How to Improve the Perceived Performance of Features?

freecodecamp.org/learn/front-end-development-libraries-v9/lecture-understanding-performance-in-web-applications/how-to-improve-the-perceived-performance-of-features

As we know, perceived performance isn't just about how fast a site actually is — it's about how fast it feels to users. So, for example, if you have ever used a website that felt slow, even though it wasn't? That's perceived performance — how fast a site feels to the user.

Small tweaks can make a huge difference in the user experience. Let's explore some practical ways to improve perceived performance!

Spinners can make users feel like they're waiting longer than they actually are. Instead, try skeleton screens — gray placeholders that mimic real content and gradually load in:

```
<div class="skeleton"></div>
<script>
  setTimeout(() => {
    document.querySelector(".skeleton").innerHTML = "✅ Content Loaded!";
  }, 2000);
</script>
```

Users feel like the page is loading progressively, rather than waiting in the dark.

In addition to this, users shouldn't have to wait for everything to load before they can start interacting. Load important content first, then load the rest in the background. We looked at this previously with lazy loading:

```
<img src="placeholder.jpg" data-src="real-image.jpg" class="lazy">
<script>
  document.addEventListener("DOMContentLoaded", function () {
    const lazyImages = document.querySelectorAll("img.lazy");
    lazyImages.forEach(img => {
      img.src = img.dataset.src;
    });
  });
</script>
```

This makes the page feel responsive, even if not everything has loaded yet.

Now - what if your site could predict what the user will click next? With preloading, you can fetch the next page before they request it:

```
<link rel="preload" href="next-page.html" as="document">
```

This reduces perceived loading time to near zero when users navigate.

And finally, users don't like uncertainty. When they click a button, acknowledge their action instantly:

```
<button onclick="this.innerText='⏳ Processing...'; setTimeout(() => this.innerText='✅
Done!', 2000);">
  Click Me
</button>
```

This means that, even if there's a delay, the user feels like something is happening.

So, in conclusion, improving perceived performance isn't just about raw speed, it's about how speed feels to the user. Use skeleton screens, prioritize important content, preload smartly, and give instant feedback to make your site feel faster.

Questions

What is the perceived performance?

How fast a site feels to the user.

Correct!

How fast a page loads.

How slow a page loads.

How fast a users performance is.

Which of the following is an example of preloading?

Fetching the next page before a user requests it.

Correct!

Fetching a webpage's content after a user interacts with the page.

Displaying placeholder content while waiting for resources to load.

Caching assets for offline use after the page has loaded.

What is an example of a skeleton screen?

A screen displaying an image of a skeleton.

The bare bones of a web page.

A gray placeholder that mimics real content and gradually loads in.

Correct!

A loading screen that shows simple shapes or text as the page content loads.

Navigated to How to Improve the Perceived Performance of Features?