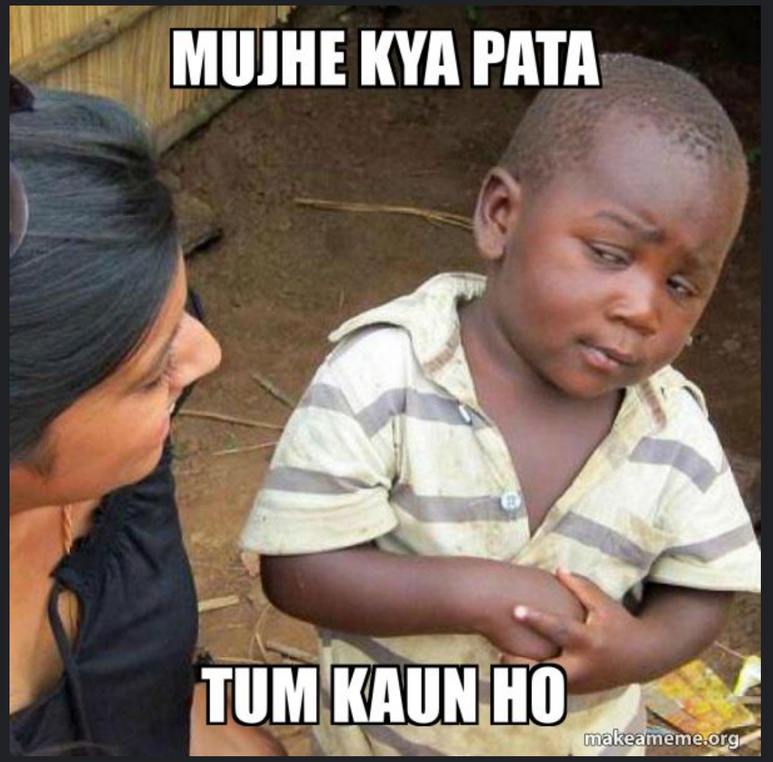
Navigating the Future of Web: Insights for Google I/O 2024



Google I/O Extended



Google I/O Extended



Who am I?

Debajit Mallick

- Software Engineer @P360
- Organizer @GDG Siliguri
- LinkedIn 2024 Top Voice for Web Development.
- Mentor and Judge of Hack4Bengal 3.0 and 2.0 Hackathon.
- Mentor of team OrganiCod3rs, the Winner of Smart India Hackathon 2022, Software Edition.
- Mentor of GirlScript Summer of Code 2023.
- Member of Team Delenitors, Smart India Hackathon 2020 Winner, Software Edition.
- Top Contributor of GirlScript Winter of Code 2021.
- Top Contributor of JGEC Winter of Code 2020

Google I/O Extended

Google I/O Extended

Table of Contents

Google I/O Extended

Google I/O Extended

Topic 1

Baseline

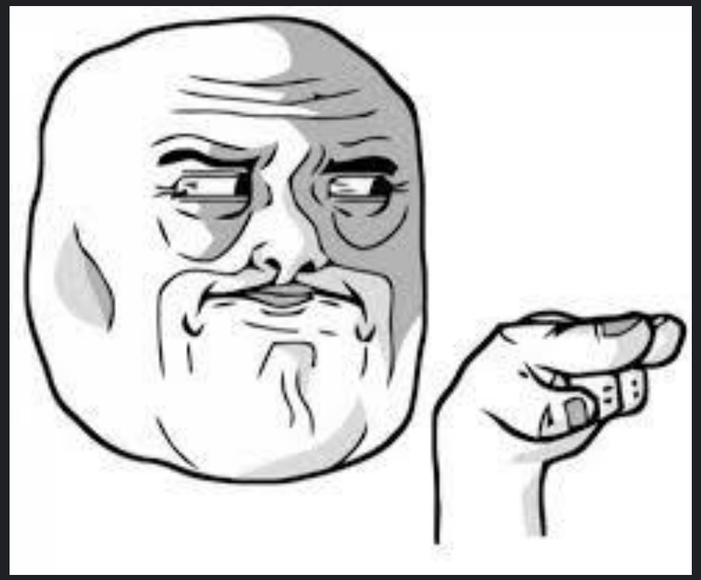
Topic 1

Baseline

Web Platform Baseline gives you clear information about which web platform features are safe to use in your projects today. Information about which web platform features are safe to use in your projects today.

The Chrome team is collaborating with other browser engines and the web community to bring more clarity. This includes our work on projects like **Interop 2023** which helps to improve interoperability of a set of key features.

Google I/O Extended



Every New Property of Browser

Baseline

Google I/O Extended

Topic 2

Speculation Rules API

Topic 2

Speculation Rules API

This new API aims to speed up web navigation by pre-rendering and pre-fetching content, allowing near-instant page transitions.

Think *milliseconds* instead of seconds.

Best of all? It only requires a few lines of code to get started, and AI can be used to intelligently predict navigation patterns.

Google I/O Extended

Everytime I visit a website

Topic 2

Key Features of Speculation Rules API

- **Pre-fetching and Pre-rendering:** The API allows browsers to pre-fetch resources like HTML, CSS, and JavaScript for upcoming pages based on user behavior or navigation patterns.

- **AI Integration:** The AI helps determine which links are most likely to be clicked and pre-renders those, thereby optimizing resource usage and reducing unnecessary pre-fetching.

- **Ease of Use:** Developers can implement the API with just a few lines of code, which makes it an attractive option for those looking to enhance the performance of their websites without significant overhead.

index.html

```
<link rel="speculationrules" href="next.html"
          content='{
              "prerender": [{"source": "list", "urls": ["next.html"]}],
              "prefetch": [{"source": "list", "urls": ["styles.css", "script.js"]}]
          }'
>
```

Google I/O Extended

Topic 3

Gemini Nano in Chrome

Topic 3

Gemini Nano in Chrome

Now Gemini Nano is embedded in Chrome, offering on-device features like “Help Me Write”.

It helps us to generate short form of content such as reviews and social media posts with a focus on privacy and offline functionality.

Google I/O Extended

Gemini Nano Be like

Topic 3

Key Features of Gemini Nano on Chrome

- **On Device AI Processing:** Unlike traditional cloud-based models, Gemini Nano runs directly on users' device. Which reduces latency a lot.

- **"Help me Write" Feature:** This assist users to draft short-form of content on device using AI.

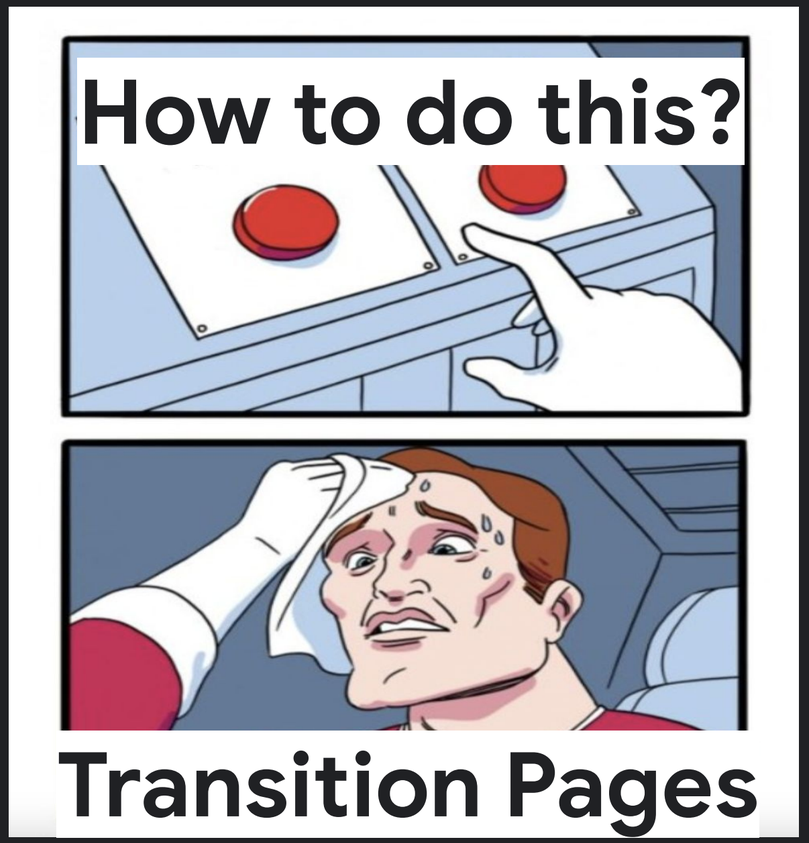
- **Privacy and Offline Availability:** It operates on-device, which ensures better privacy. Also, offline feature is great for places with limited access of internet or no internet places.

Google I/O Extended

Topic 4

View Transitions for Multi-Page Apps

Google I/O Extended



home.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        .view-transition-old {
            opacity: 1;
            transition: opacity 0.5s ease-out;
        }

        .view-transition-new {
            opacity: 0;
            transition: opacity 0.5s ease-in;
        }
    </style>
</head>
<body>
    <h1>Welcome to the Home Page</h1>
    <p>This is the home page content.</p>
    <a href="about.html" id="about-link">Go to About Page</a>

    <script>
        document.getElementById('about-link').addEventListener('click',
function(event) {
            event.preventDefault();
            document.startViewTransition(() => {
                window.location.href = event.target.href;
            });
        });
    </script>
</body>
</html>
```

about.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        .view-transition-old {
            opacity: 1;
            transition: opacity 0.5s ease-out;
        }

        .view-transition-new {
            opacity: 0;
            transition: opacity 0.5s ease-in;
        }
    </style>
</head>
<body>
    <h1>About Us</h1>
    <p>This is the about page content.</p>
    <a href="home.html" id="home-link">Go back to Home</a>

    <script>
        document.getElementById('home-link').addEventListener('click', function(event)
{
            event.preventDefault();
            document.startViewTransition(() => {
                window.location.href = event.target.href;
            });
        });
    </script>
</body>
</html>
```

Google I/O Extended

Topic 5

Dialog Element

Topic 5

Key Features of Dialog Element

- **Focus Management:** As it is a native HTML element. So focus management is very much easy.

- **Keeping the Stacking Context:** It’s very easy to handle the stacking context for multiple dialog elements.

- **Easy to implement:** Previously it is pretty hard to implement dialog in html. Now it is pretty easy to implement using the dialog element with less JS and CSS.

Google I/O Extended

index.html

```
<dialog id="d">
    <form method="dialog">
        <p>I'm a dialog</p>
        <button>ok</button>
    </form>
</dialog>

<button onclick="d.showModal()">
    Open Dialog
</button>
```

Google I/O Extended

Topic 6

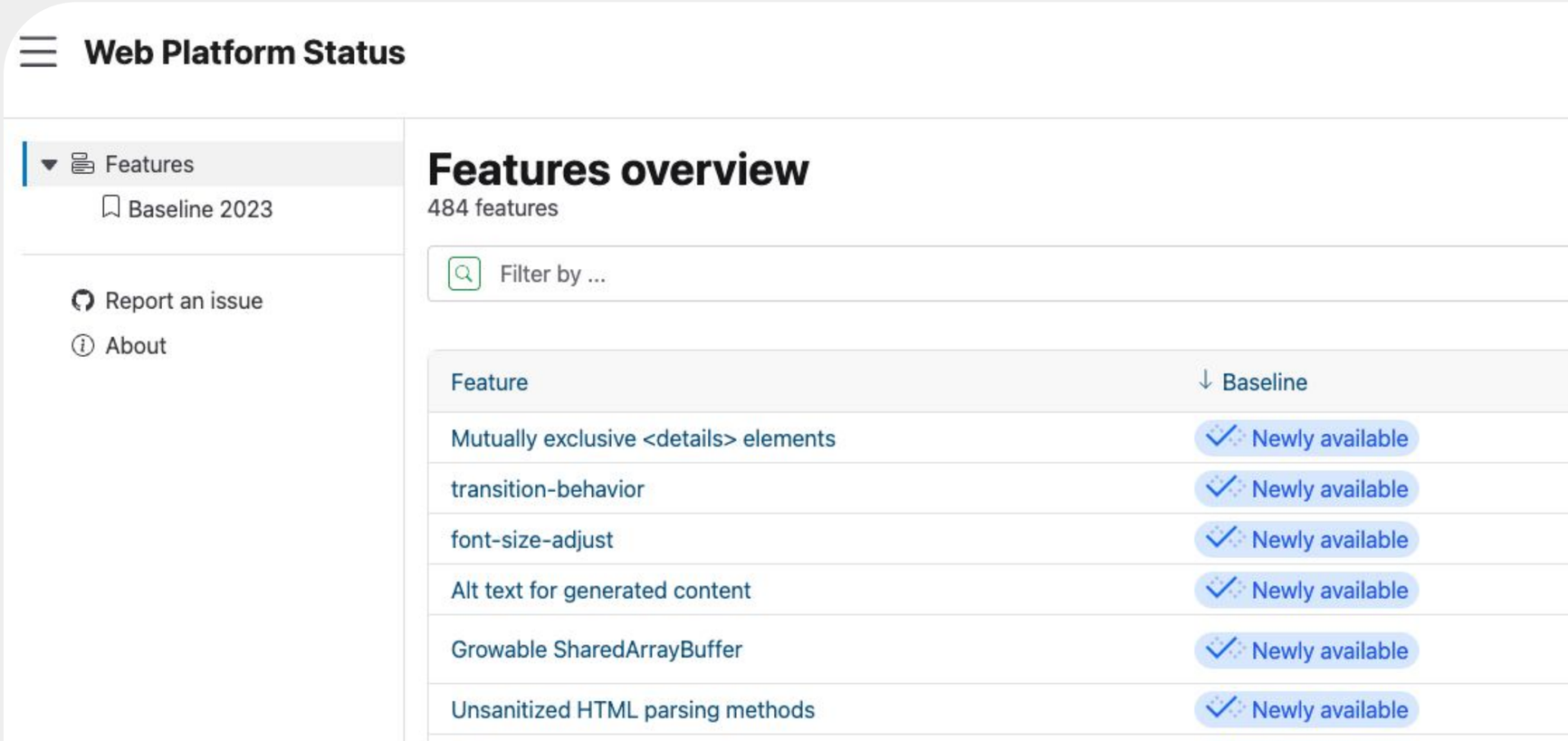
Web Platform Dashboard

Topic 6

Web Platform Dashboard:

It is a tool to simplify the process of tracking web platform changes, API updates, and cross-browser compatibility, which provides a unified view of the web technologies.

Check the dashboard:
https://webstatus.dev/



Google I/O Extended

Topic 6

Key Features of Web Platform Dashboard

- **Unified View of Web Features:** The Web Platform Dashboard provide developers with a consolidated view of the web platform.

- **Integration with Baseline:** The dashboard is integrated with the Baseline Project, so users can see which feature is available for which browsers.

- **Developer Workflows:** Developers can integrate the dashboard in their workflows.

Google I/O Extended

Topic 7

Web GPUs

Topic 7

Web GPUs:

Web GPUs are to improve the web performance and enabling more powerful use cases on web.

The updates on Web GPUs aim to make web apps more capable, particularly in areas like AI processing, graphics rendering and high performance computing.

Google I/O Extended

New Web GPUs

Old Web GPUs

Topic 7

Key Features of Web GPUs

- **On-Device AI:** By leveraging the power of Web GPUs, AI models can run on device directly on users browser.

- **Cross Platform Support:** Web GPU enables developers to write GPU accelerated code that works across a wide range of devices.

Google I/O Extended

Connect with Me

Google I/O Extended

Thank You

Debajit Mallick

Software Engineer @P360