



KEEPING SSR COOL WITH REACT HYDRATION

MONICA POWELL

React Rally 2020

 www.monica.dev |  @waterproofheart

HI, I'M MONICA



I'm a software engineer who enjoys building technology that elevates people whether that's increasing access to e-books, creating tools for Meetup's community organizers or sending curated career opportunities to diverse job-seekers at scale. I'm also passionate about making open-source more accessible and recently became an inaugural

GitHub Star

www.monica.dev | [@waterproofheart](https://twitter.com/waterproofheart)



OVERVIEW

The purpose of this talk is to share some helpful things to keep in mind to render a seamless experience as a Server-Side Rendered (SSR) site transitions from a window-less (server) environment to a browser.

SSR apps tend to have faster initial loading Times and better SEO than client-side only apps.

 www.monica.dev |  @waterproofheart

Server-side rendering can be powerful but it does require thinking in multiple contexts and I want to share some of the gotchas I've run into while developing Server-Side Rendered websites.

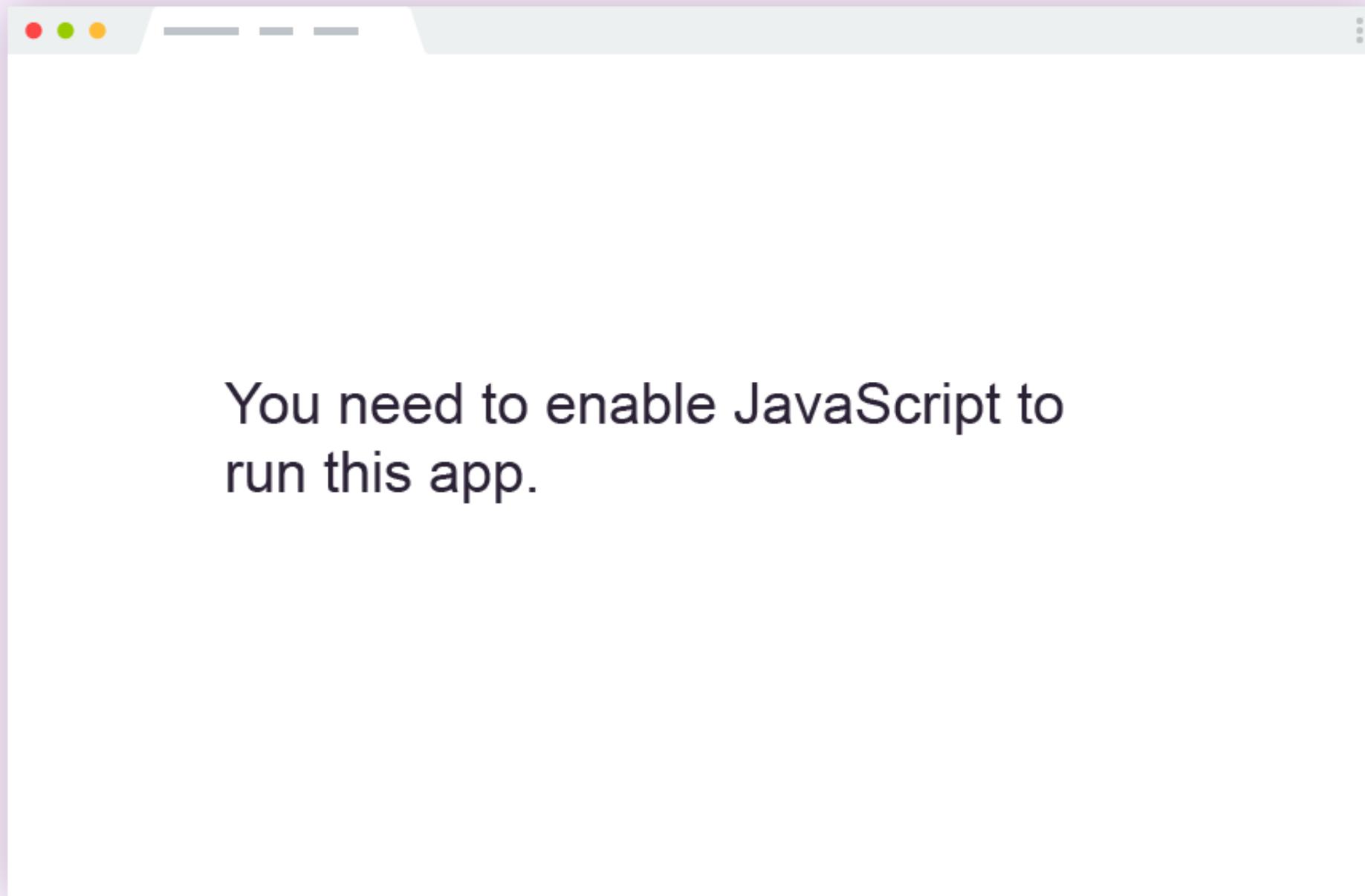
WHAT IS SERVER-SIDE RENDERING (SSR)?

A server generates the initial HTML that loads in a browser from JavaScript. Frameworks like NextJS and GatsbyJS support SSR out-of-the box

 www.monica.dev |  @waterproofheart

There are multiple types of Server-Side Rendering for example SSR can be used to render every single page request or only the initial page request. NextJS offers more robust server-side support than Gatsby. In contrast, you may be familiar with Create React App which does NOT come with SSR functionality out of the box.

WHAT IS CLIENT-SIDE RENDERING?



 www.monica.dev |  @waterproofheart

With Client-Side Rendering
you must have HTML enabled
in order to view content on the
site.

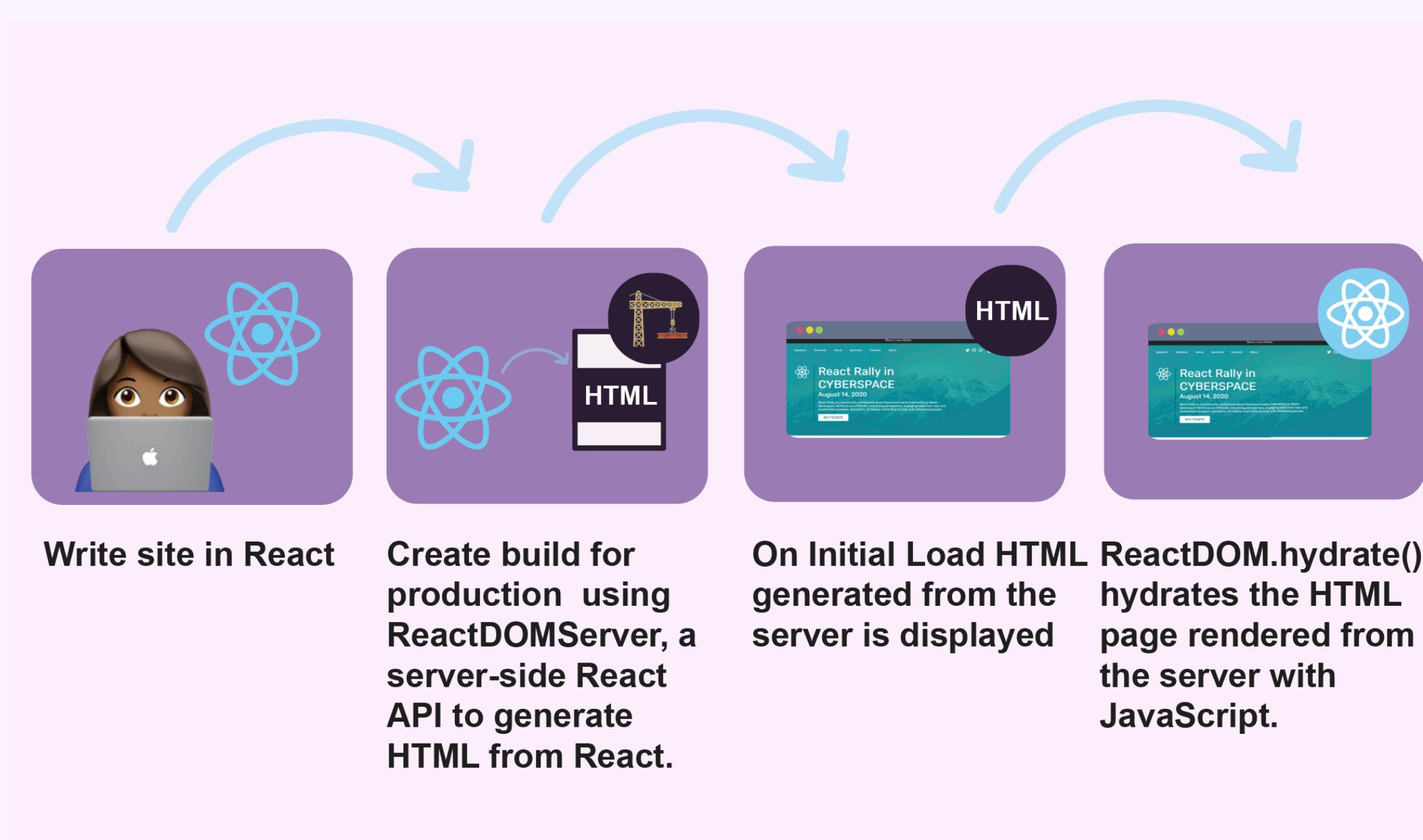
CLIENT-SIDE INITIAL DOM

```
html>
<head><!-- SEO/Metadata here --></head>
<body>
  <div>You need to enable JavaScript to run this app.</div>
  <div id="root"></div>
  <script><!-- all of the JavaScript --></script>
  <script src="/static/js/2.6158a3d8.chunk.js"></script>
  <script src="/static/js/main.ba831a9f.chunk.js"></script>
</body>
</html>
```

monicadev | @waterproofheart

If you look at the DOM on an a Create React App you'll notice very little HTML markup in the DOM. You'll see the root where React is injected, a message saying you need to enable JavaScript to run the app as well as script tags that link to the JavaScript that needs to be loaded to hydrate the page.

OVERVIEW OF SSR (IN STATIC CONTEXT)



monicadev | @waterproofheart

Write site in React  -> Gatsby creates a production build of your site using ReactDOMServer, a React server-side API to generate HTML from React. -> Someone visits your website and the HTML generated from the server is displayed on the page -> ReactDOM.hydrate(), hydrates the HTML page that was rendered from the server with JavaScript -> React reconciler APIs take over

TOGGGLING JAVASCRIPT

NICA POWELL

Software engineer, content creator &
Community organizer



Hi! I'm a software engineer who is passionate about making open-source more accessible, technology to elevate people, and building community. You can find me teaching web development on [Egghead](#), contributing to open-source projects like [NYPL-Simplified](#) and [Gatsby](#), and co-leading [React Ladies](#), a community I founded for women and non-binary React developers.

Writing

[View All Writing](#)

How To Create A GitHub Profile README

Exploring Art Direction With Gatsby-Image

Potential Thoughts On Migrating from gatsby-transformer-remark to gatsby-plugin-mdx

Unleash Your CSS Superpowers with CSS Variables



▶ GatsbyJS vs. Create React App
Grey's Anatomy
Lorem Ipsum Generator

How many paragraphs should be generated?

4

[Generate Lorem Ipsum](#)

Grey's Anatomy Lorem Ipsum Generator

By: Monica Powell • [View Code](#)



[www.monica.dev](#)



[@waterproofheart](#)

Copyright to "Grey's Anatomy" is held by various outside entities and is provided here for educational purposes only.

Let's take a look at what happens when JavaScript is enabled or disabled in a Create React App or Gatsby App

The screenshot shows two side-by-side web pages. On the left is Monica Powell's personal website, featuring a dark header with her name, a bio about being a software engineer and content creator, and a portrait of her holding a laptop. Below this are sections for writing, with a button to 'View All Writing'. On the right is a tool titled 'Grey's Anatomy Lorem Ipsum Generator', which asks for the number of paragraphs (set to 4) and has a red 'Generate Lorem Ipsum' button.

Monica POWELL

Software engineer, content creator & community organizer

Monica! I'm a software engineer who is passionate about making open-source more accessible, using technology to elevate people, and building community. You can find me teaching web development on [Egghead](#), contributing to open-source projects like [NYPL-Simplified](#) and [Gatsby](#), and co-founding [React Ladies](#), a community I founded for women and non-binary React developers.

Writing

[View All Writing](#)

How To Create A GitHub Profile README

Exploring Art Direction With Gatsby-Image

Potential Thoughts On Migrating from gatsby-transformer-remark to gatsby-plugin-mdx

Give Your CSS Superpowers with CSS Variables

www.monica.dev |  @waterproofheart

Grey's Anatomy Lorem Ipsum Generator

By: [Monica Powell](#) • [View Code](#)

Copyright to "Grey's Anatomy" is held by various outside entities and is provided here for educational purposes only.

Create-React-App in contrast uses client-side rendering where the browser is responsible for constructing the initial HTML therefore, we just see the bare bones HTML as opposed to a full HTML document with initial content.

IMPORTANT NOTES

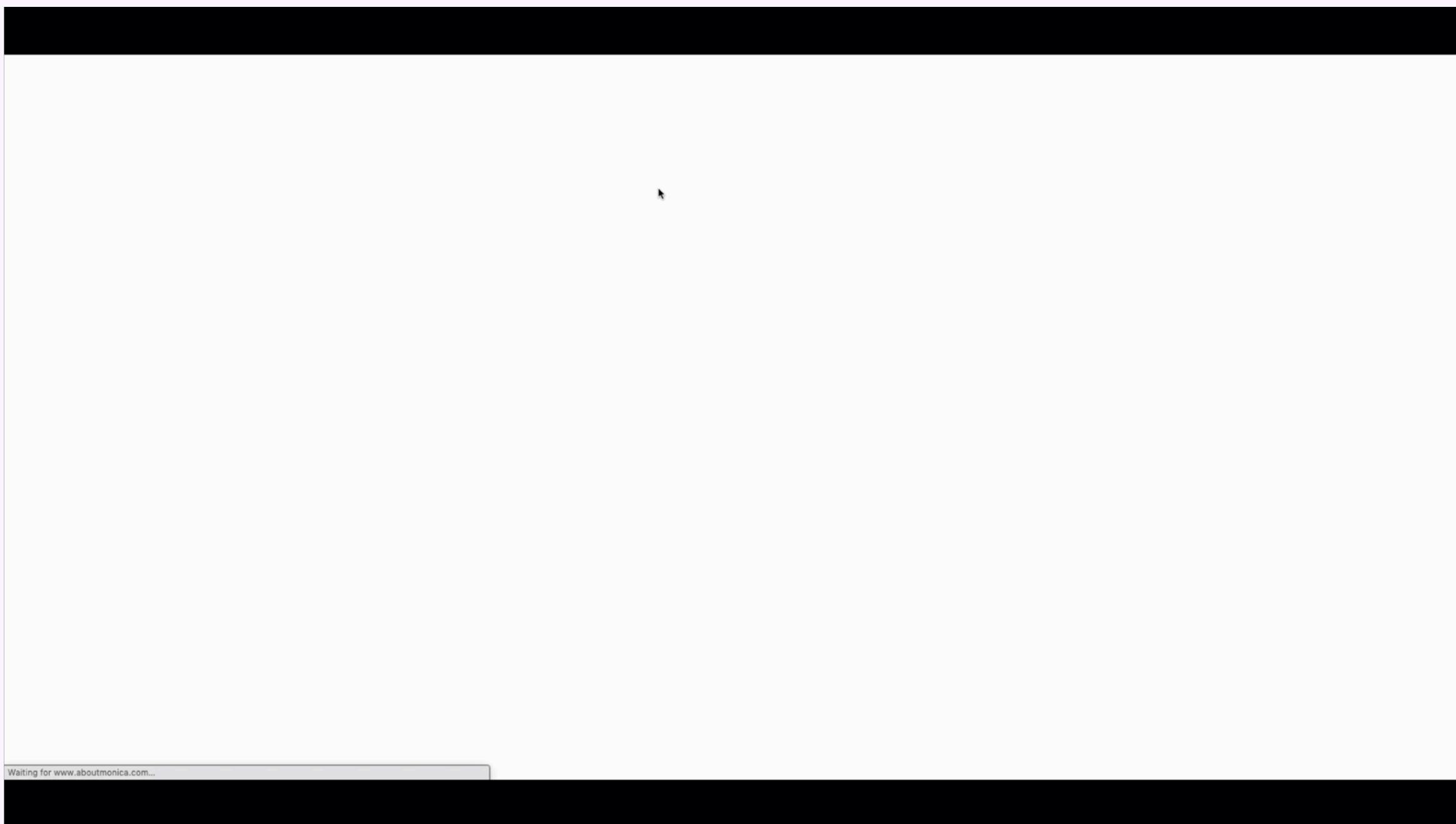
- User or browser-specific data is not available in the server when HTML for the site is generated (i.e., window size, authentication status, local storage etc)



monicadev | @waterproofheart

In this image you'll see that the store location data, my username and items in shopping cart were not available on initial page load. Patterns like this can be common on server-side rendered applications.

WHAT COULD GO WRONG? 😅



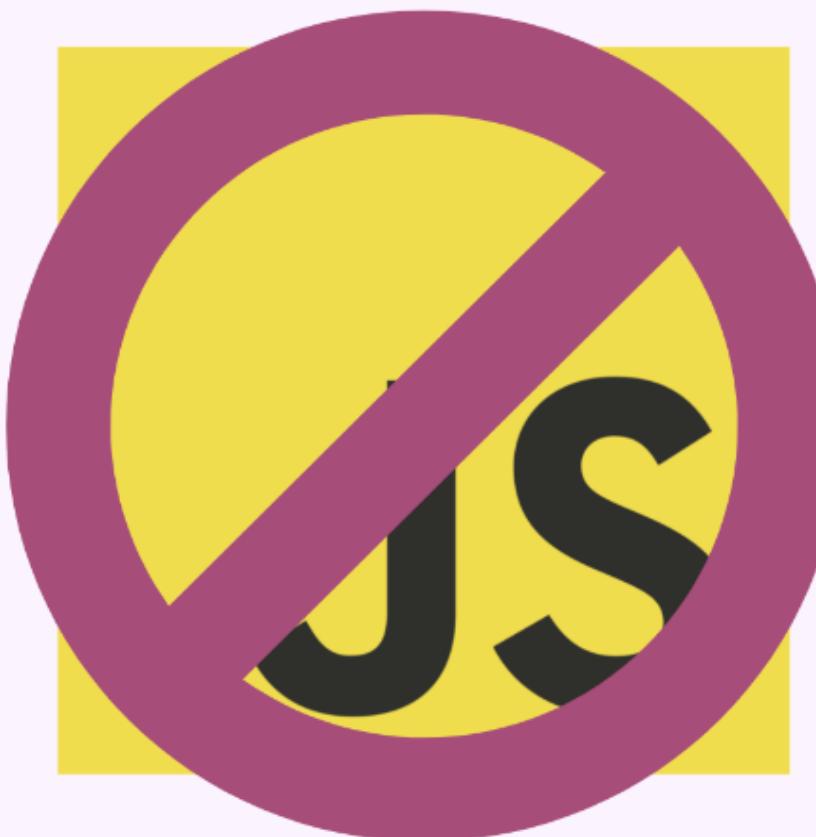
monicadev | @waterproofheart

Developing in an SSR context can introduce a new set of issues as you need to consider how the page loads **BEFORE** JavaScript is available since JavaScript is not required for the page to load.

DEBUGGING SSR HYDRATION ISSUES



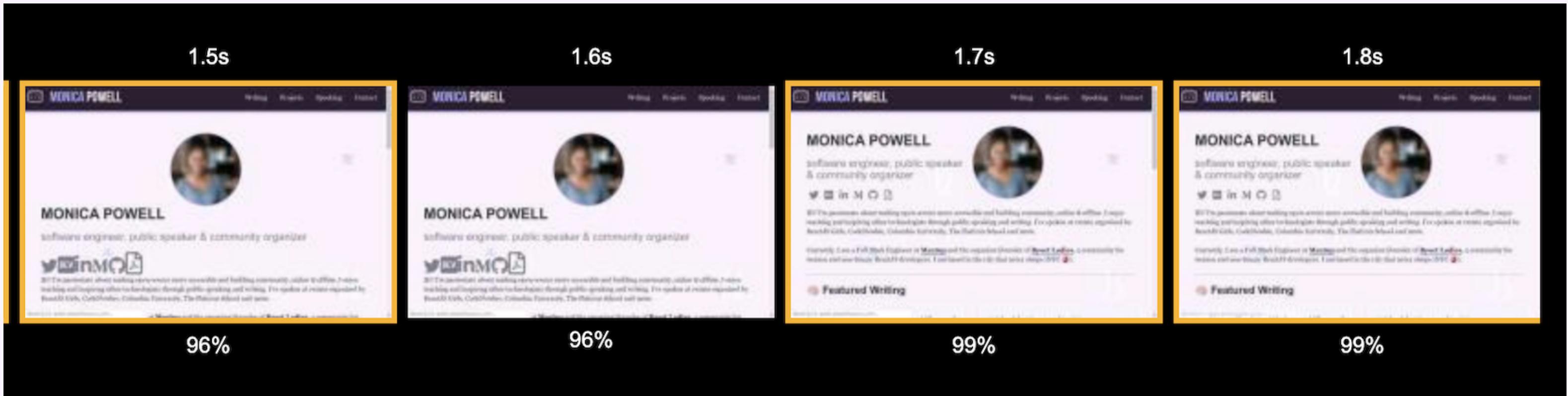
- ▶ Disable JavaScript
- ▶ View filmstrips to see how the website hydrates



 www.monica.dev |  @waterproofheart

You may notice weird changes on initial page load that change too quickly to properly inspect. But there's another way to slow down and really see what is going on. You can disable JavaScript and also use a site like webpage speed test to generate thumbnails that show you exactly how the page is loading step by step.

Before: Styled icons with CSS from Font Awesome NPM Package



👤 www.monica.dev | 🐦 @waterproofheart

Here's a waterfall I took of the issue on my site. You can see one of the issues is that the size of the icons changes drastically between 96% and 99% loaded which can be a disjointing experience.

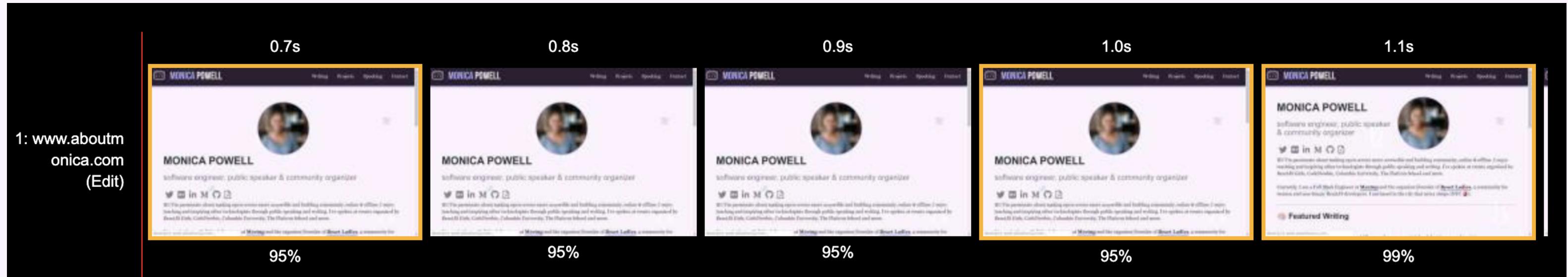
SOLUTION

- ▶ Replicate the final styling with local CSS without relying on FontAwesome's external CSS which required JavaScript to be applied

 www.monica.dev |  @waterproofheart

Once I disabled JavaScript, I was able to discover that the ways the icons look before they were fully loaded mirrored the app without JavaScript, I determined font awesome was using its own styling that was coming in through JS. In order to resolve the issue I disabled font awesome's external CSS since it's loaded via JavaScript and replicated the css I wanted locally.

After: Styling icons locally and disable Font Awesome's CSS



monicapowell | @waterproofheart

Now you'll notice that the styling of the icons is consistent as the application loads

IMMUTABLE LAYOUT

- ▶ Avoid unnecessary layout shifts during page load
- ▶ Implement layouts with placeholder/gap for expected client-side content
- ▶ Use CSS instead of JS to style the page

 www.monica.dev |  @waterproofheart

Another culprit in the previous example was using JS to position content instead of media queries. CSS loads before JS and is less resource-intensive. This can be a common issue when loading the page as there may still be some data unavailable as the page loads. You can try to design around this by leaving room for data to come in, for example in the Target nav there was no shift as the user/store specific data loaded. You should also use CSS instead of JavaScript for styling ;)

Learn more at: https://www.speedpatterns.com/patterns/immutable_layout.html

"JavaScript is a powerful language that can do some incredible things, but it's incredibly easy to jump to using it too early in development, when you could be using HTML and CSS instead..."

-- Iain Bean, Your blog doesn't need a JavaScript framework

...Consider the rule of least power: Don't use the more powerful language (JavaScript) until you've exhausted the capabilities of less powerful languages (HTML)."

-- Iain Bean, Your blog doesn't need a JavaScript framework

ART DIRECTION: RULE OF LEAST POWER

The screenshot shows the homepage of monica.dev. At the top, there's a dark header bar with a small circular profile picture of Monica on the left, followed by the name "MONICA POWELL" in white capital letters. To the right of the name are links for "Writing", "Projects", "Speaking", "Newsletter", and "Contact". Below the header, the main content area has a light blue background. On the left, the name "MONICA POWELL" is displayed in large, bold, dark letters. Below it is a description: "software engineer, content creator & community organizer". Underneath this description are icons for GitHub, Twitter, LinkedIn, and a video camera. On the right side of the page, there's a cartoon illustration of Monica holding a laptop. Below the illustration is a paragraph of text about her work and contributions. At the bottom left, there's a section titled "Latest Writing" with a "View All Writing" button. The overall design is clean and modern.

 www.monica.dev |  @waterproofheart

I want to walk through a time I recently used the rule of least power to make the initial page load faster and reduce relying on JavaScript to dynamically load different images on my site

ART DIRECTION: HTML OR JAVASCRIPT?

- ▶ HTML art direction can be used to dynamically load of images based on screen size using srcset attributes with `` and `<source>` instead of JavaScript.

ART DIRECTION: WHY HTML?

- ▶ Swapping images at different screen sizes can be done with JavaScript or CSS instead of native HTML attributes however using HTML can improve page loading performance as it prevents unnecessarily preloading two images

ART DIRECTION IN HTML



```
<picture>
  <source media="(min-width: 625px)" srcset="animonica-full.png" />
  <source srcset="animonica-headshot-cropped.png" />
  
</picture>
```

 www.monica.dev |  @waterproofheart

In order to set up this functionality in HTML you can use the picture attribute and set media queries on each source image. It will return the first condition the is true and as a fall back it'll return the image from the img tag.

```
import React from "react"
import { useStaticQuery, graphql } from "gatsby"
import Img from "gatsby-image"

const Avatar = () => {
  const data = useStaticQuery(graphql`query {
    mobileImage: file(relativePath: { eq: "animonica-headshot-cropped.png" })
    ) {
      childImageSharp {
        fluid(maxWidth: 300, quality: 100) {
          ...GatsbyImageSharpFluid_tracedSVG
        }
      }
    desktopImage: file(relativePath: { eq: "animonica-full.png" }) {
      childImageSharp {
        fluid(maxWidth: 300, quality: 100) {
          ...GatsbyImageSharpFluid_tracedSVG
        }
      }
    }
  }

  const sources = [
    data.mobileImage.childImageSharp.fluid,
    {
      ...data.desktopImage.childImageSharp.fluid,
      media: `(min-width: 625px)`,
    },
  ]

  return <Img fluid={sources} alt="Illustrated Monica" />
}

export default Avatar
```

monicadev | @waterproofheart

SUMMARY

- ▶ When developing in a Server-Side Rendered context it's important to consider how the page loads and what data is or is not available on initial page load.
- ▶ CSS is a better tool for styling, especially in a Server-Side Rendered app

RESOURCES

- ▶ <https://www.gatsbyjs.org/docs/react-hydration/>
- ▶ <https://joshwcomeau.com/react/the-perils-of-rehydration/>
- ▶ <https://reactjs.org/docs/reconciliation.html>
- ▶ <https://www.webpagetest.org/>

THANK YOU!

SEE YOU IN CYBERSPACE



@waterproofheart



www.monica.dev



www.monica.dev | 🐦 @waterproofheart