Ditch Your JavaScript For CSS!

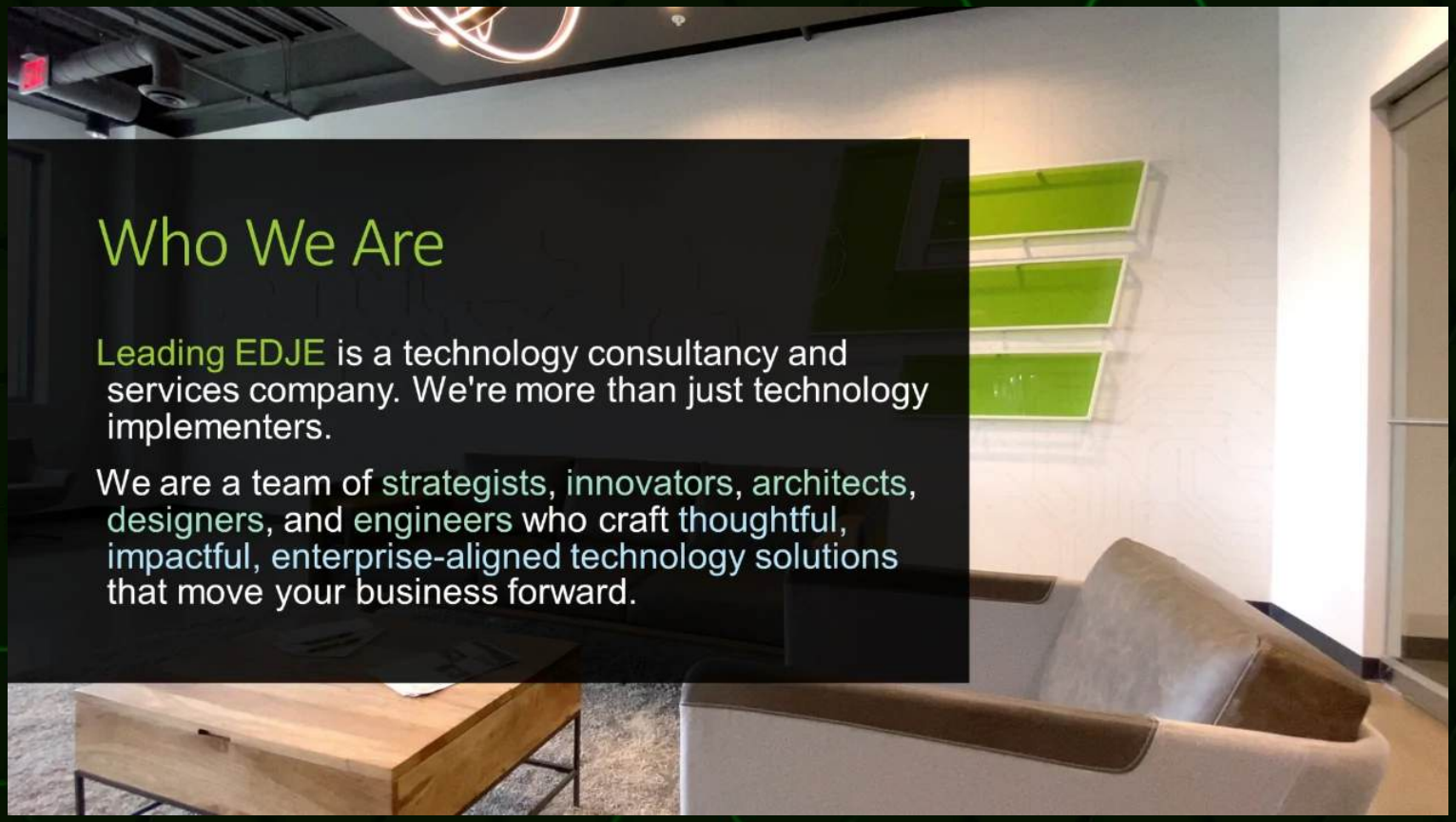
Bob Fornal

Entrepreneur
Code-squid provides solid, in-depth frontend training that is supported with real-world code projects. Blessed husband and proud father of two.

https://code-squid.com

Senior Solutions Developer
Leading EDJE, Inc.
Passionate about learning, testing, mentoring, speaking, and personal growth.

https://leadingedje.com


Who We Are
Leading EDJE is a technology consultancy and services company. We're more than just technology implementers.
We are a team of strategists, innovators, architects, designers, and engineers who craft thoughtful, impactful, enterprise-aligned technology solutions that move your business forward.

I ditched JavaScript for Performance!

JavaScript

Main thread; can lead to performance bottlenecks if complex calculations or frequent DOM manipulations are involved, potentially causing jank or stuttering.

While offering immense flexibility for complex, dynamic animations and interactions, this dynamism comes with the overhead of runtime calculations and potential layout recalculations.

Achieving optimal performance with JavaScript animations often requires careful implementation, including techniques like `requestAnimationFrame` to synchronize with browser repaint cycles and avoid unnecessary work.

CSS

Browsers are highly optimized to parse and render CSS, leveraging native rendering engines and often offloading CSS animations to the GPU, leading to smoother performance and reduced strain on the main thread.

CSS is declarative, meaning you describe the desired end state, and the browser handles the transitions. This allows for efficient pre-calculation and optimization.

CSS typically does not block the main thread during rendering, allowing for a more responsive user experience, especially during page load.

Should I keep JavaScript? Maybe

JavaScript

For highly interactive elements, game-like experiences, or animations requiring intricate logic, user input, or data manipulation, JavaScript offers the necessary control and flexibility.

The Web Animations API provides a more performant way to create JavaScript-based animations, allowing for greater control while leveraging browser optimizations.

When animations need to be tightly integrated with other browser APIs or data sources, JavaScript is the natural choice.

Summary

For simple or intermediate transitions, visual effects, and static layout changes, CSS is generally the more performant and efficient choice.

For complex, dynamic animations or interactions requiring intricate logic and control, JavaScript becomes necessary, but careful implementation is crucial for optimal performance.

The "rule of least power" suggests using the simplest technology capable of achieving the desired outcome, often favoring CSS when possible.


TIME FOR QUESTIONS