Navigating Google Core Web Vitals

with Gatsby and Yoast

Gatsby

Housekeeping

- ➔ This webinar will be recorded!
- ➔ We'll send an email in the next 48 hours
- ➔ Please ask questions in the **Questions** tab

Gatsby

What we're covering today

- What are Core Web Vitals
- How to Measure them
- How to Optimize for them

BORN ON

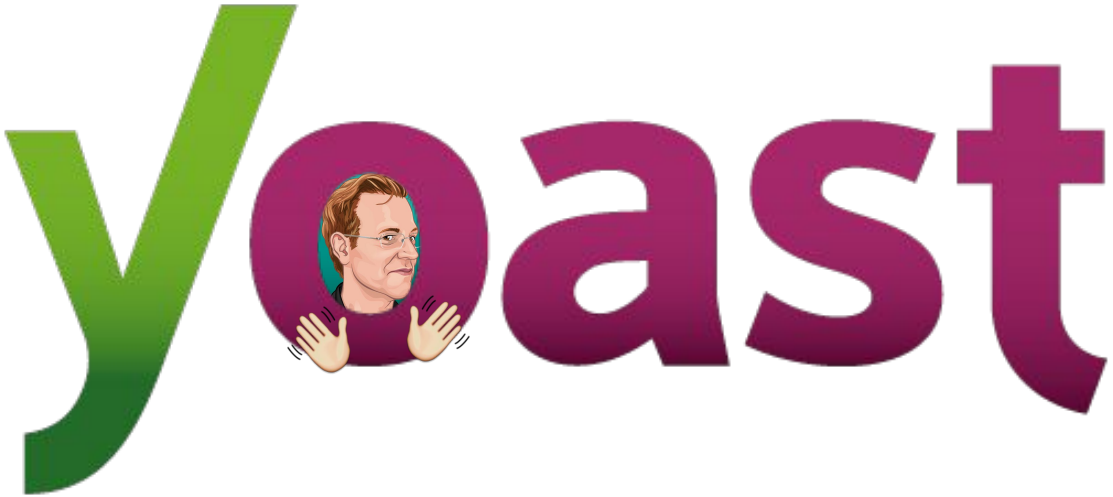
2015

DOWNLOADS

48,000,000

GITHUB STARS

50,000


yoast

Yoast SEO - REST API

In Yoast SEO 14.0 we introduced a REST API endpoint that'll give you all the metadata you need for a specific URL. This will make it very easy for headless WordPress sites to use Yoast SEO for all their SEO meta output.

There are two ways of using this: through its inclusion in the normal WP REST API responses and through our own endpoint.

Inclusion in WP REST API responses

When you're retrieving a post like so:

`https://example.com/wp-json/wp/v2/posts/1` or `http://example.com/wp-json/wp/v2/posts?slug=hello-world`, you'll receive a normal WP REST API response, with an additional field: `yoast_head`. This additional field will contain a blob with all the necessary meta tags for that page. This works for the `posts`, `pages`, `categories`, `tags` and all custom post types and custom taxonomies.

For post type archives, when you query the types endpoint the meta is included there, also on the `yoast_head` field. If it is not there, the post type does not have a post type archive enabled.

Yoast SEO REST API syntax

The syntax is very simple, you just `GET` to `/wp-json/yoast/v1/get_head?url=` with the proper URL, for example:

```
https://example.com/wp-json/yoast/v1/get_head?url=https://example.com/hello-world/
```

This will return the following:

```
{
  "head": "the complete, escaped, <head> output for Yoast SEO",
  "status": 200,
}
```

The `head` contains the complete meta output for the page. This means the Yoast SEO REST API output contains everything:

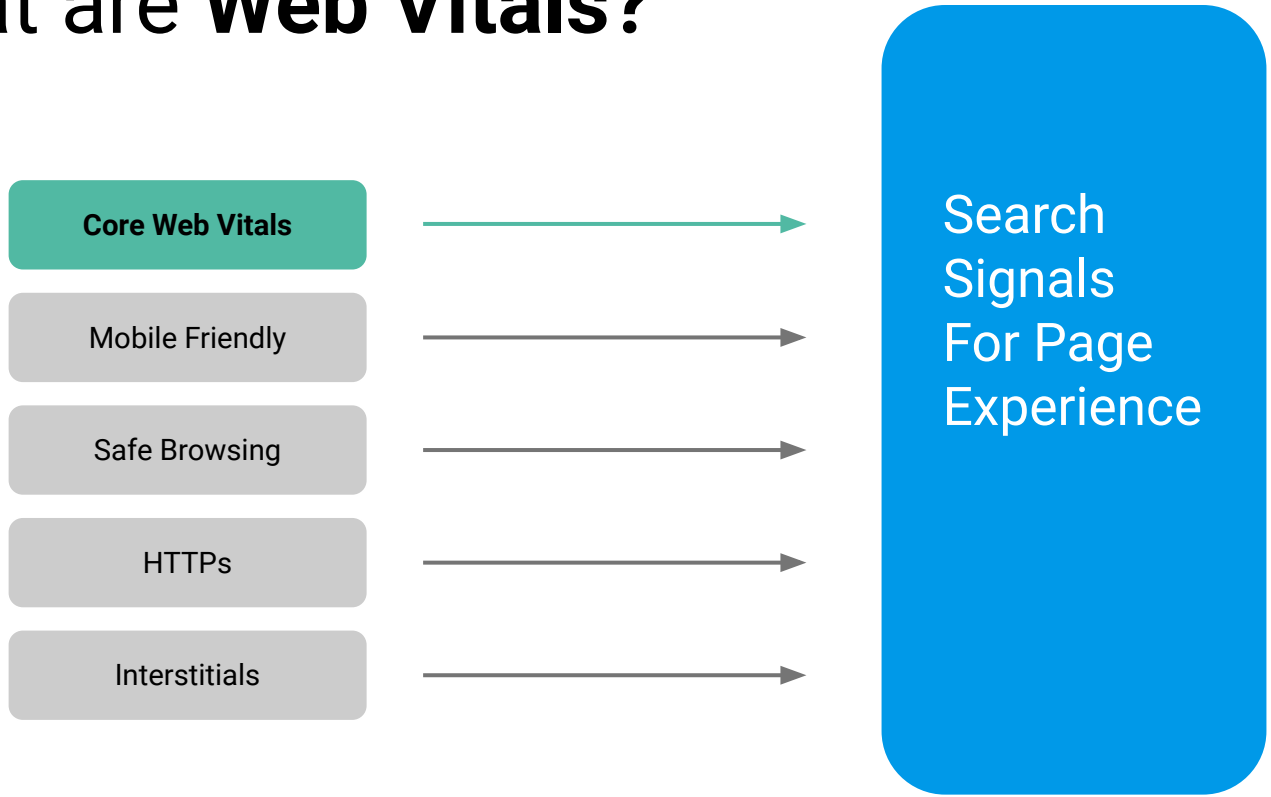
- The title
- The meta description, if you have one
- Robots meta tags
- The canonical URL
- Our Schema output
- OpenGraph meta data

Polls

Gatsby

Core Web Vitals

What are **Web Vitals?**



Gatsby

the subset of “Web Vitals” that Google is emphasizing that specifically measures the different facets of the user experience of a web page.

LCP

Largest Contentful Paint

Good	Needs Improving	Poor

FID

First Input Delay

Good	Needs Improving	Poor

CLS

Cumulative Layout Shift

Good	Needs Improving	Poor

Gatsby

Largest Contentful Paint

Are your largest images and videos rendered? Is all text visible at this point?

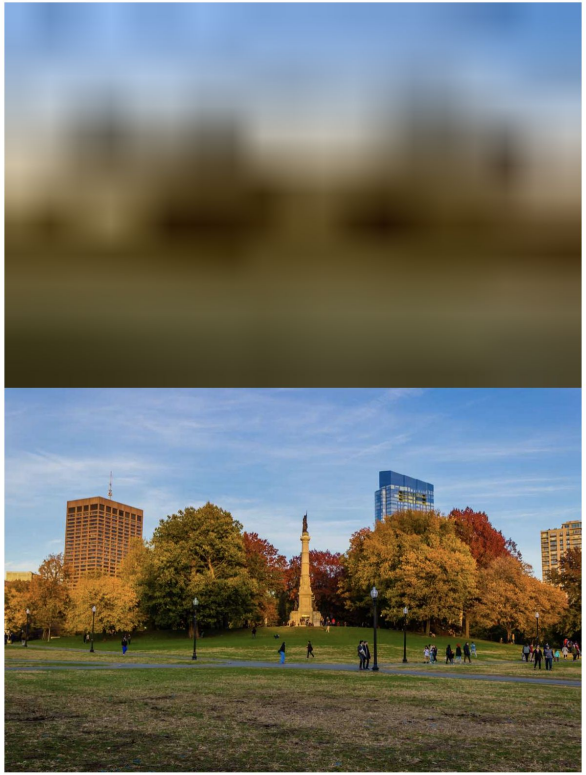
0-2.5 Seconds

2.6-4.0 Seconds

> 4 Seconds

Hello world!

Testing image



localhost:8000/static/9d7d3c45a8a4418eee11ae6c95c38ee8/33840/boston.jpg

Gatsby

First Input Delay

When a user is finally allowed to click something, how fast does your website respond? Are all scripts loaded?

0-100ms

101-300ms

> 300ms

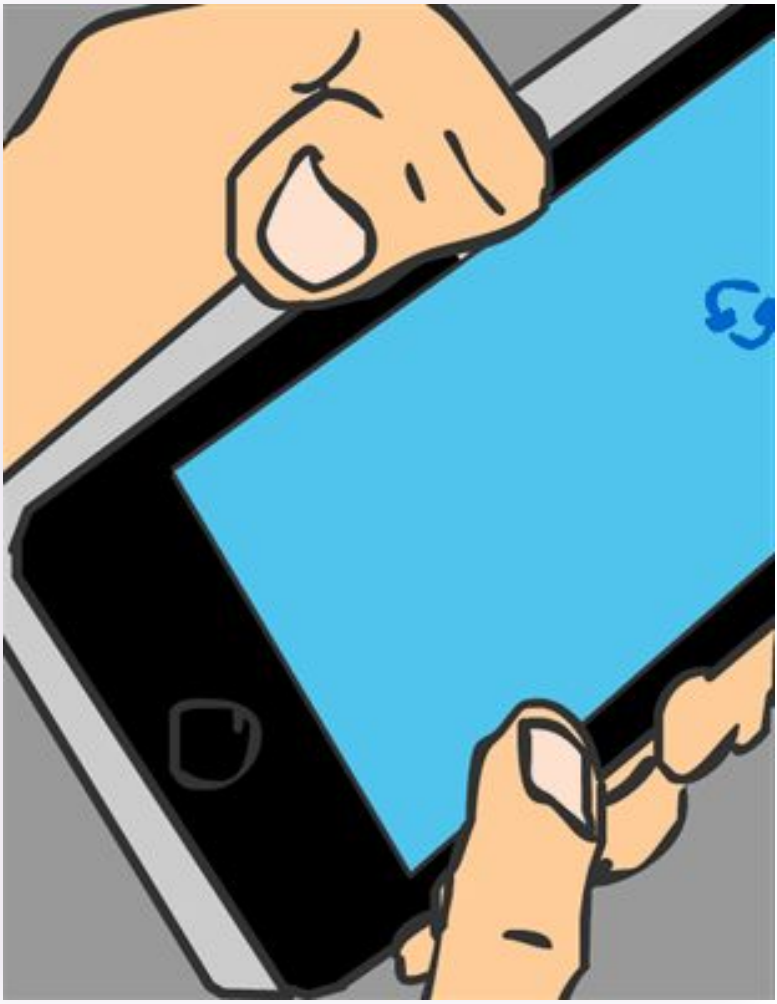
Cumulative Layout Shift

Are there unexpected large shifts in the layout of the website? Is the layout all settled?

Less than .1

Less than .25

Greater than .25

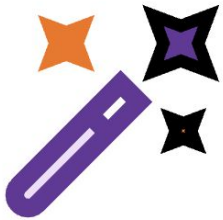


The **Tools** You Need

Developer Tools for Measuring Web Vitals

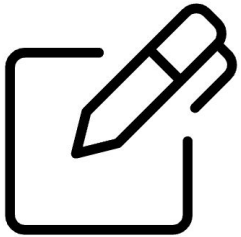
	LCP	FID	CLS
PageSpeed Insights	✅	✅	✅
Chrome UX Report Brand new API, BigQuery and Dashboard	✅	✅	✅
Search Console	✅	✅	✅
Chrome DevTools	✅	TBT	✅
Lighthouse	✅	TBT	✅
Web Vitals Extension	✅	✅	✅

Gatsby



Lab Data

Synthetic measurement of performance



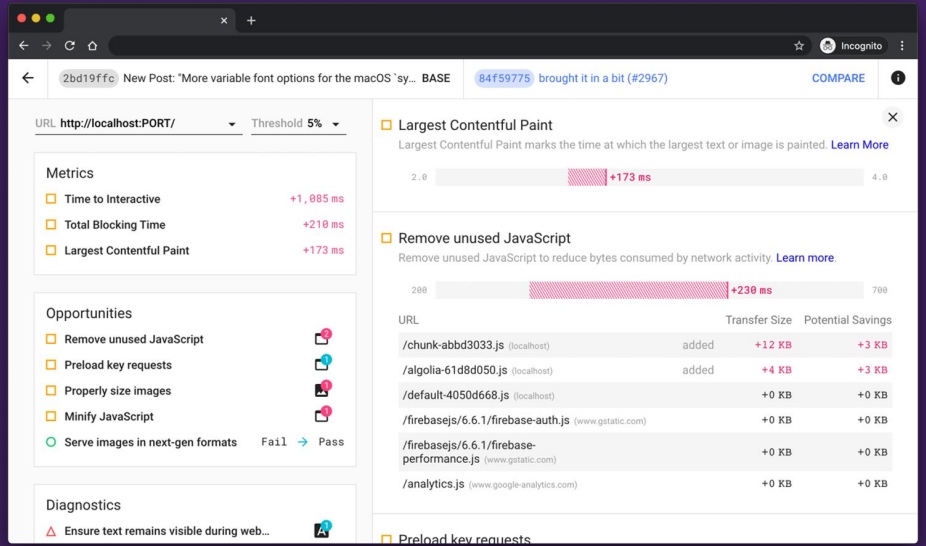
Field Data

Real User measurement of performance

Gatsby

Lighthouse

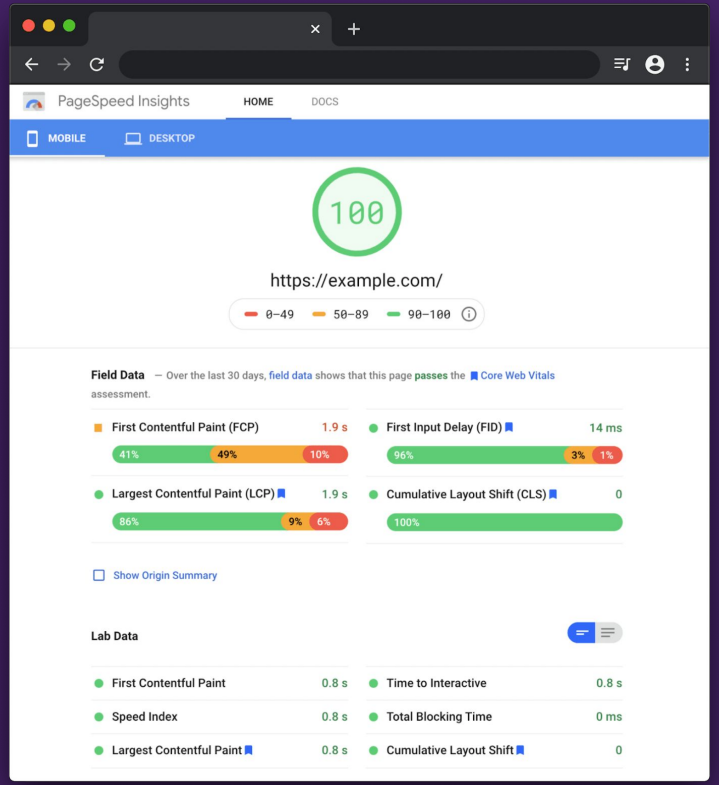
- Open source tooling for assessing page performance (from, and used by, Google)
- Can be run in the browser via DevTools or programmatically via a CLI
- LCP and CLS are measured by Lab metrics
- FID is measured as a field metric



Gatsby

PageSpeed Insights

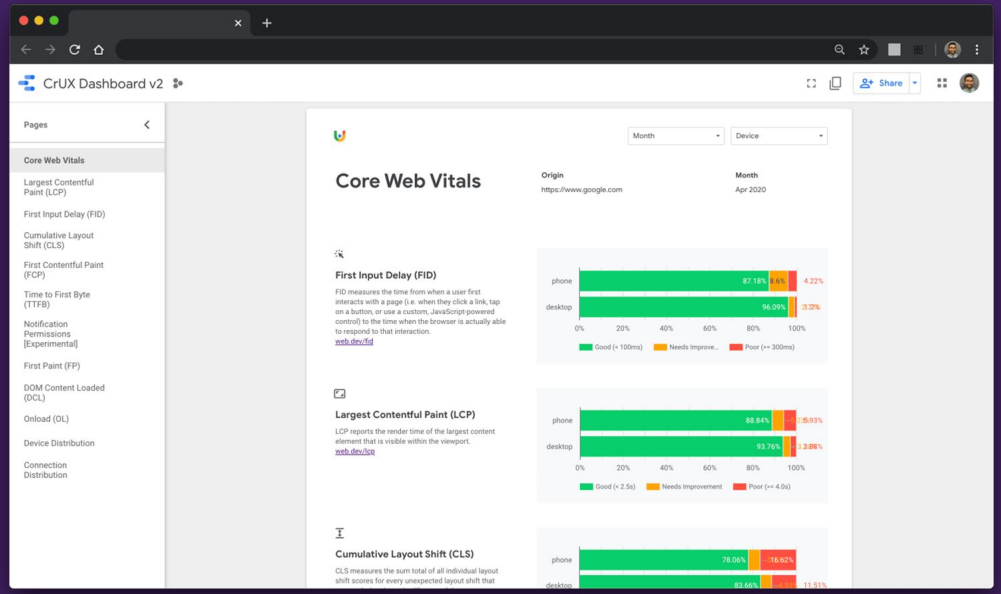
- Powered by Lighthouse
- Reports on the PSI come from both Field performance and Lab Performance
- Browser based + API



Gatsby

Chrome UX Report

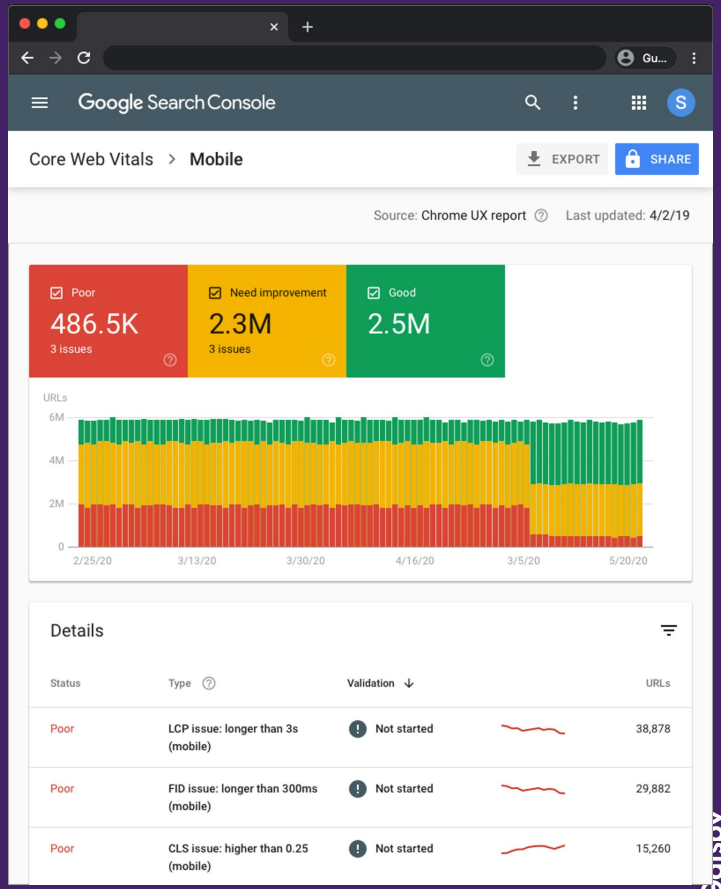
- The ultimate field performance metric tool
- Measures real data from opted-in users
- Aggregates data from all (competitor!) domains, over time
- Can use via BigQuery or via API



Gatsby

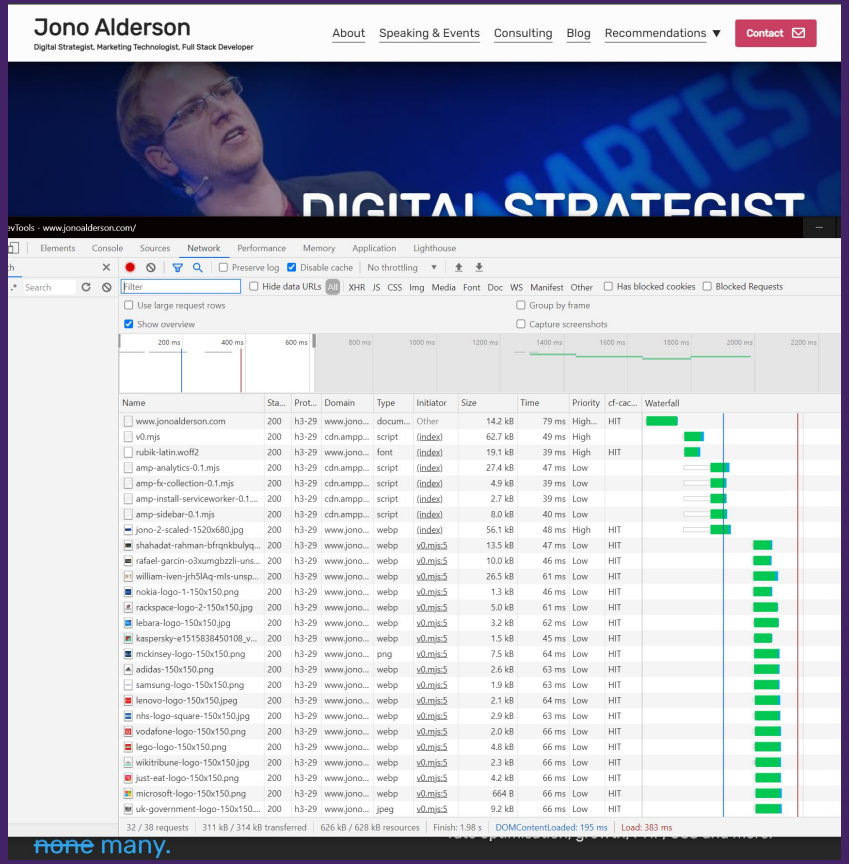
Search Console

- Best to measure and identify large groups of pages that need attention.
- Performance is grouped by status, metric type and URL group



Chrome DevTools

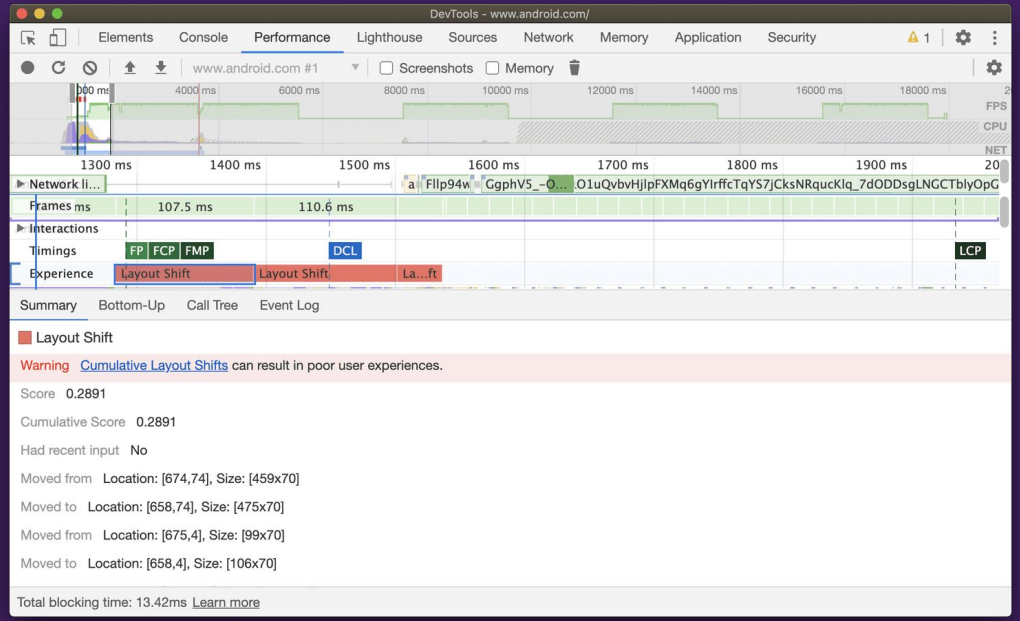
- Chrome devtools gives you almost everything you need to explore, diagnose, and identify issues.
- Waterfall reports let you interrogate every single asset's loading behaviour. This is where you'll get your biggest wins.



Gatsby

Chrome DevTools

- The Performance panel in DevTools has a new Experience section that can help you detect unexpected layout shifts.
- It can also help find sections of your site that have large Total Blocking Times



Gatsby

The **Future** of CWV

What’s next for Google?

- Explicit (minor*) ranking factor later this year
- Changing, and tougher score thresholds
- New metrics and measurements (smoothness?)
- Expansion beyond speed (accessibility, SEO, etc)

Sam Bhagwat

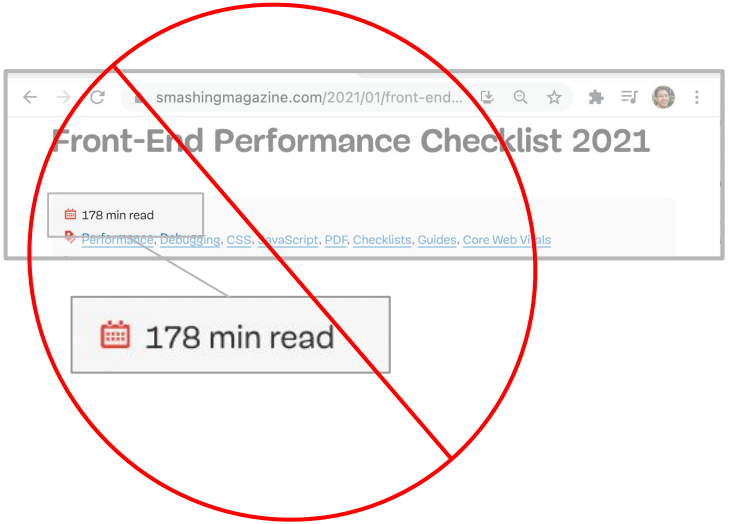
Cofounder, Chief Strategy Officer @ Gatsby

Core Web Vitals: Gaining Your Intuition

What you're not going to take away

Details about what every single perf bottleneck is

....or how to fix each of them



Gatsby

What I hope you'll take away

When you watch a page load on your website, you have an intuitive feel for:

a. how much it's been optimized
b. how good the Web Core Vitals look
c. how to make it better

Gatsby

What we're going to be doing

1. Watch an *optimized* page load, and an *unoptimized page load*
2. See where "Core Web Vitals" events happen in each experience
3. Discuss 7 optimizations that will get your perf work to >=90% done

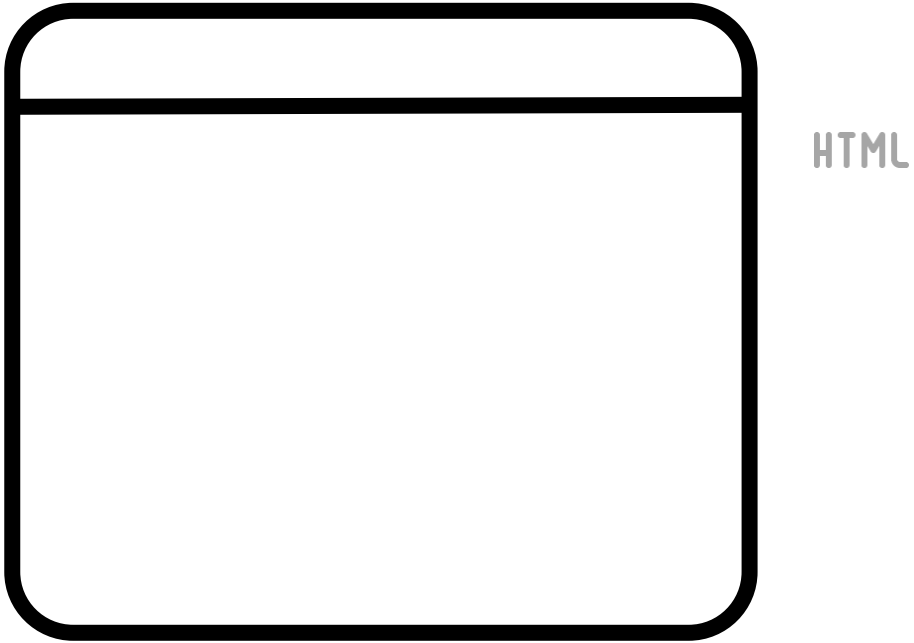
Gatsby

Gatsby

Unoptimized Page Load

Gatsby

UNOPTIMIZED WEBSITE

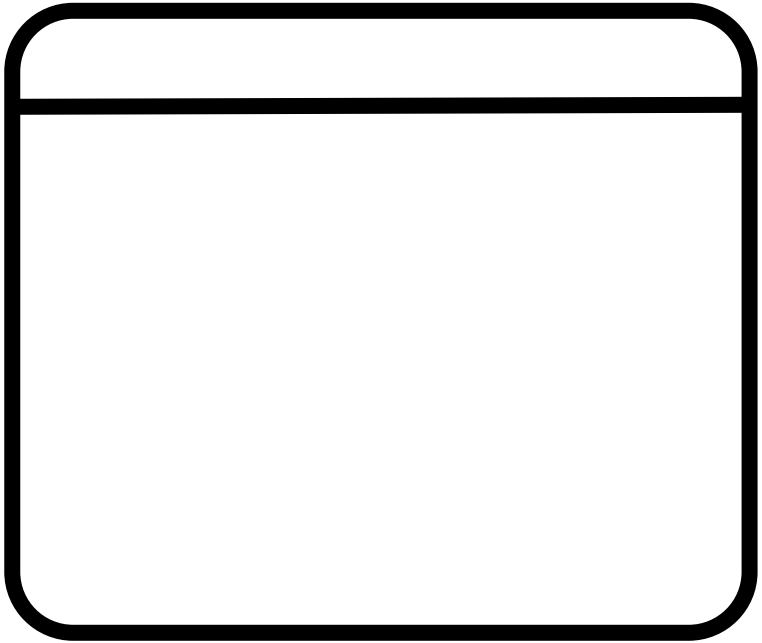


HTML

0s

Gatsby

UNOPTIMIZED WEBSITE

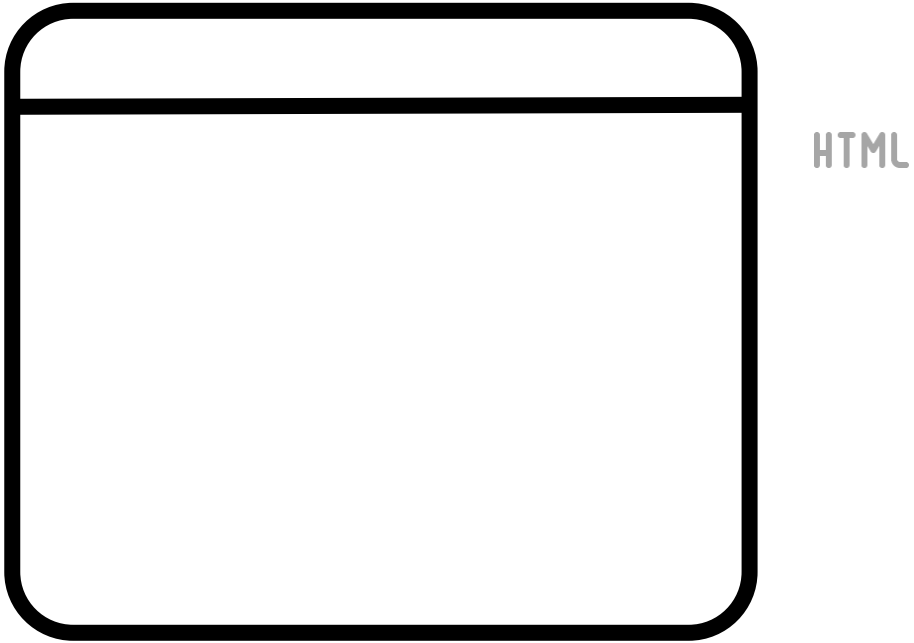


HTML

0.5s

Gatsby

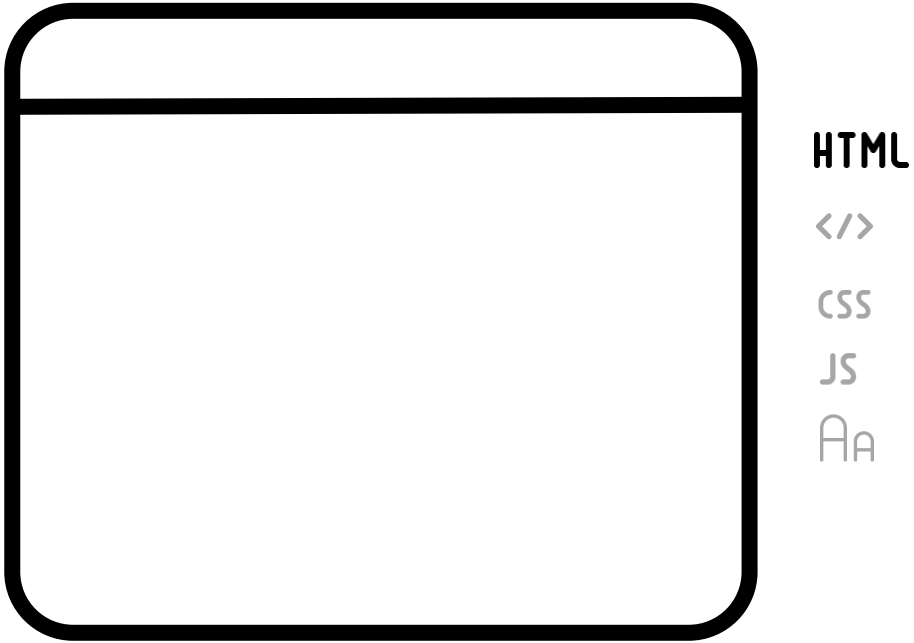
UNOPTIMIZED WEBSITE



1.0s

Gatsby

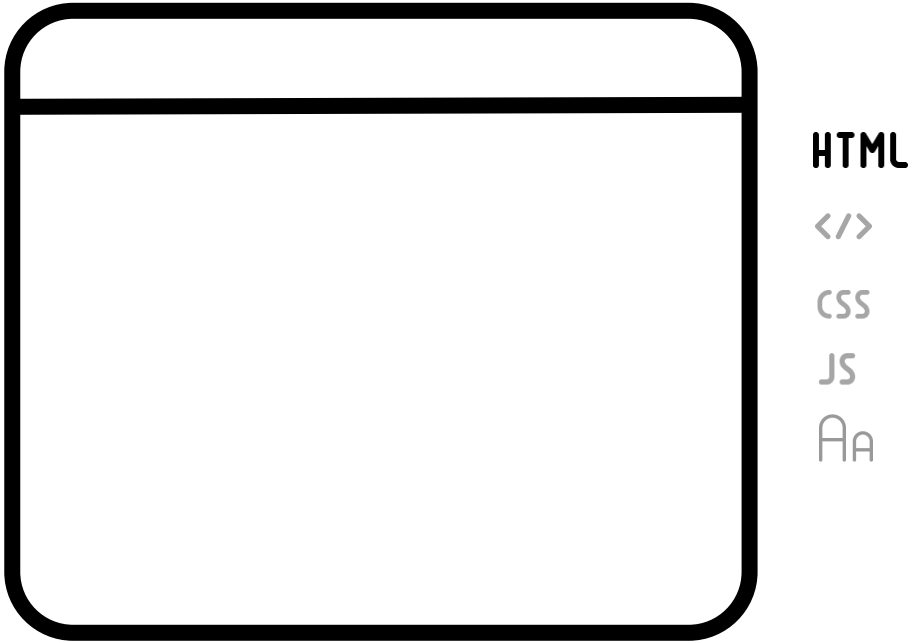
UNOPTIMIZED WEBSITE



1.5s

Gatsby

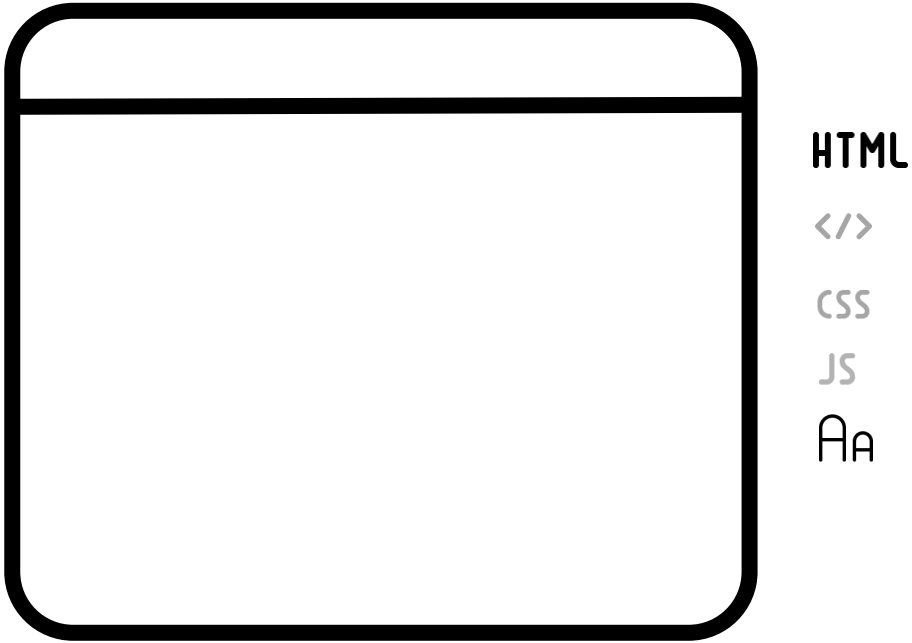
UNOPTIMIZED WEBSITE



2.0s

Gatsby

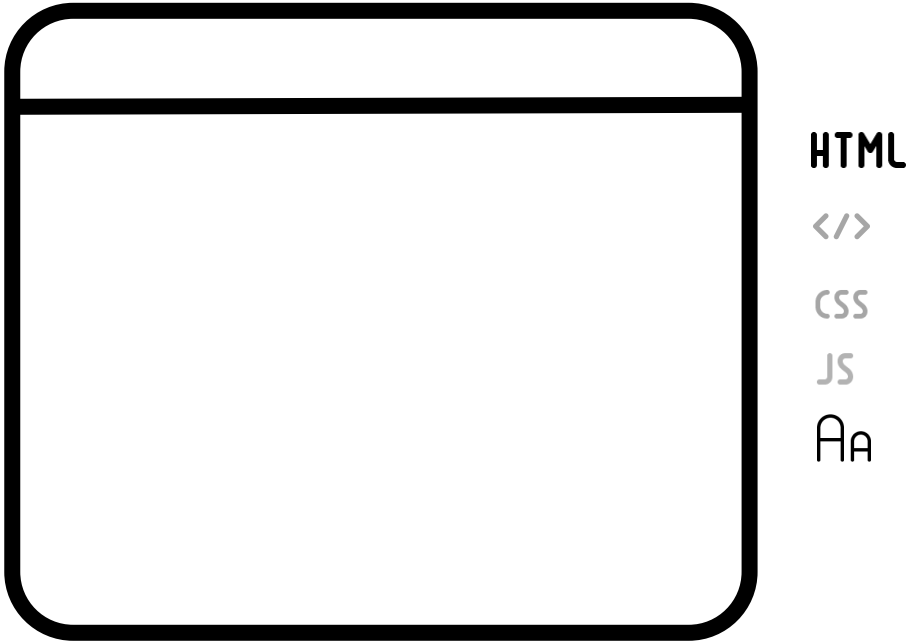
UNOPTIMIZED WEBSITE



2.5s

Gatsby

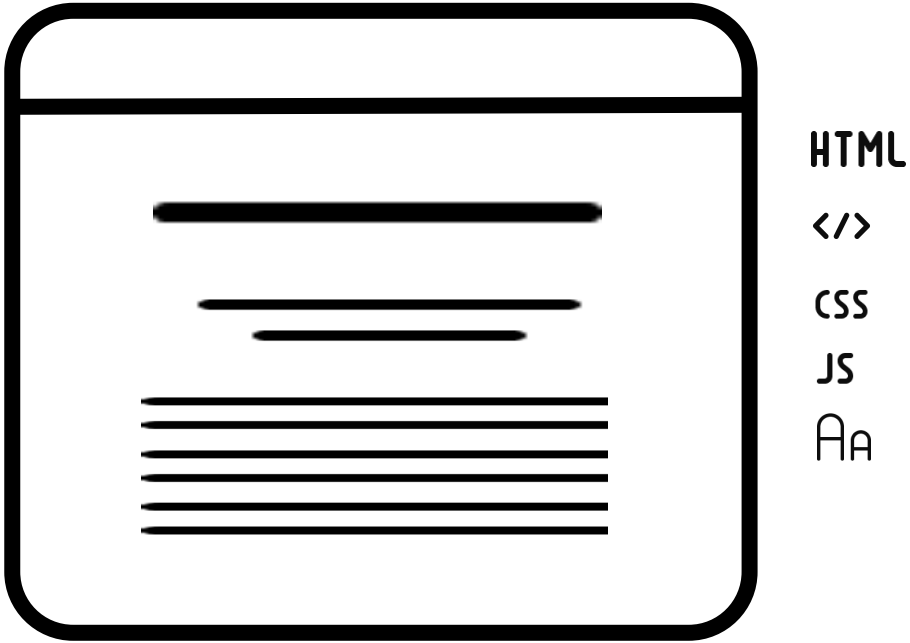
UNOPTIMIZED WEBSITE



3.0s

Gatsby

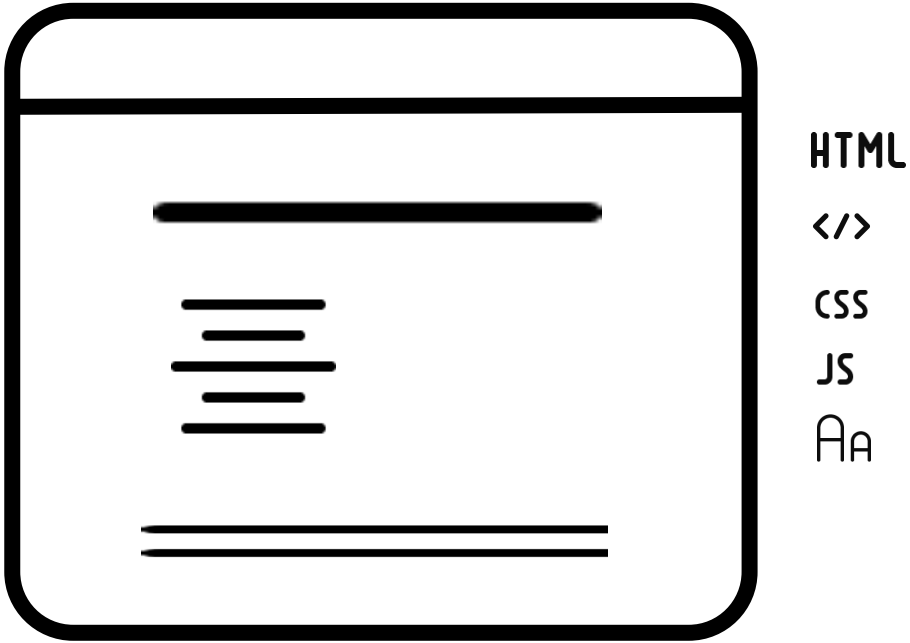
UNOPTIMIZED WEBSITE



3.5s

Gatsby

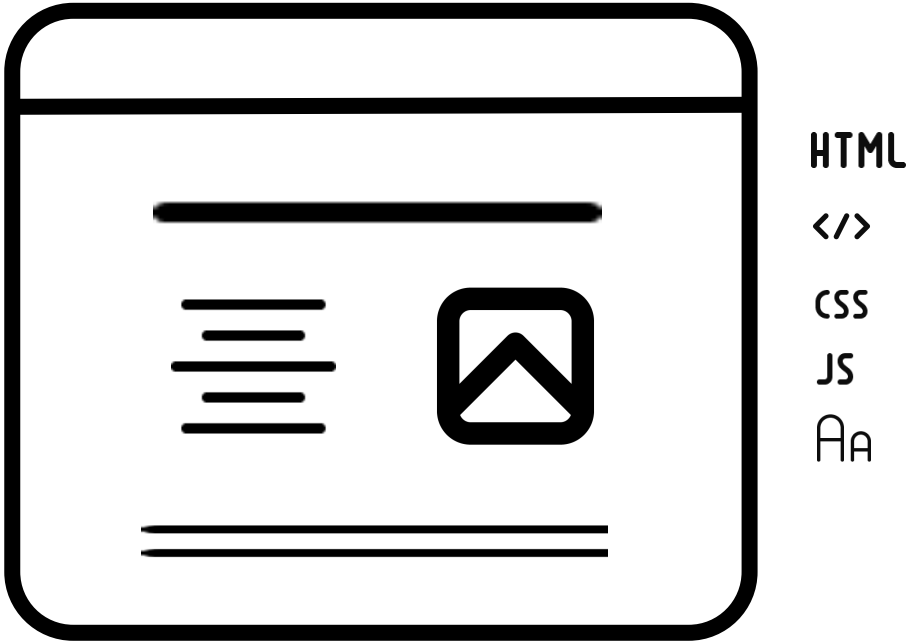
UNOPTIMIZED WEBSITE



4.0s

Gatsby

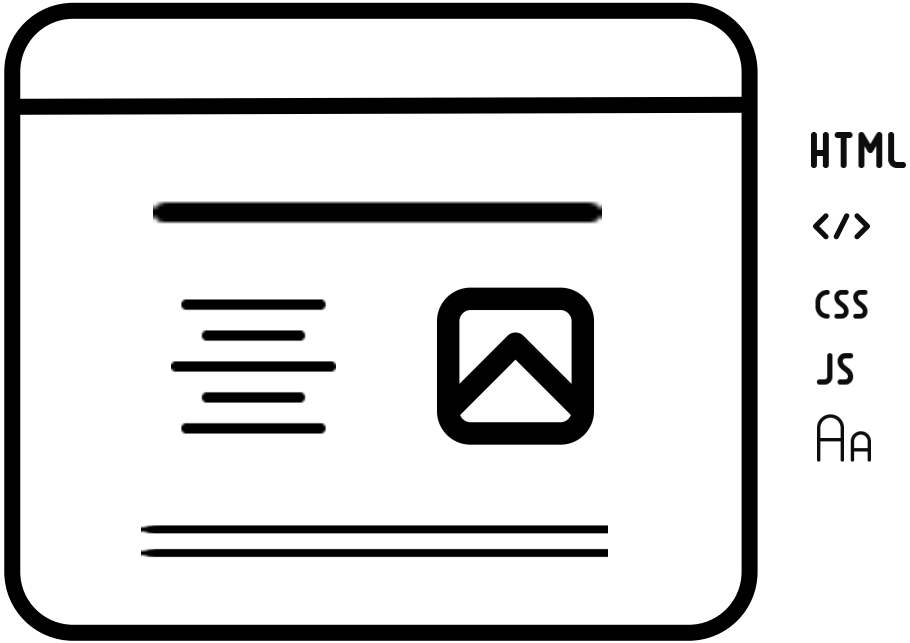
UNOPTIMIZED WEBSITE



4.5s

Gatsby

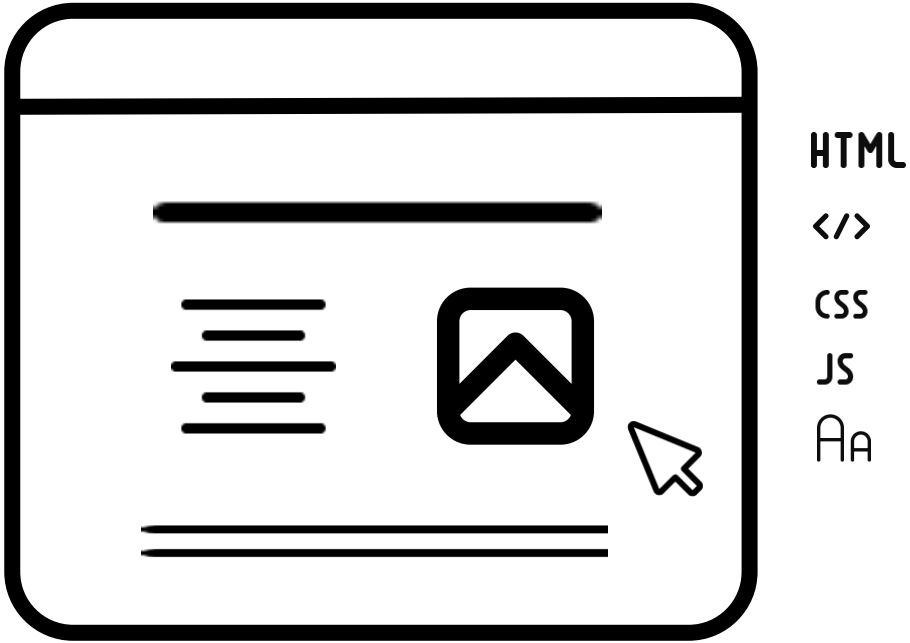
UNOPTIMIZED WEBSITE



5.0s

Gatsby

UNOPTIMIZED WEBSITE



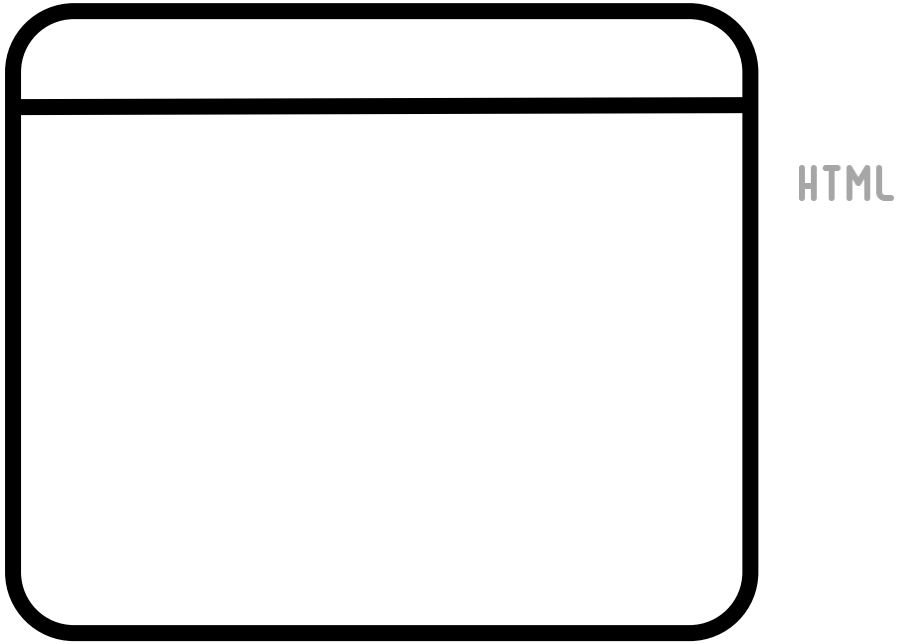
5.5s

Gatsby

Optimized Page Load

Gatsby

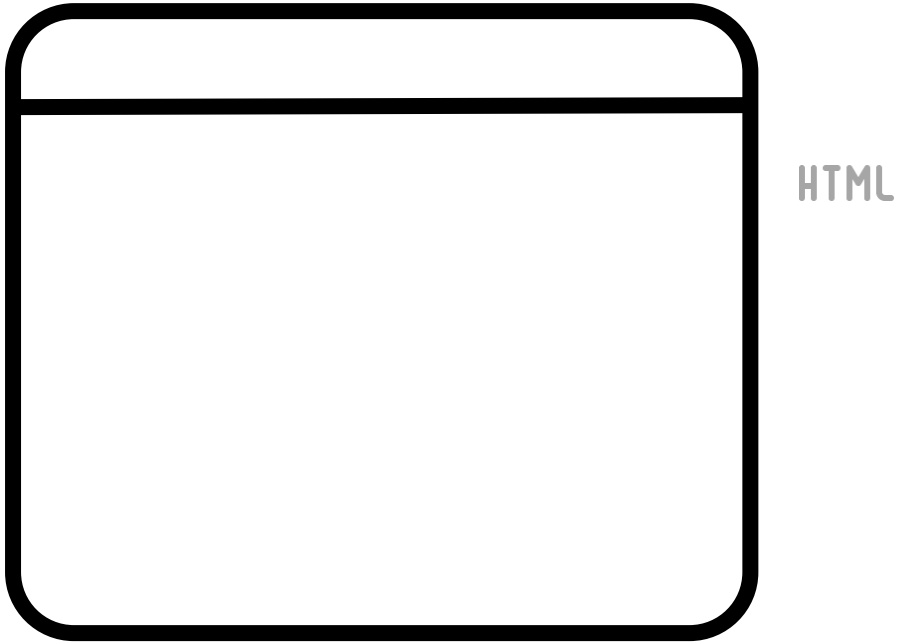
OPTIMIZED WEBSITE



0s

Gatsby

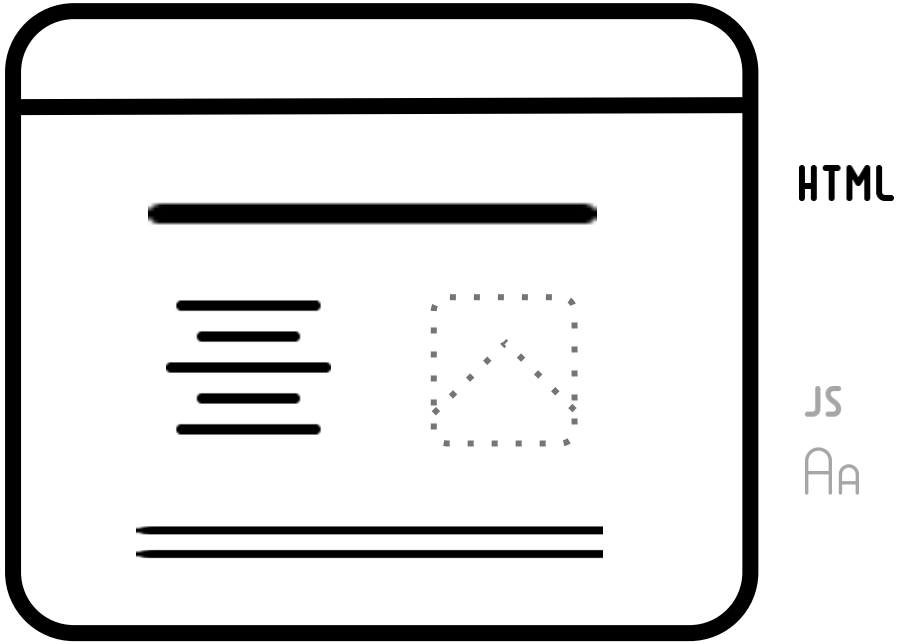
OPTIMIZED WEBSITE



0.5s

Gatsby

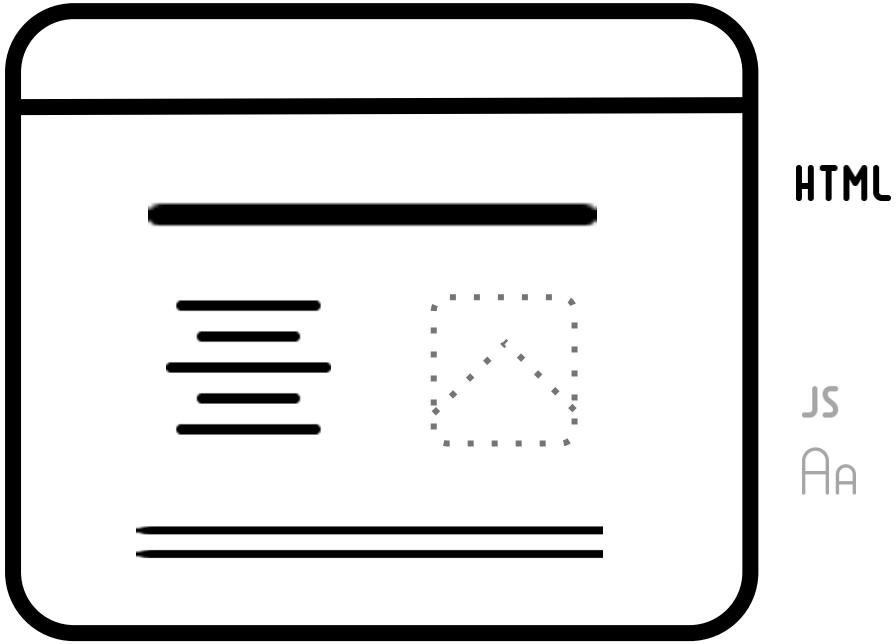
OPTIMIZED WEBSITE



1.0s

Gatsby

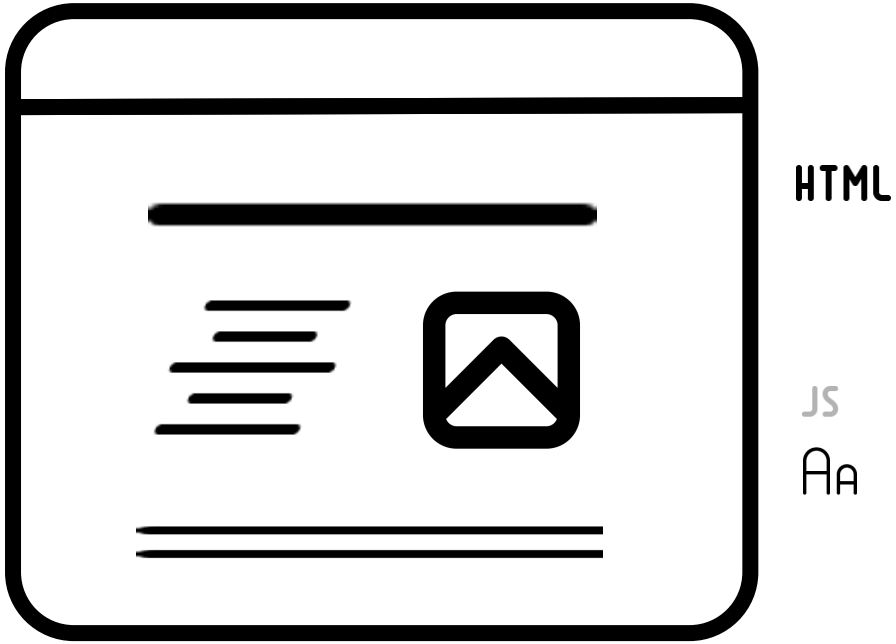
OPTIMIZED WEBSITE



1.5s

Gatsby

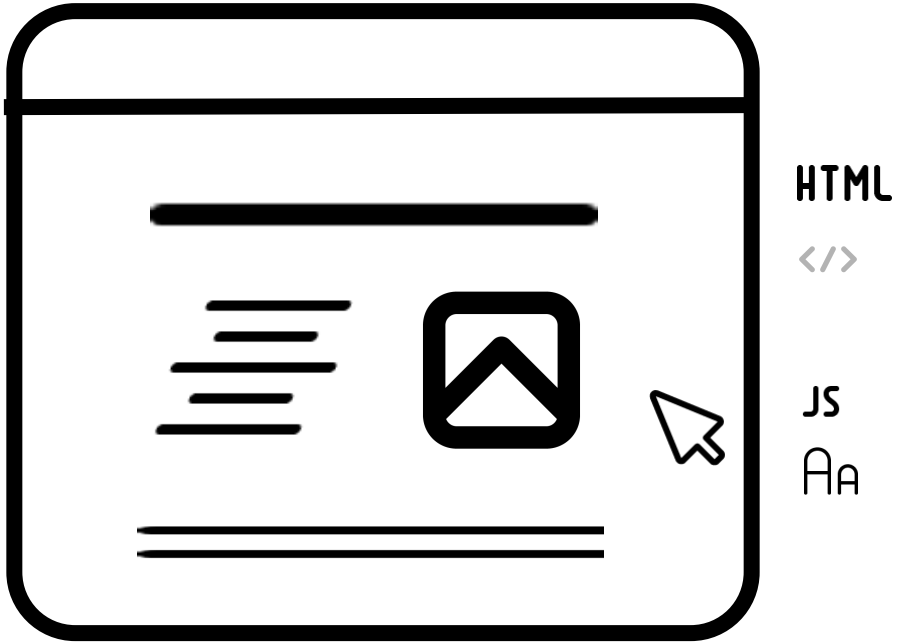
OPTIMIZED WEBSITE



2.0s

Gatsby

OPTIMIZED WEBSITE



2.5s

Gatsby

Where do “Core Web Vitals” events happen in each experience?

Gatsby

Largest Contentful Paint

Are your largest images and videos rendered? Is all text visible at this point?

0-2.5 Seconds

2.6-4.0 Seconds

> 4 Seconds

OPTIMIZED WEBSITE

HTML

JS

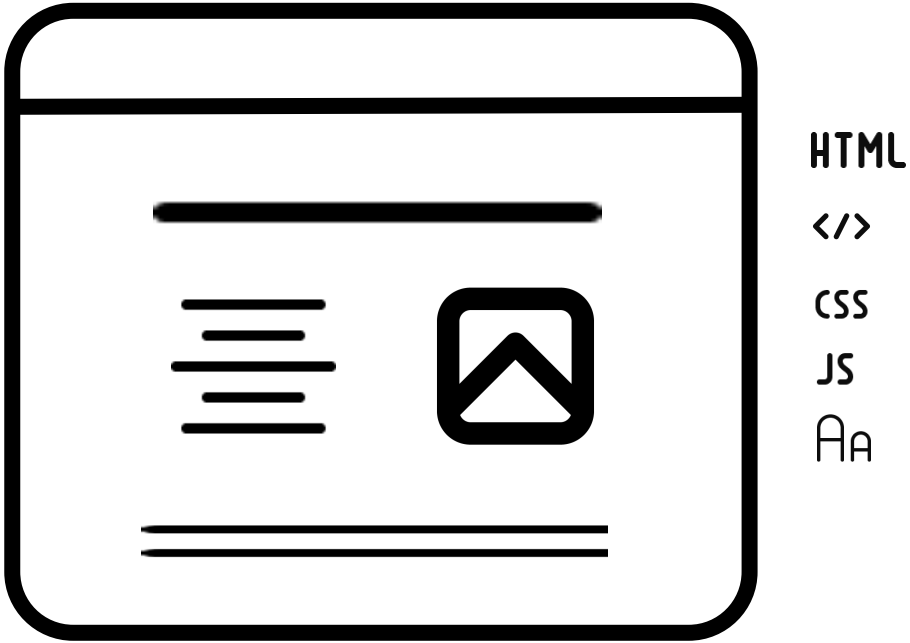
Aa

Largest Contentful Paint

2.0s

Gatsby

UNOPTIMIZED WEBSITE



Largest Contentful Paint

4.5s

Gatsby

Cumulative Layout Shift

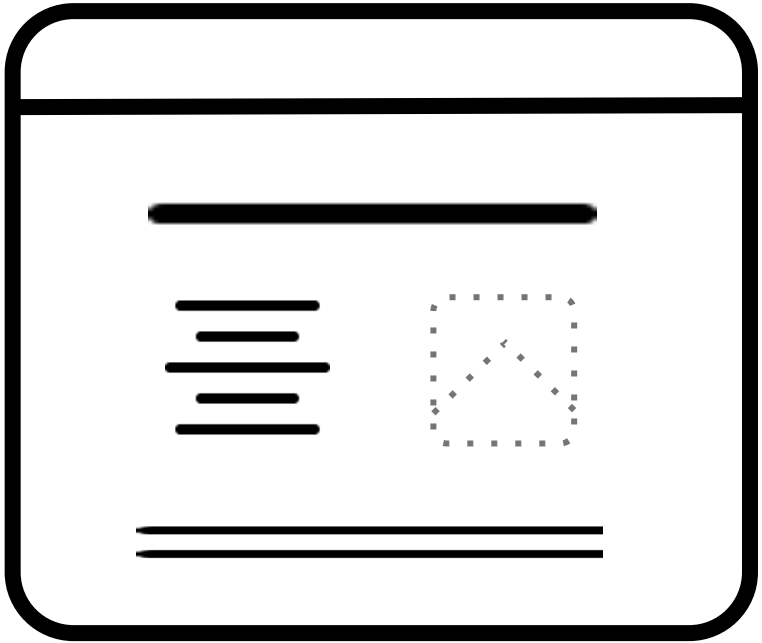
Are there unexpected large shifts in the layout of the website? Is the layout all settled?

Less than .1

Less than .25

Greater than .25

OPTIMIZED WEBSITE



HTML

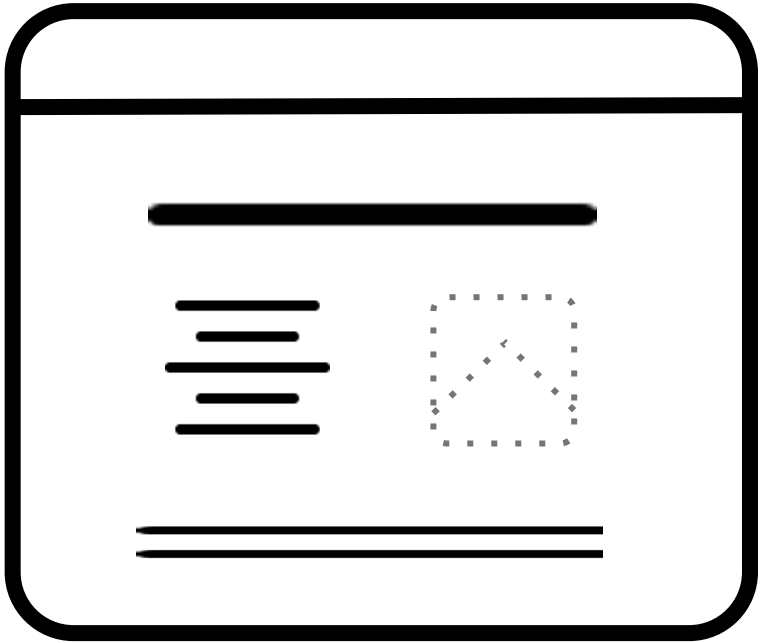
JS

Aa

1.0s

Gatsby

OPTIMIZED WEBSITE



HTML

JS

Aa

1.5s

Gatsby

OPTIMIZED WEBSITE

HTML

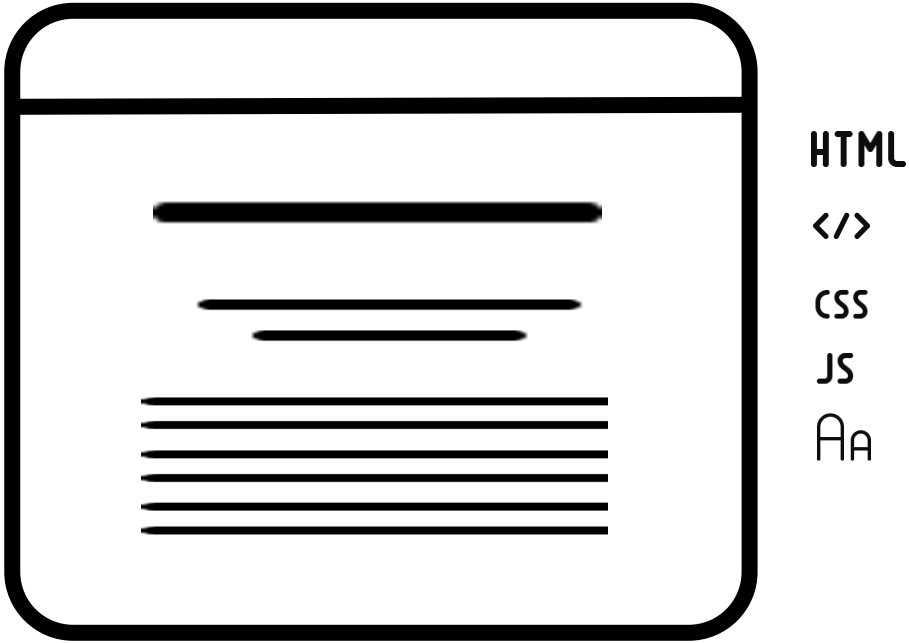
JS

Aa

2.0s

Gatsby

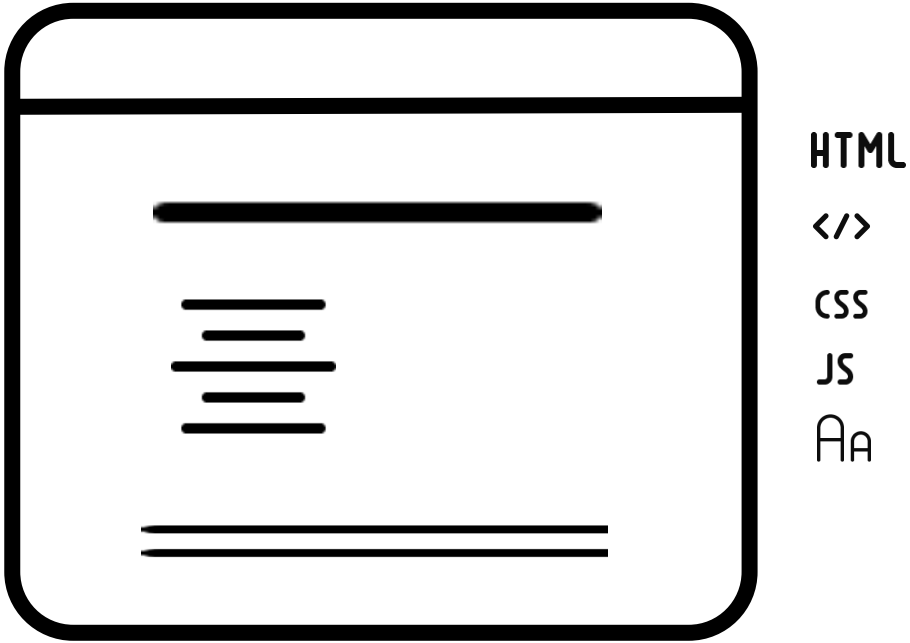
UNOPTIMIZED WEBSITE



3.5s

Gatsby

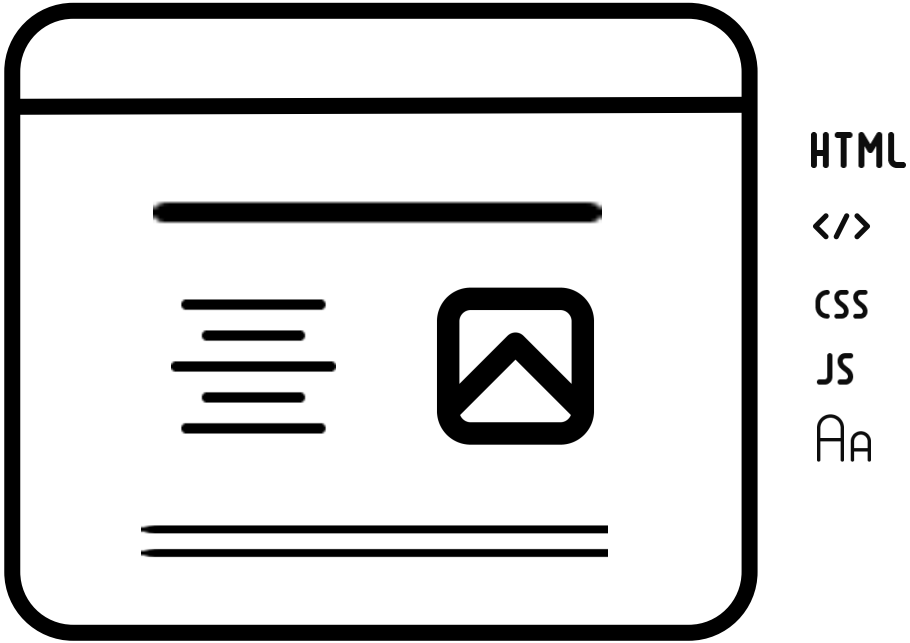
UNOPTIMIZED WEBSITE



4.0s

Gatsby

UNOPTIMIZED WEBSITE



4.5s

Gatsby

First Input Delay

When a user is finally allowed to click something, how fast does your website respond? Are all scripts loaded?

0-100ms

101-300ms

> 300ms

OPTIMIZED WEBSITE

HTML

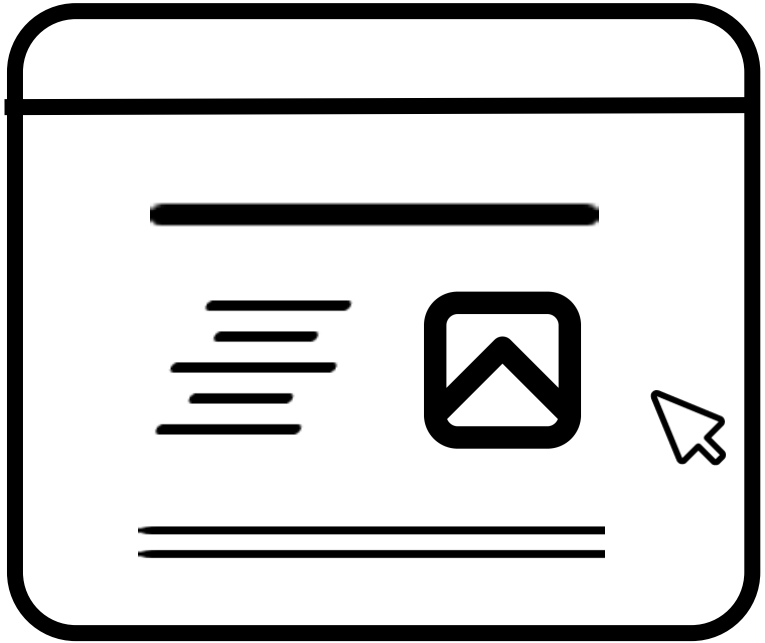
JS

Aa

2.0s

Gatsby

OPTIMIZED WEBSITE



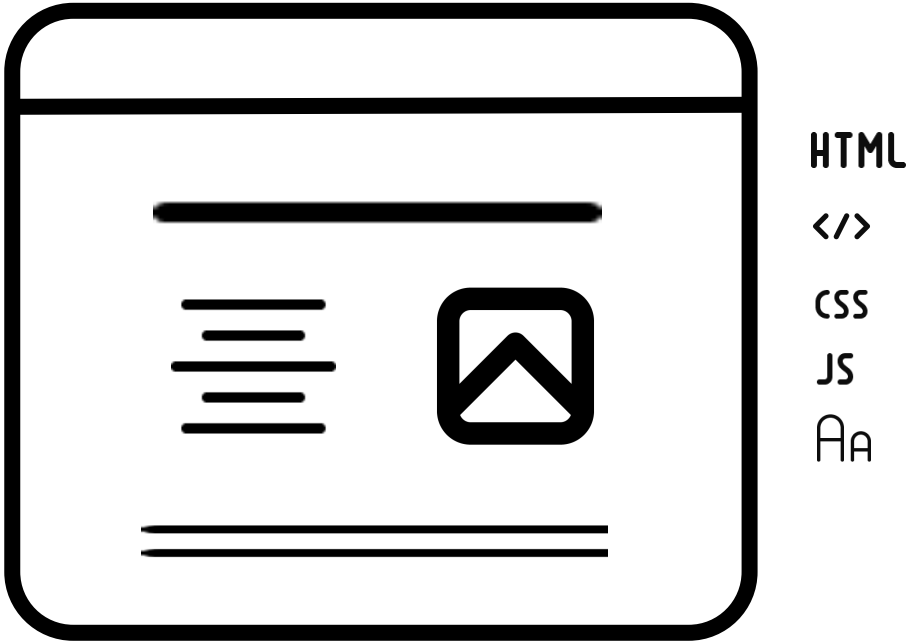
HTML

JS

2.5s

Gatsby

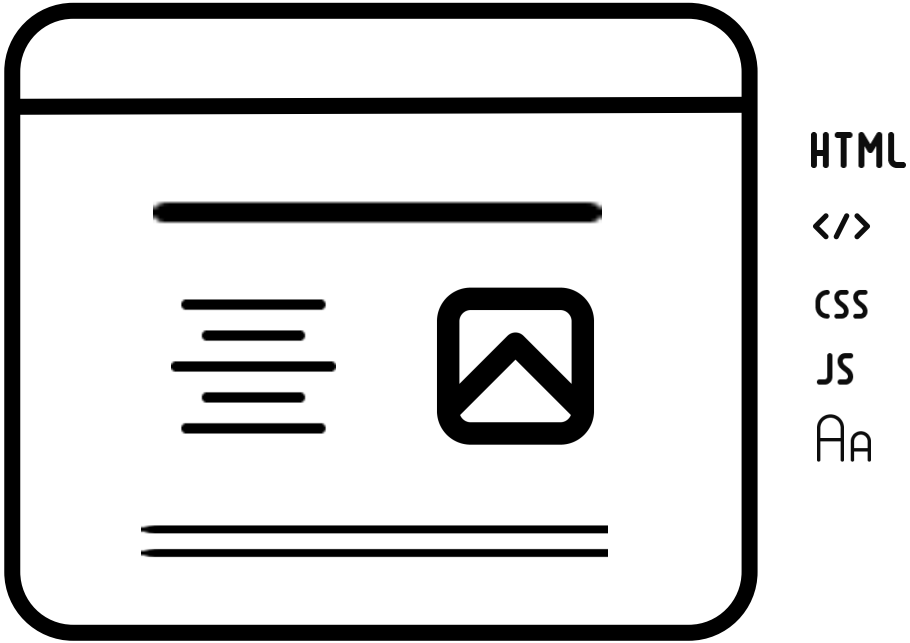
UNOPTIMIZED WEBSITE



4.5s

Gatsby

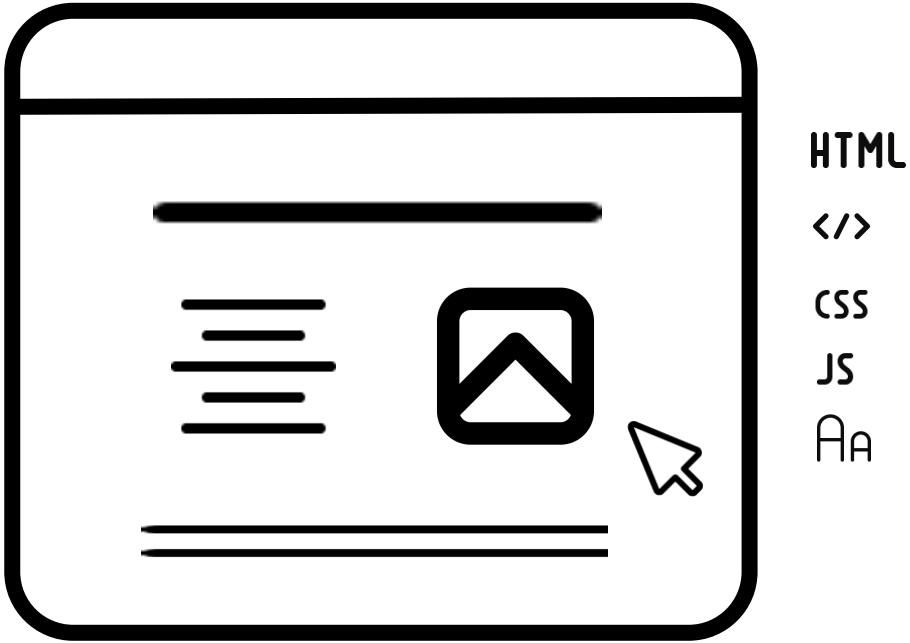
UNOPTIMIZED WEBSITE



5.0s

Gatsby

UNOPTIMIZED WEBSITE



5.5s

Gatsby

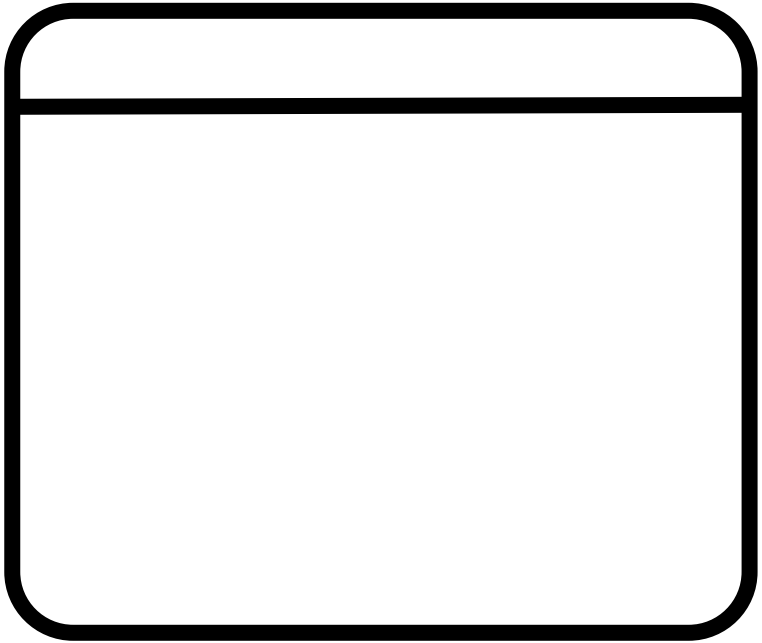
Unoptimized Page Load

vs

Optimized Page Load

Gatsby

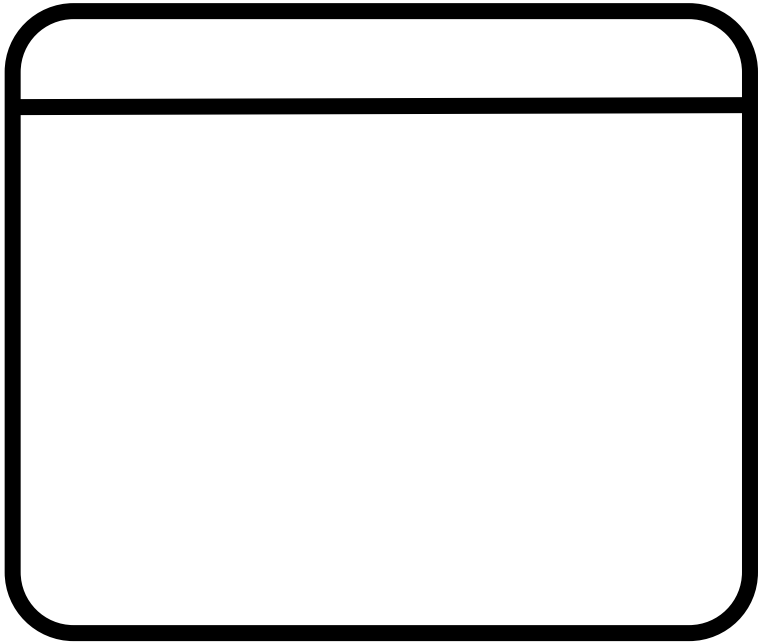
UNOPTIMIZED WEBSITE



HTML

0s

OPTIMIZED WEBSITE

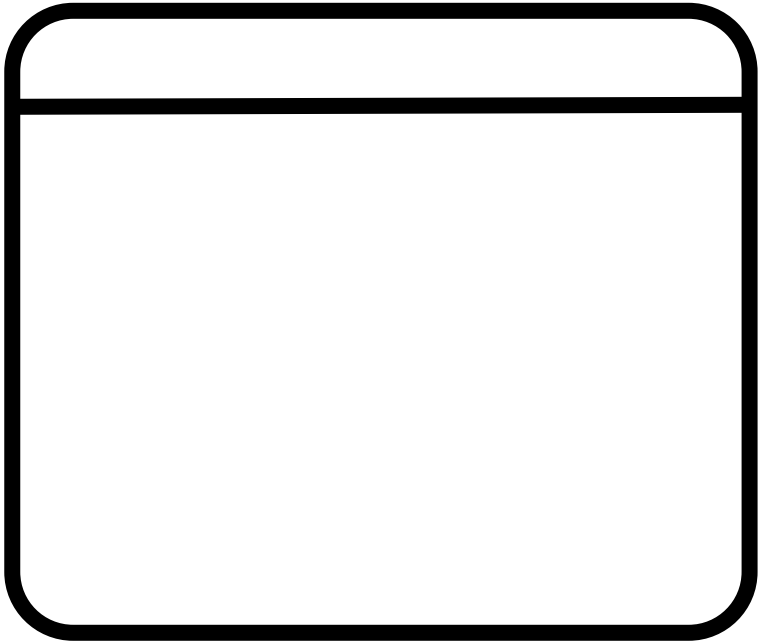


HTML

0s

Gatsby

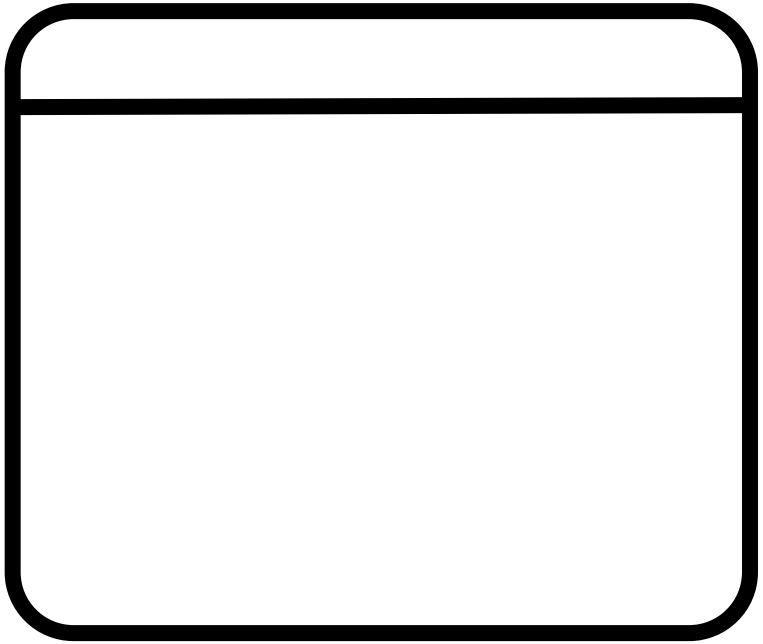
UNOPTIMIZED WEBSITE



HTML

0.5s

OPTIMIZED WEBSITE

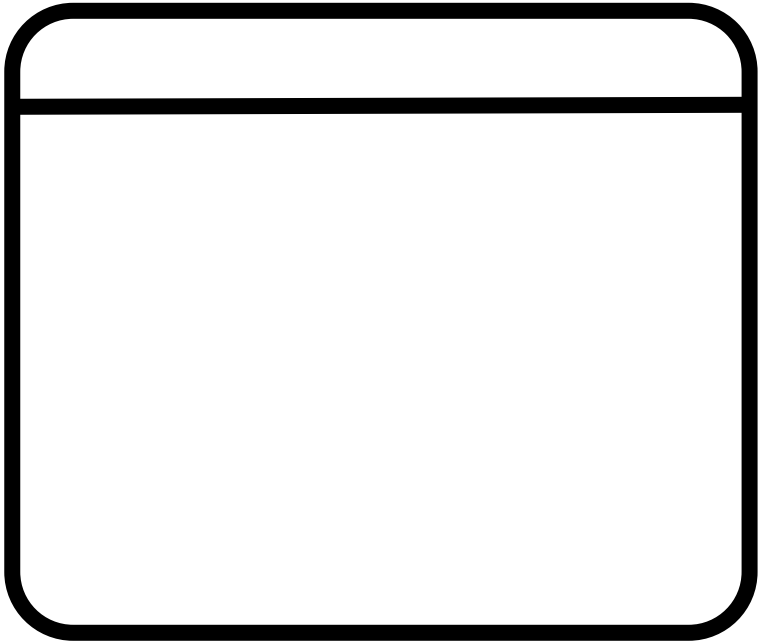


HTML

0.5s

Gatsby

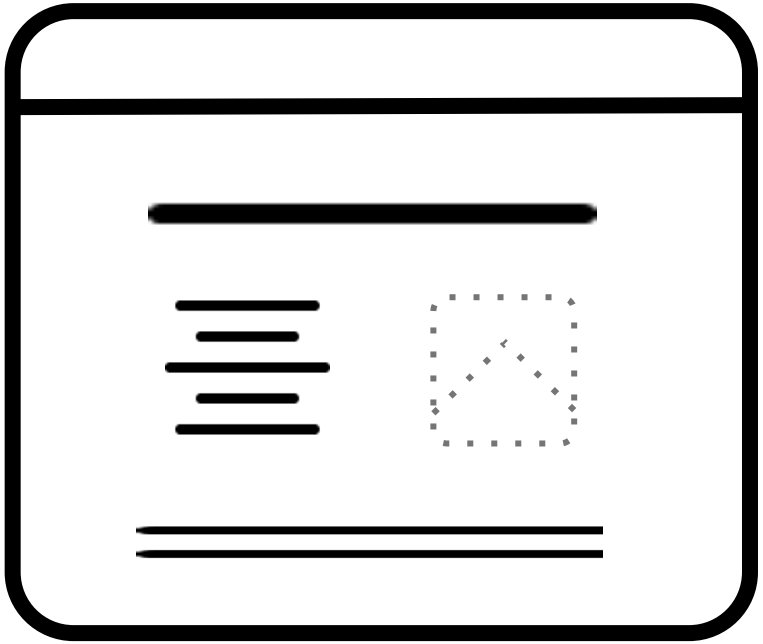
UNOPTIMIZED WEBSITE



HTML

1.0s

OPTIMIZED WEBSITE



HTML

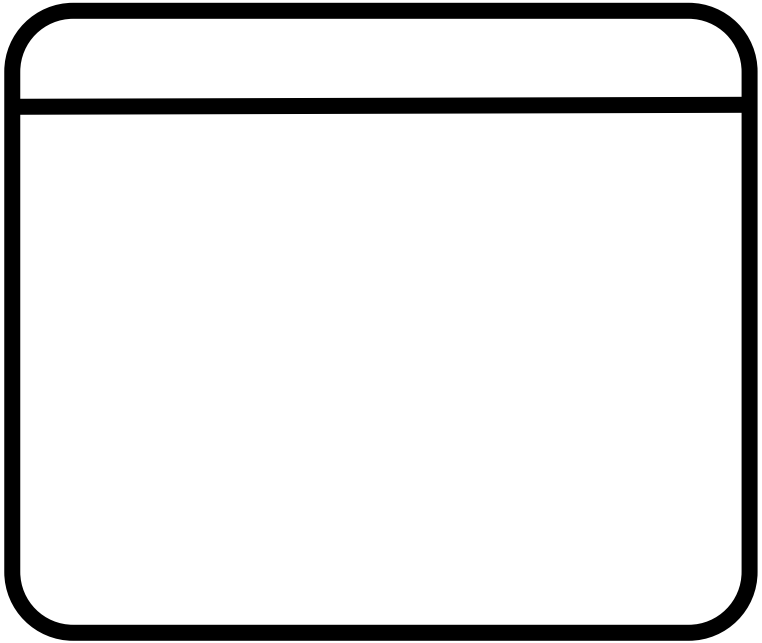
JS

Aa

1.0s

Gatsby

UNOPTIMIZED WEBSITE



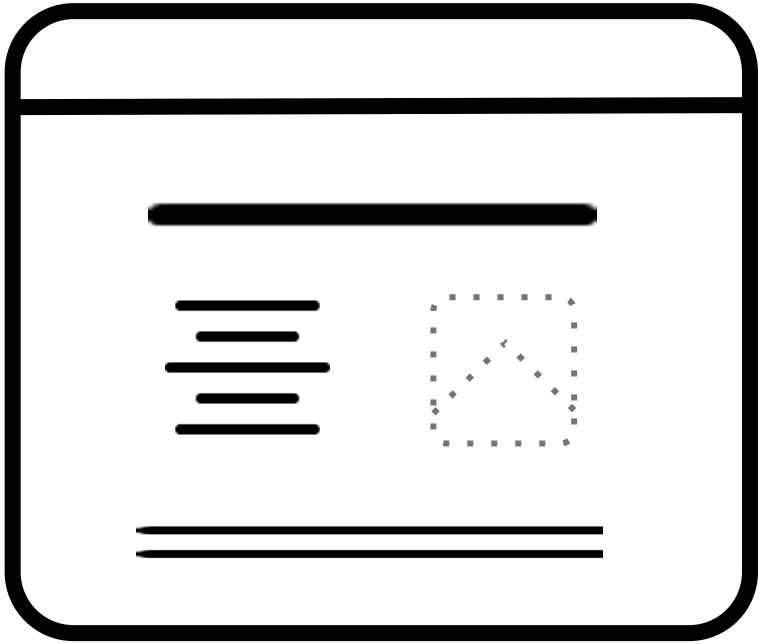
HTML

CSS

JS

1.5s

OPTIMIZED WEBSITE



HTML

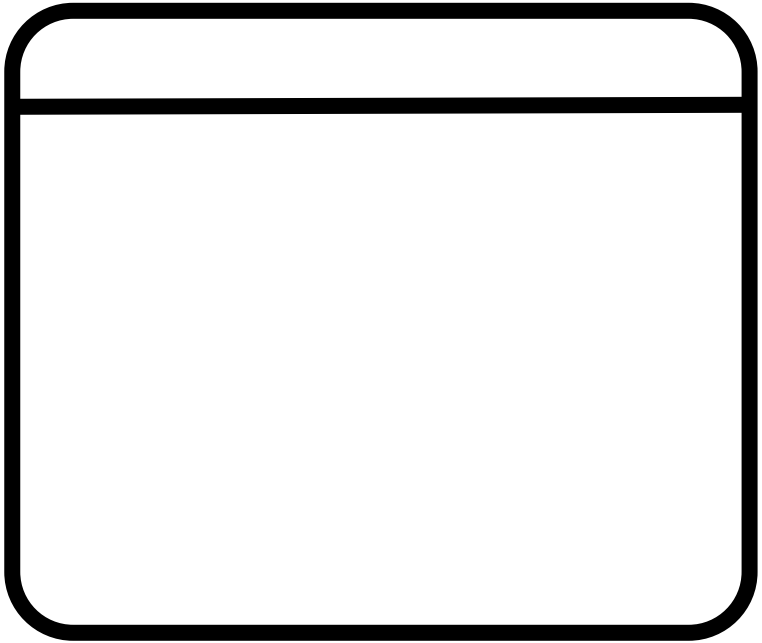
JS

(no Layout Shift)

1.5s

Gatsby

UNOPTIMIZED WEBSITE



HTML

</>

CSS

JS

Aa

2.0s

OPTIMIZED WEBSITE

HTML

JS

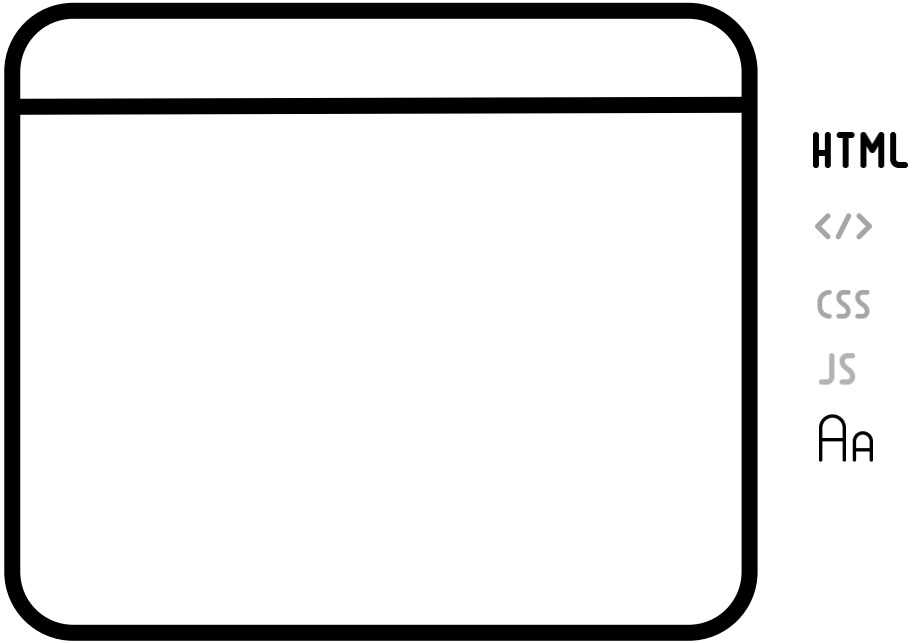
Aa

Largest Contentful Paint

2.0s

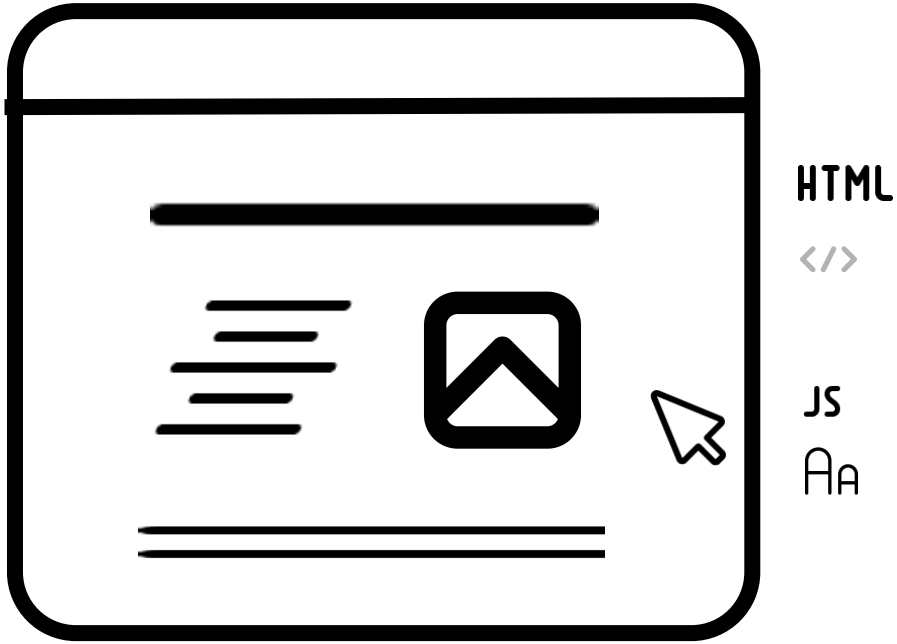
Gatsby

UNOPTIMIZED WEBSITE



2.5s

OPTIMIZED WEBSITE

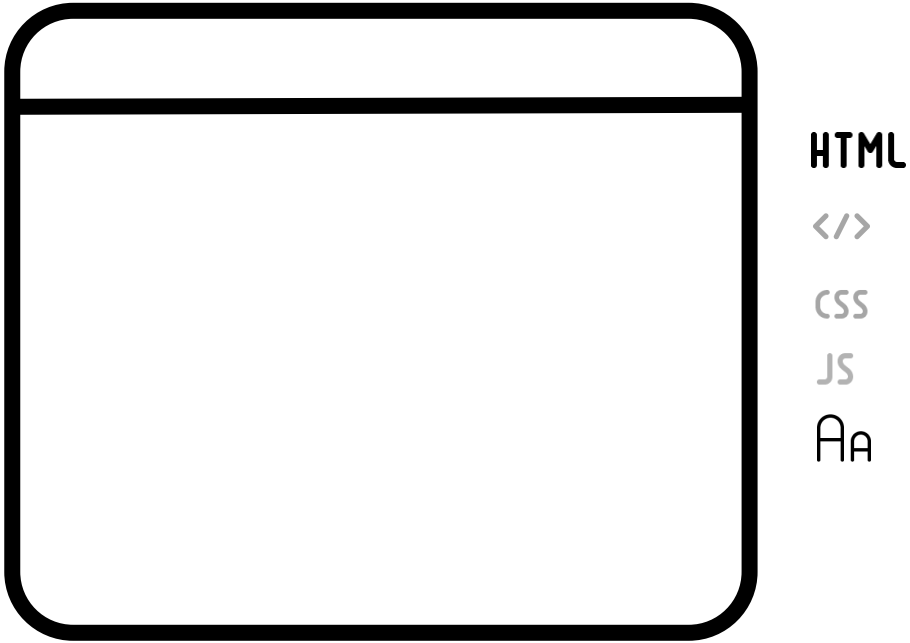


(low First Input Delay)

2.5s

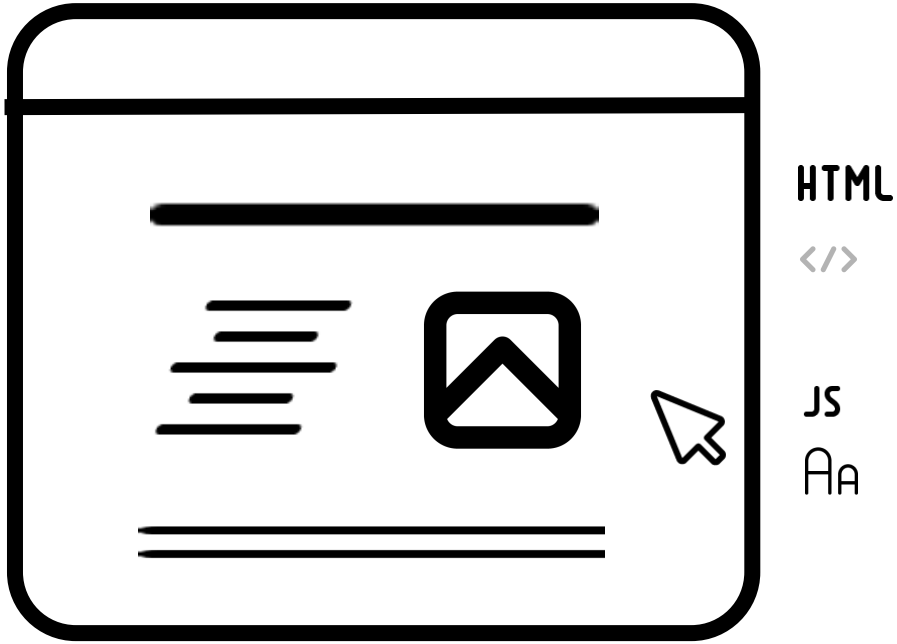
Gatsby

UNOPTIMIZED WEBSITE



3.0s

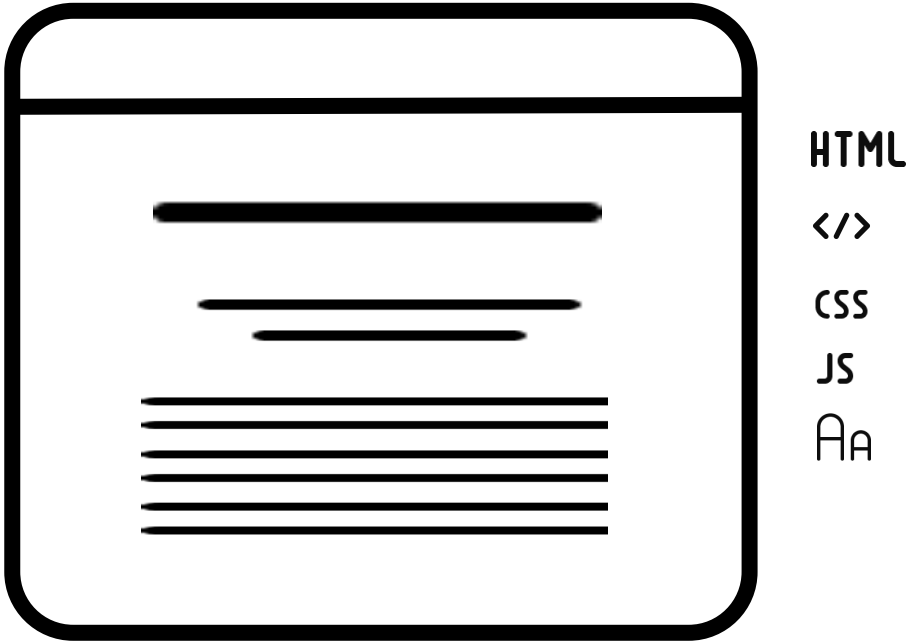
OPTIMIZED WEBSITE



3.0s

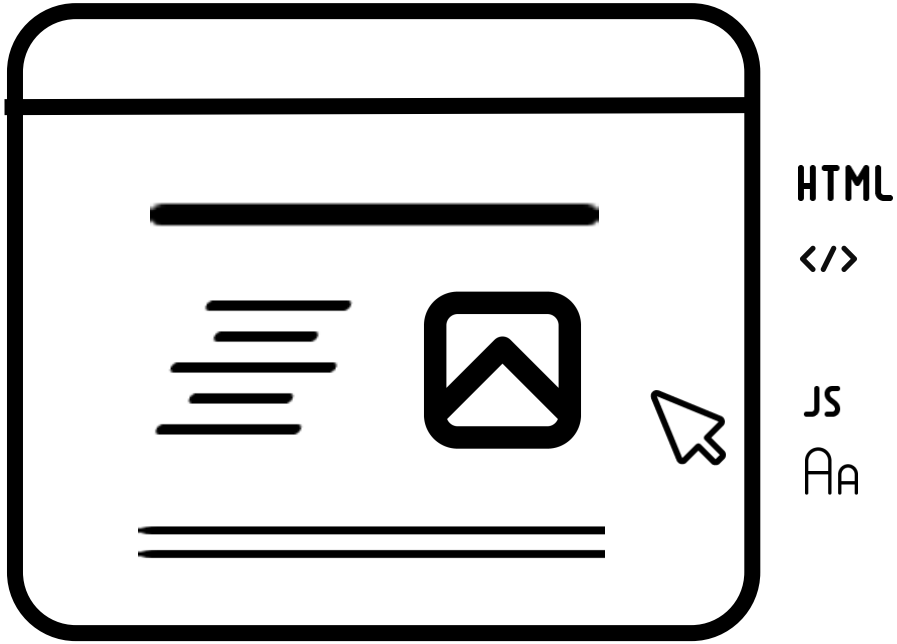
Gatsby

UNOPTIMIZED WEBSITE



3.5s

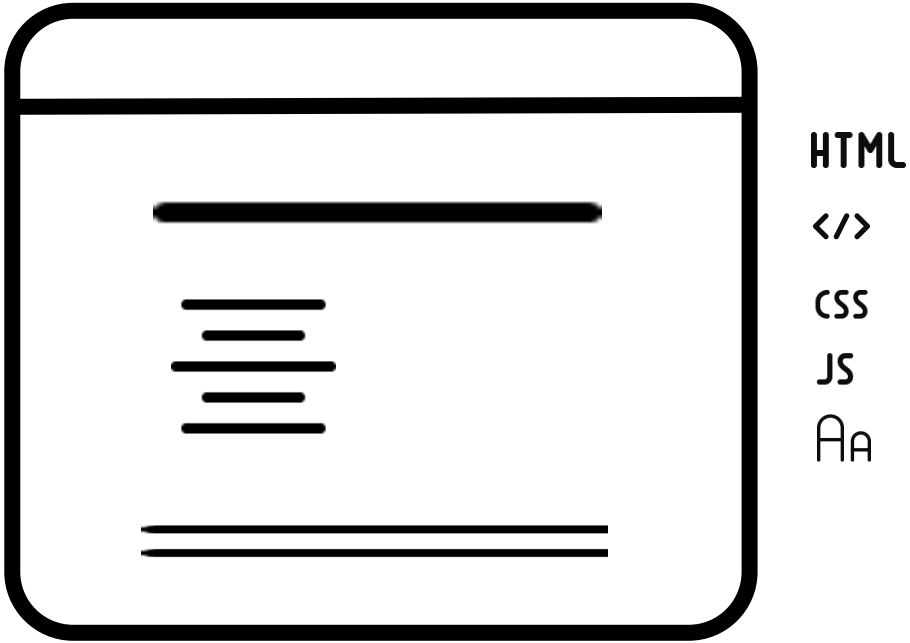
OPTIMIZED WEBSITE



3.5s

Gatsby

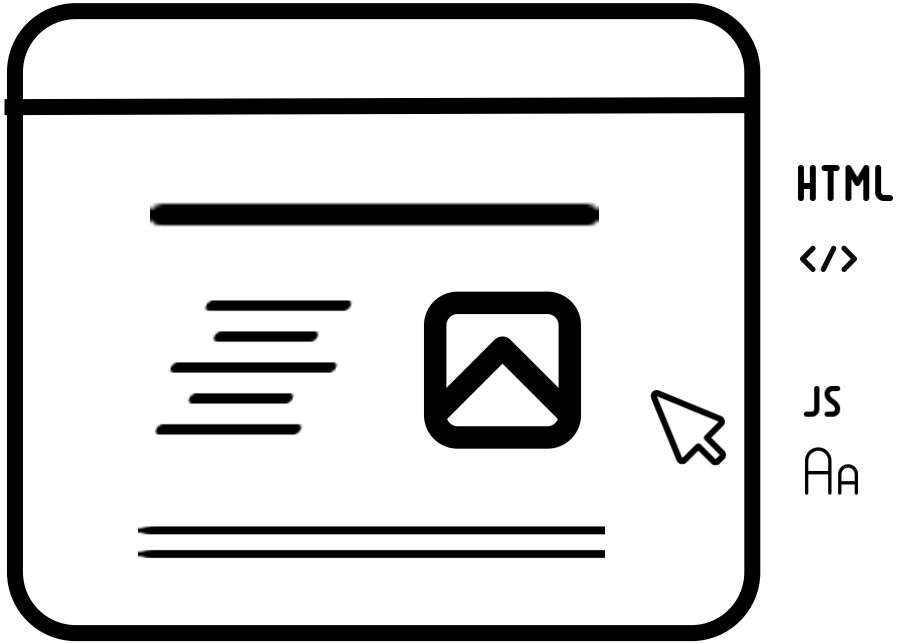
UNOPTIMIZED WEBSITE



(large Layout Shift)

4.0s

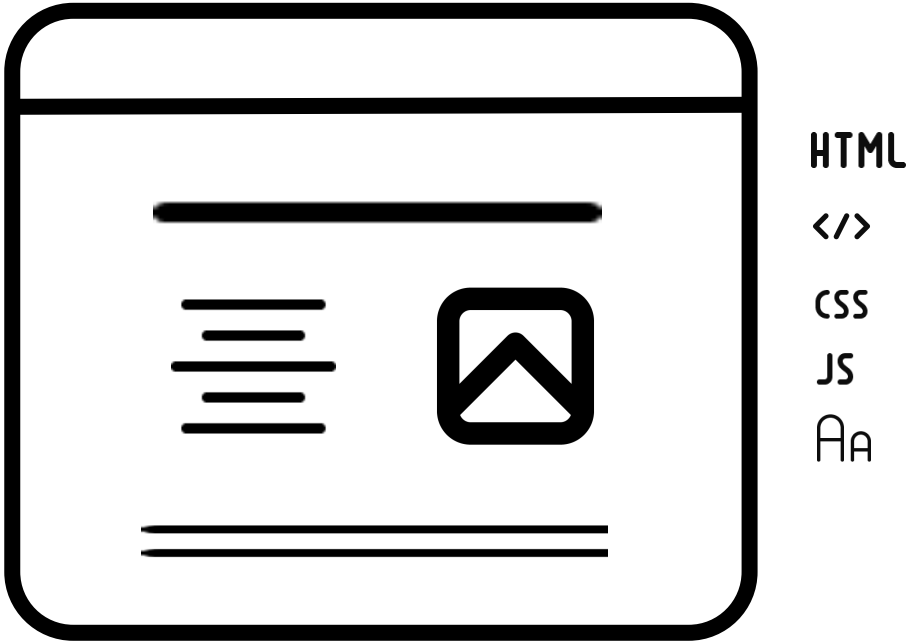
OPTIMIZED WEBSITE



4.0s

Gatsby

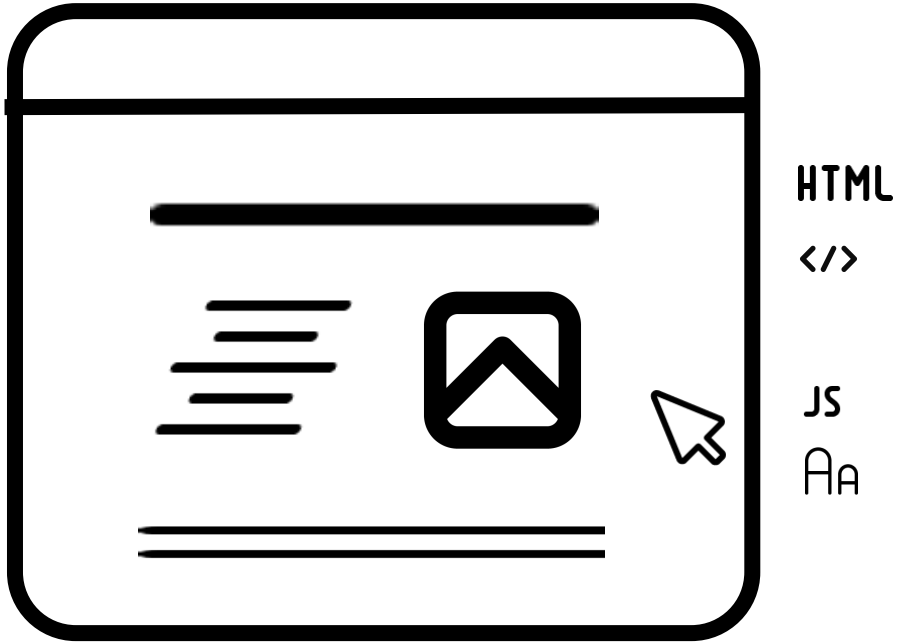
UNOPTIMIZED WEBSITE



long Largest Contentful Paint

4.5s

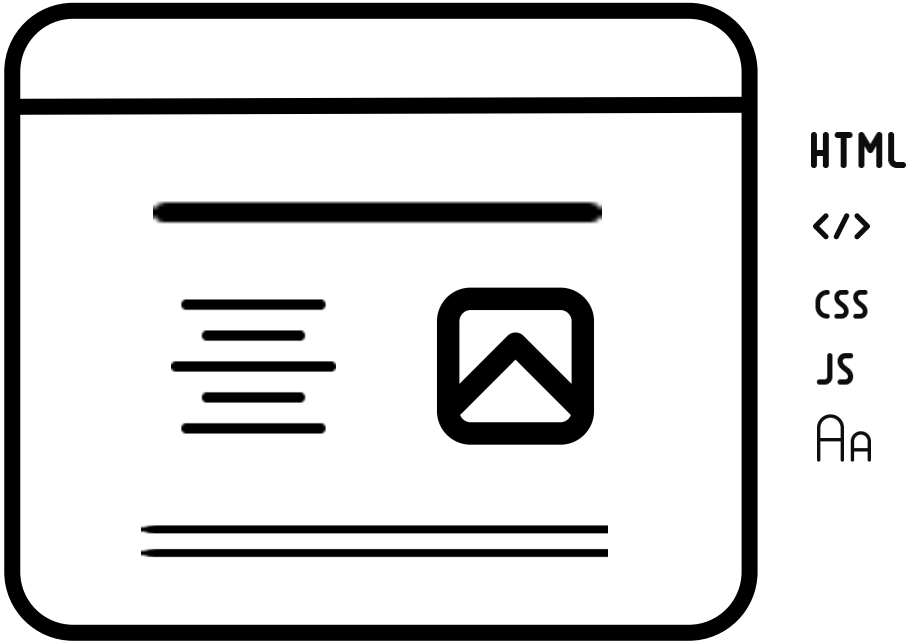
OPTIMIZED WEBSITE



4.5s

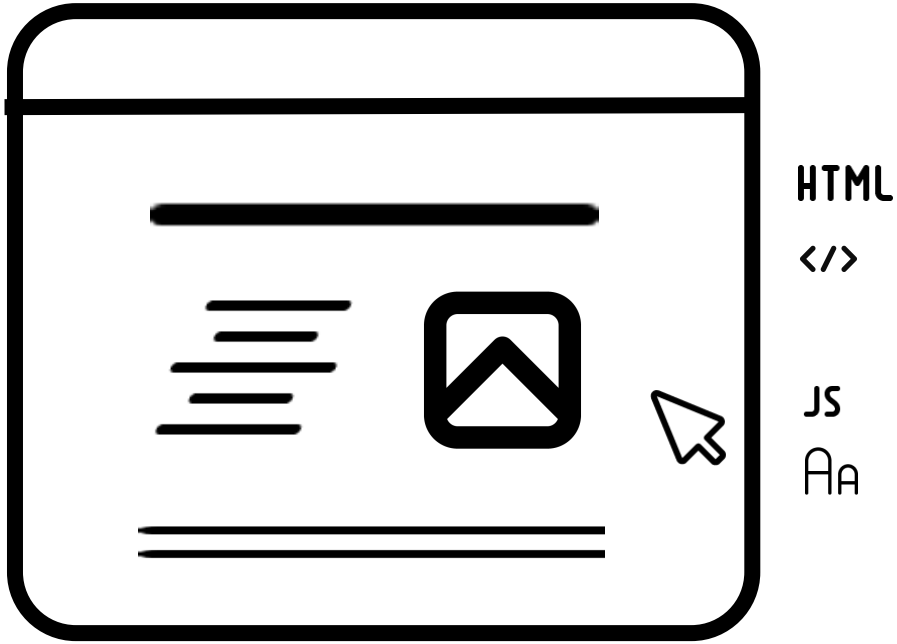
Gatsby

UNOPTIMIZED WEBSITE



5.0s

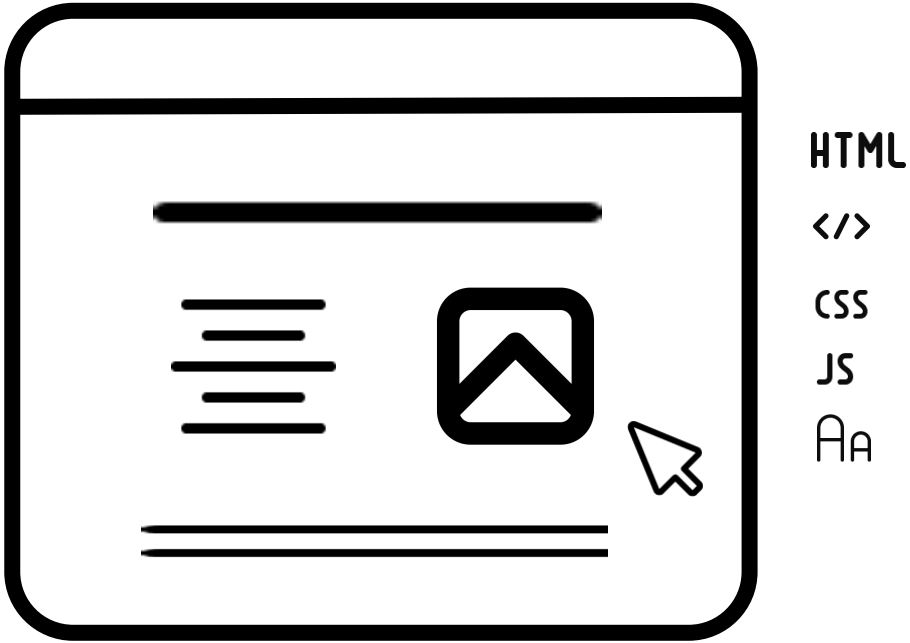
OPTIMIZED WEBSITE



5.0s

Gatsby

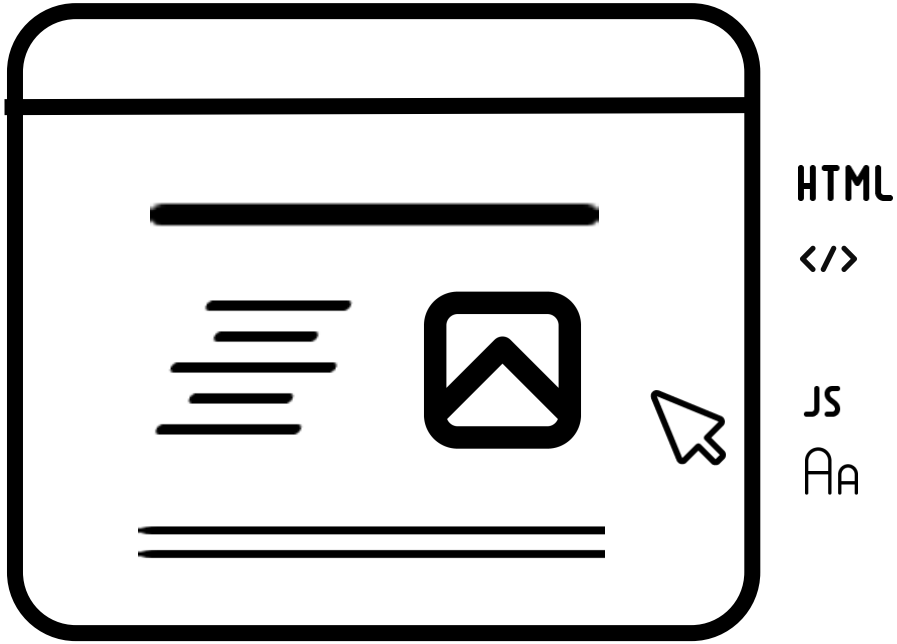
UNOPTIMIZED WEBSITE



(potential First Input Delay)

5.5s

OPTIMIZED WEBSITE



5.5s

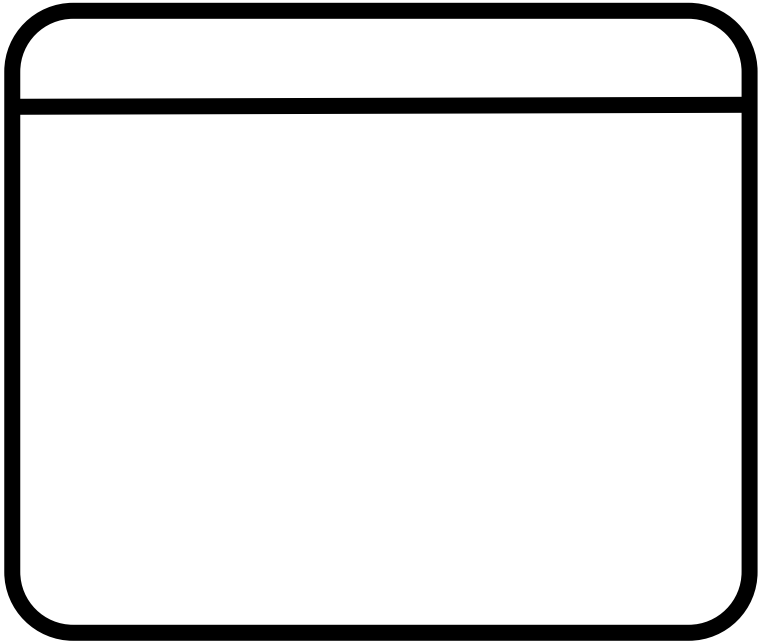
Gatsby

Seven steps to >=90% done

1. Use a CDN
2. Server-side render a “shell”
3. Inline styles*
4. Stop requests from blocking page load
5. Delay third-party scripts
6. Minimize JavaScript bundle size
7. Progressively load images

Gatsby

UNOPTIMIZED WEBSITE

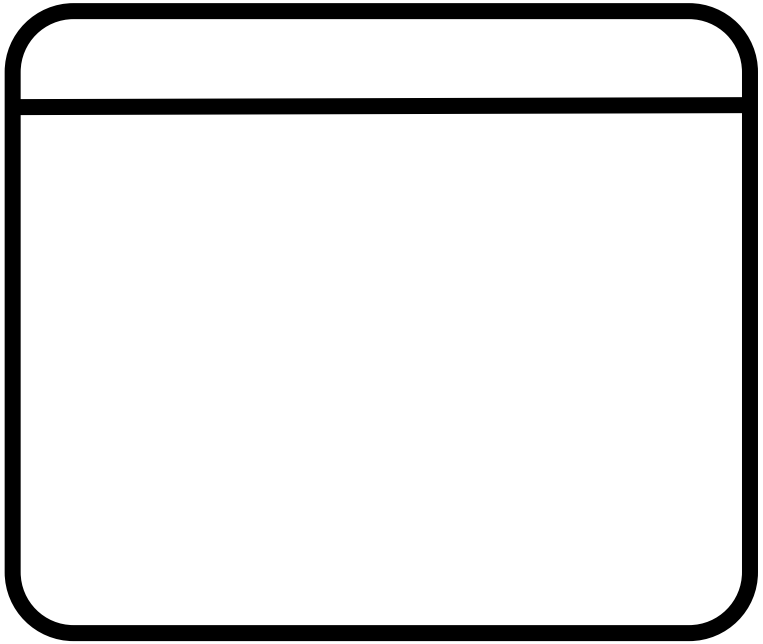


HTML

❌ Using CDN

0s

OPTIMIZED WEBSITE



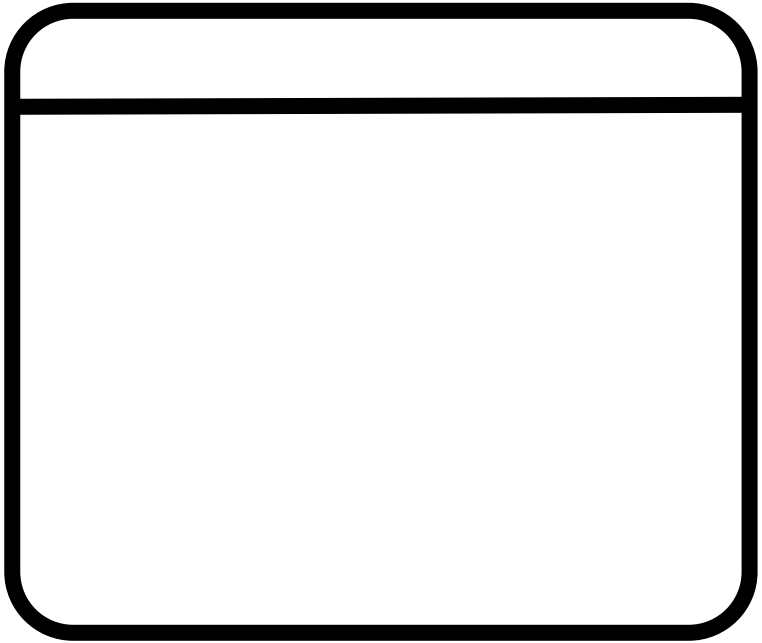
HTML

✅ Using CDN

0s

Gatsby

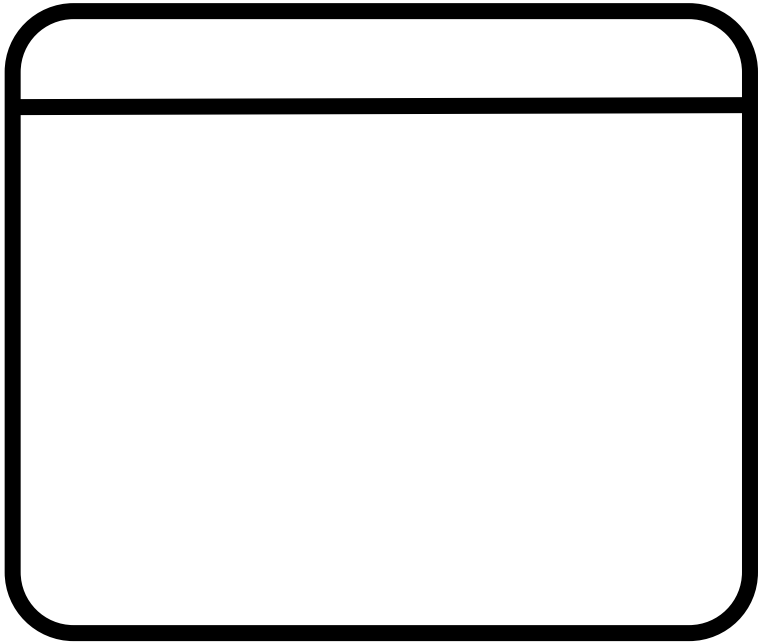
UNOPTIMIZED WEBSITE



HTML

0.5s

OPTIMIZED WEBSITE

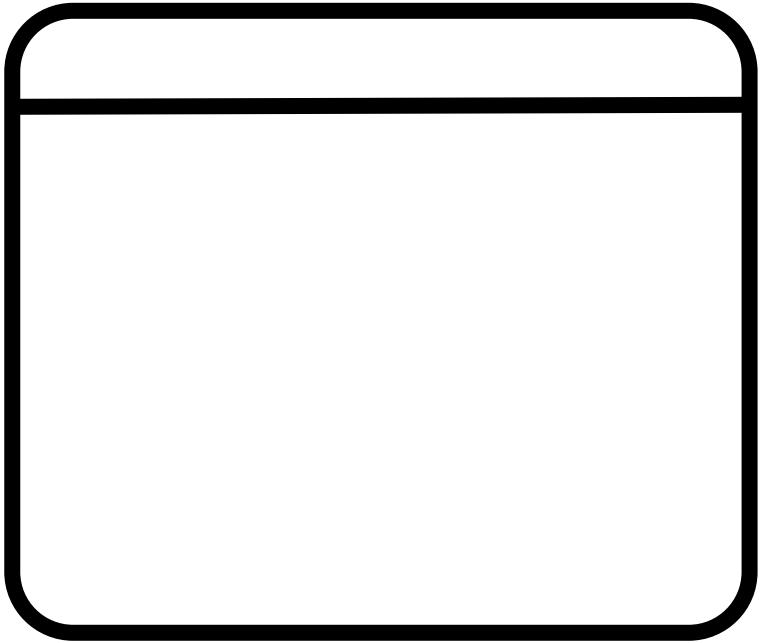


HTML

0.5s

Gatsby

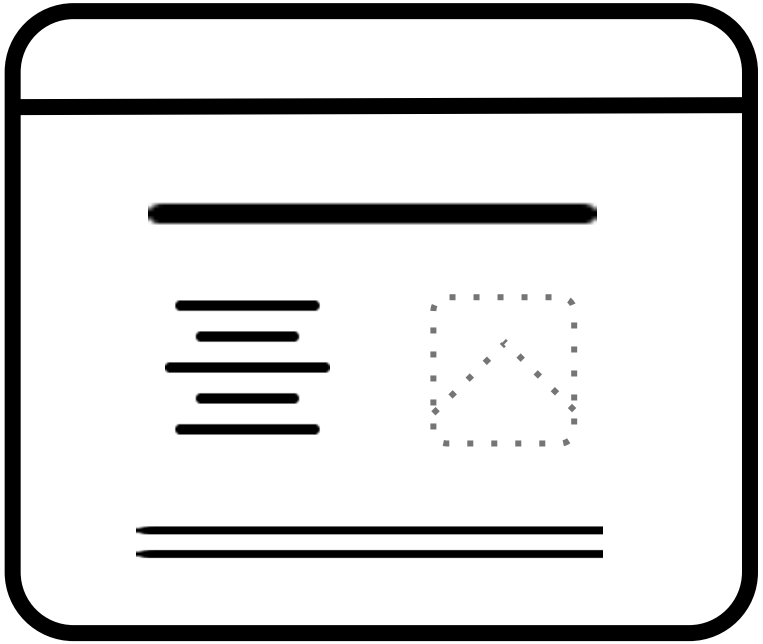
UNOPTIMIZED WEBSITE



HTML

1s

OPTIMIZED WEBSITE



HTML

JS

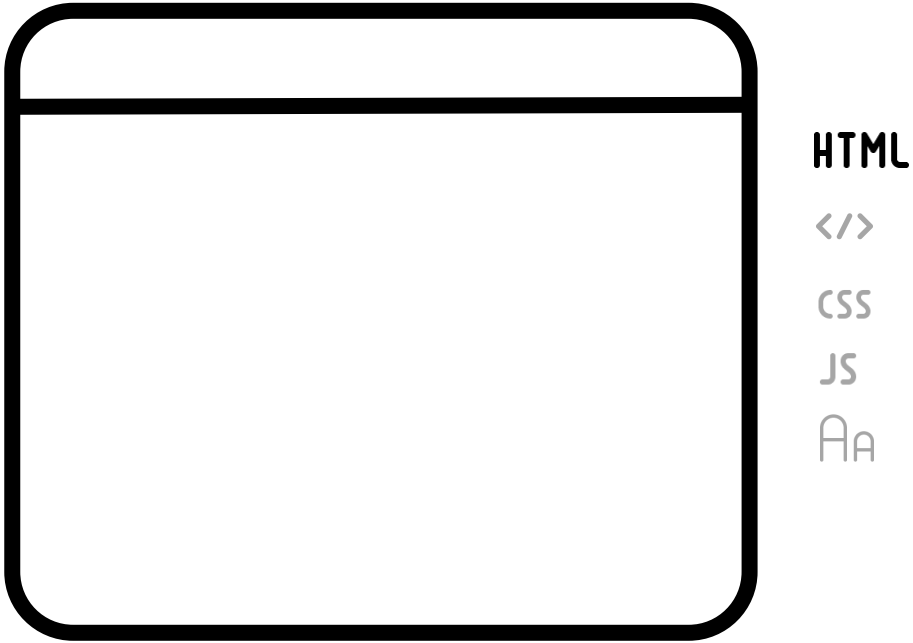
Aa

✅ Render page shell
✅ Inline critical CSS

1s

Gatsby

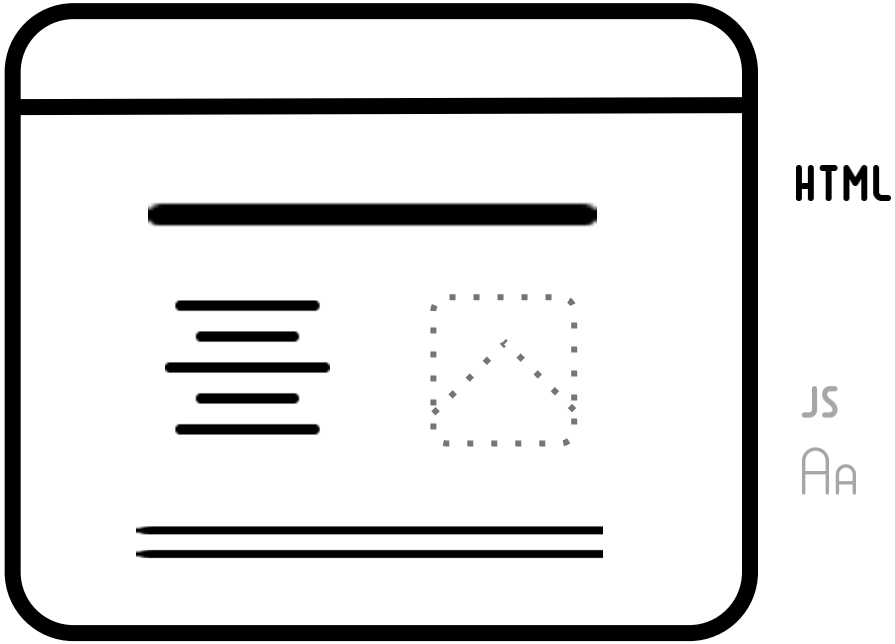
UNOPTIMIZED WEBSITE



❌ Render page shell
❌ Inline critical CSS

1.5s

OPTIMIZED WEBSITE

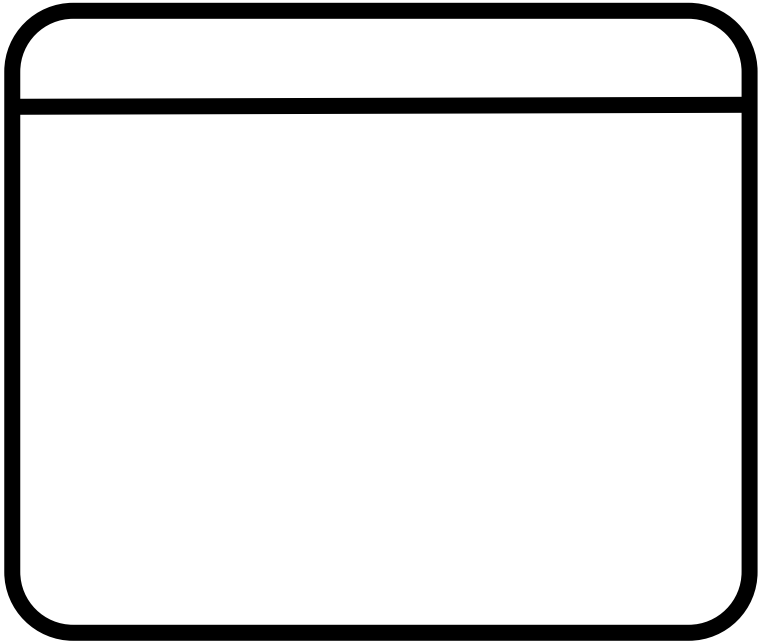


✅ Progressively load images

1.5s

Gatsby

UNOPTIMIZED WEBSITE



HTML

</>

CSS

JS

Aa

❌ Requests don't block page load
❌ Delay third-party scripts

2s

OPTIMIZED WEBSITE

HTML

JS

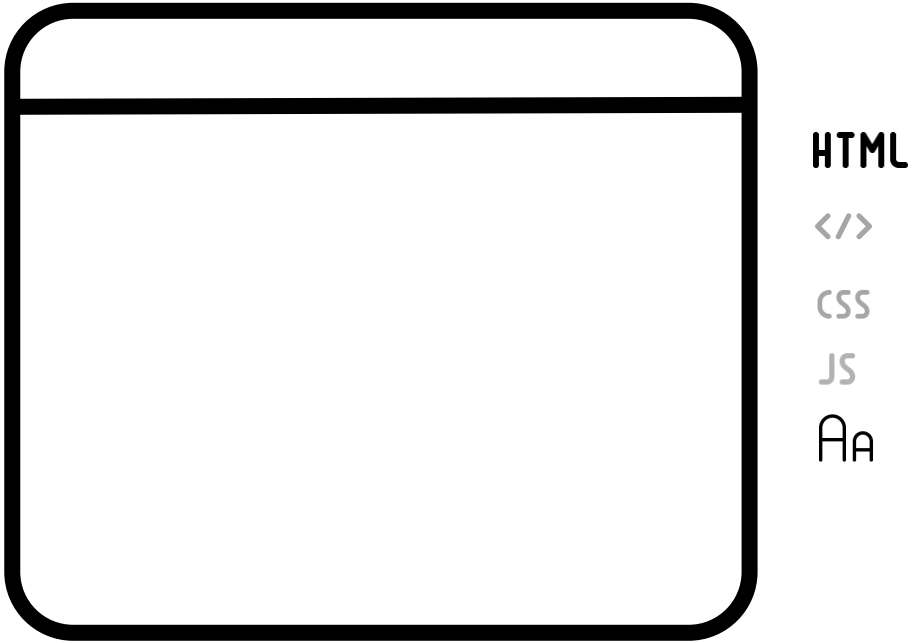
Aa

✅ Requests don't block page load
✅ Delay third-party scripts

2s

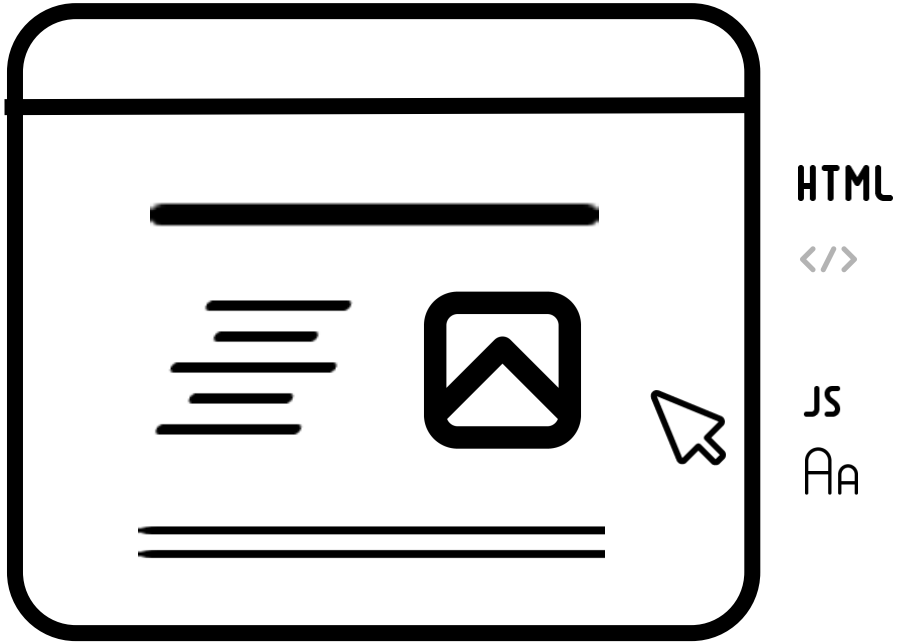
Gatsby

UNOPTIMIZED WEBSITE



2.5s

OPTIMIZED WEBSITE

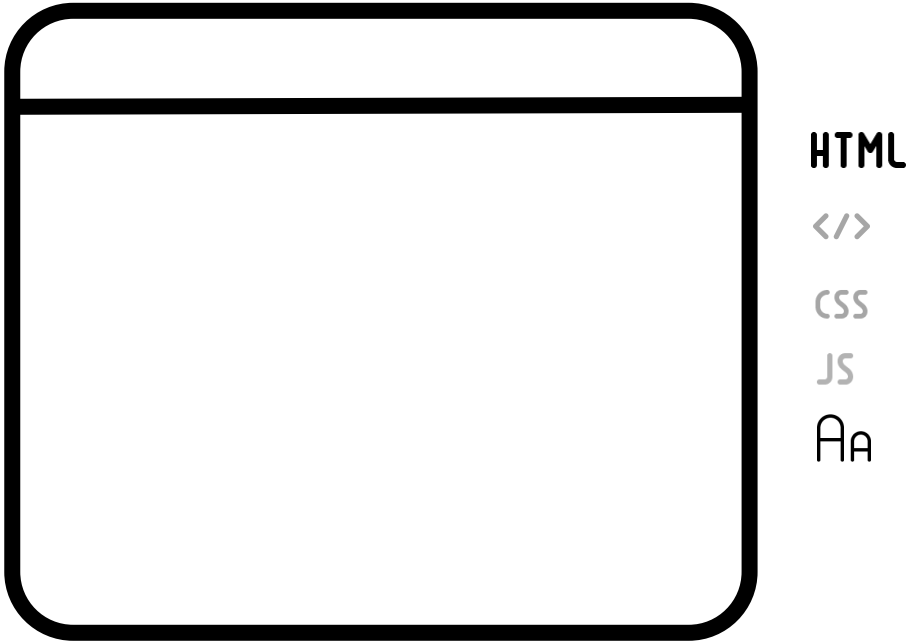


✅ Minimize bundle size

2.5s

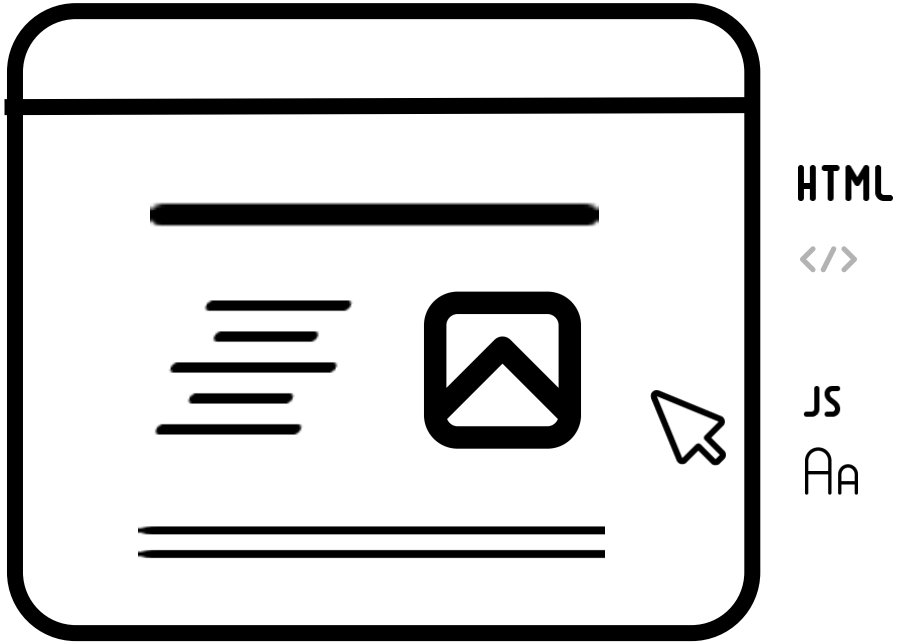
Gatsby

UNOPTIMIZED WEBSITE



3s

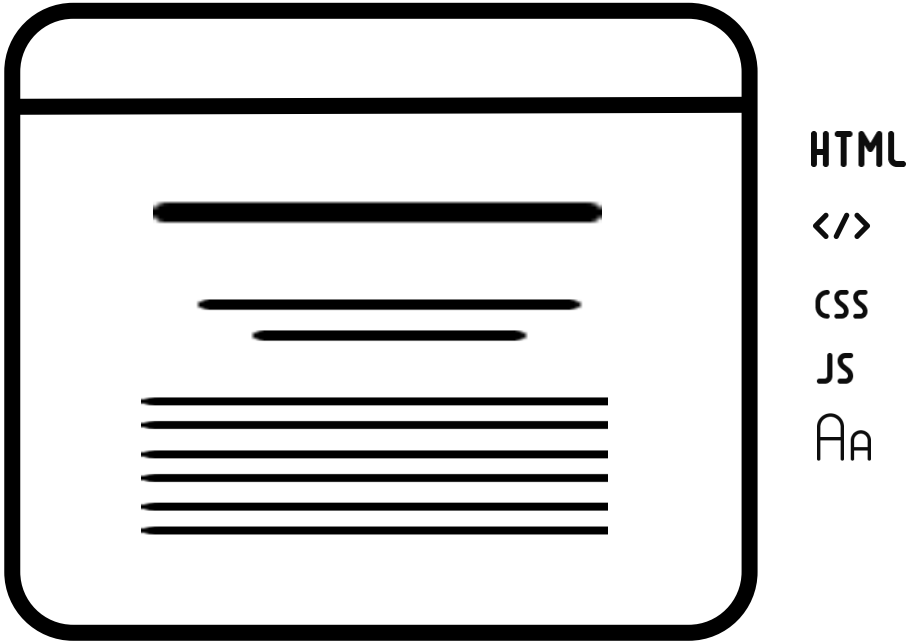
OPTIMIZED WEBSITE



3s

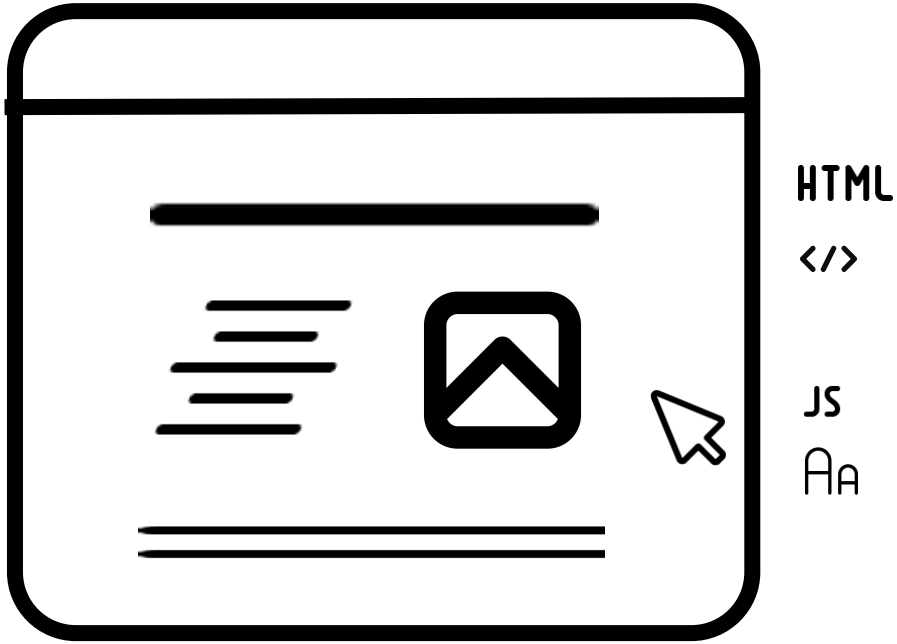
Gatsby

UNOPTIMIZED WEBSITE



3.5s

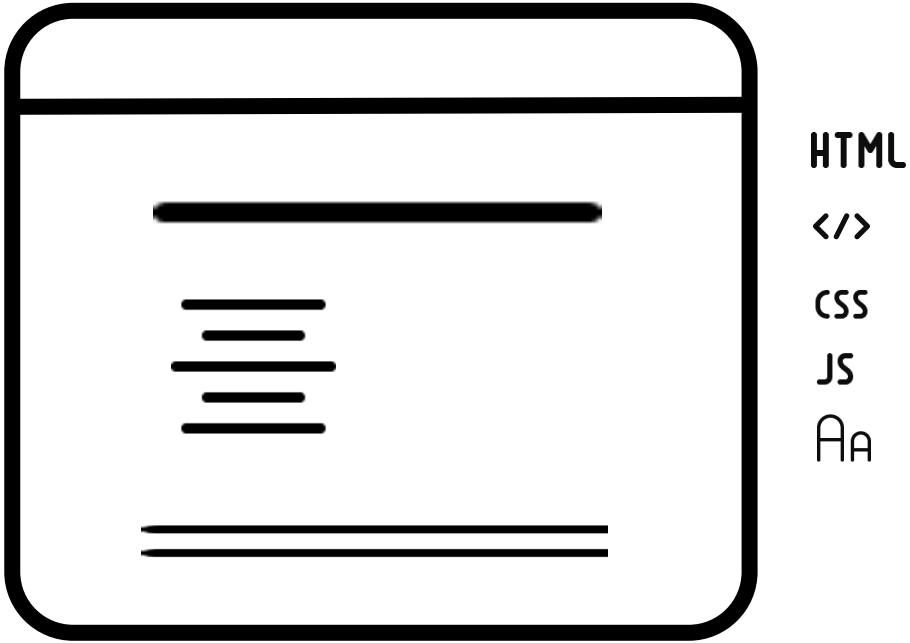
OPTIMIZED WEBSITE



3.5s

Gatsby

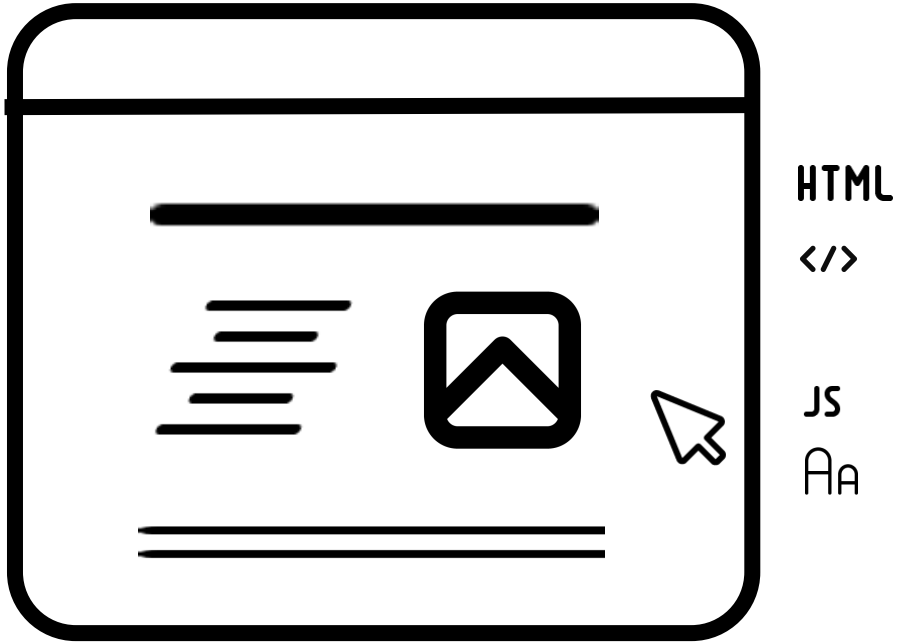
UNOPTIMIZED WEBSITE



❌ Progressively load images

4.0s

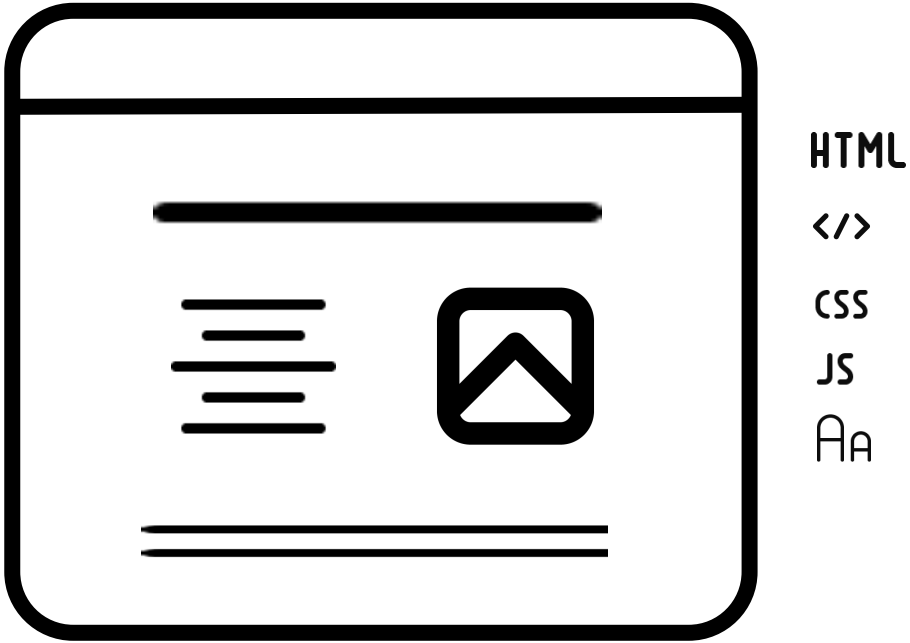
OPTIMIZED WEBSITE



4.0s

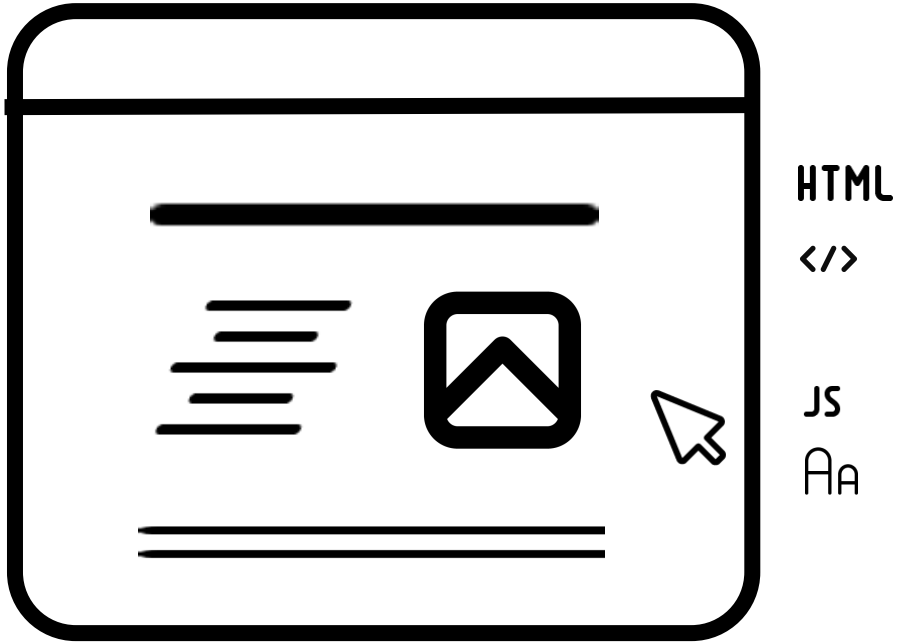
Gatsby

UNOPTIMIZED WEBSITE



4.5s

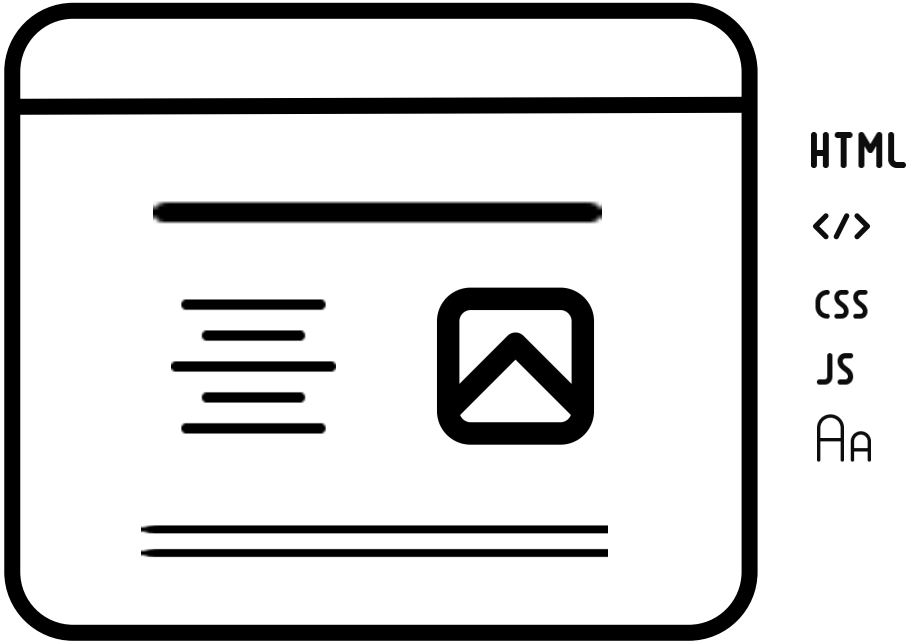
OPTIMIZED WEBSITE



4.5s

Gatsby

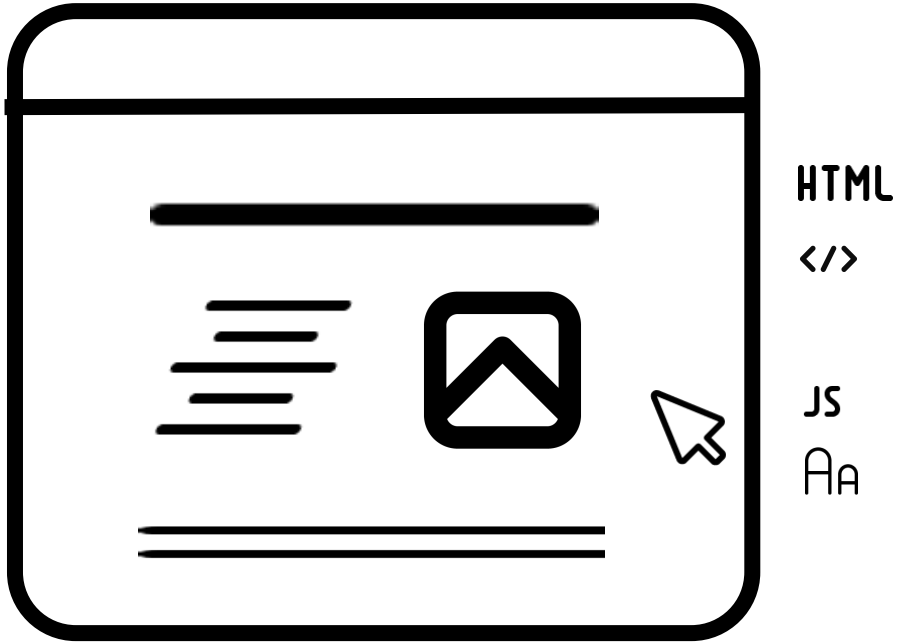
UNOPTIMIZED WEBSITE



❌ Minimize bundle size

5.0s

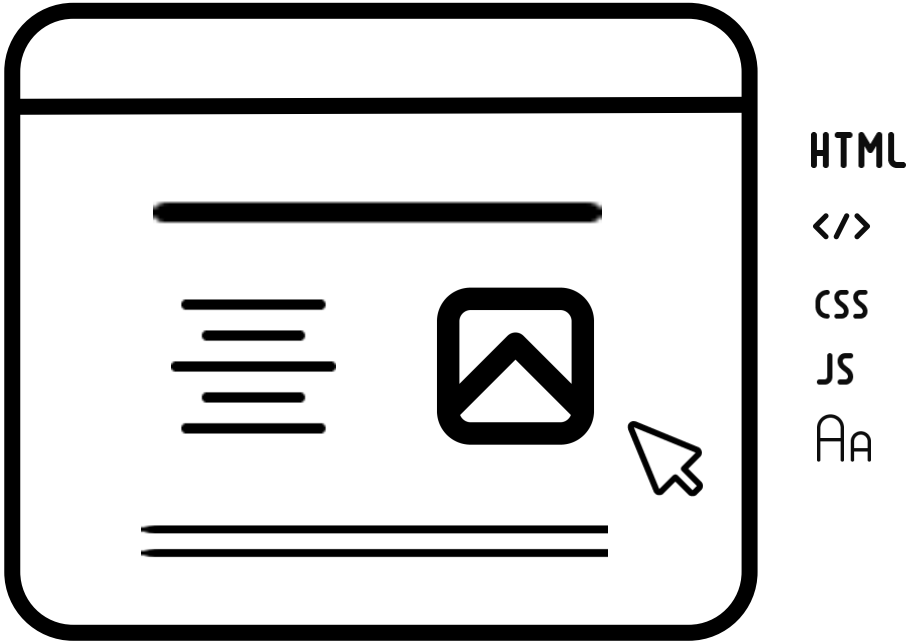
OPTIMIZED WEBSITE



5.0s

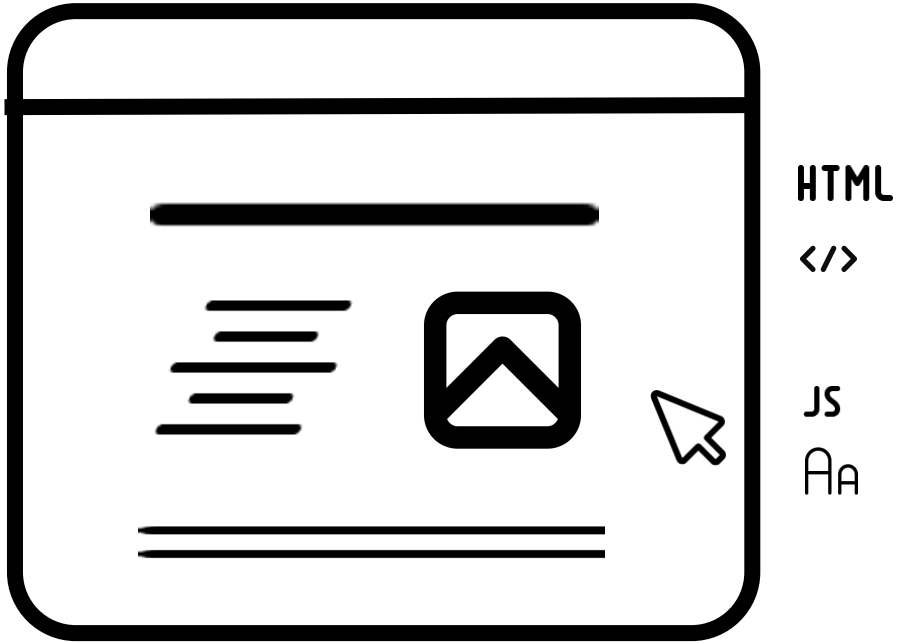
Gatsby

UNOPTIMIZED WEBSITE



5.5s

OPTIMIZED WEBSITE



5.5s

Gatsby

How each step affects your Core Web Vitals

Performance Optimization	LCP	FID	CLS
Use a CDN	✔		
Server-side render a “shell”	✔		✔
Inline styles	✔		✔
Requests don’t block page load	✔		
Delay third-party scripts	✔	✔	
Minimize JavaScript bundle size	✔	✔	
Progressively load images	✔		✔

Gatsby

How Gatsby helps you

Performance Optimization	How to do it with Gatsby
Use a CDN	**Makes easier**. Use Gatsby Cloud, Netlify, Cloudflare, etc
Server-side render a “shell”	**Out of the box.** No work required
Inline styles	**Out of the box.** No work required
Requests don’t block page load	**Makes easier.** Reduces “chaining” problem.
Delay third-party scripts	**Makes easier**. Use onInitialClientRender in gatsby-ssr.js
Minimize JavaScript bundle size	**Makes easier**. Use gatsby-plugin-perf-pbudgets to audit
Progressively load images	**Out of the box**. Use gatsby-plugin-image.

Gatsby

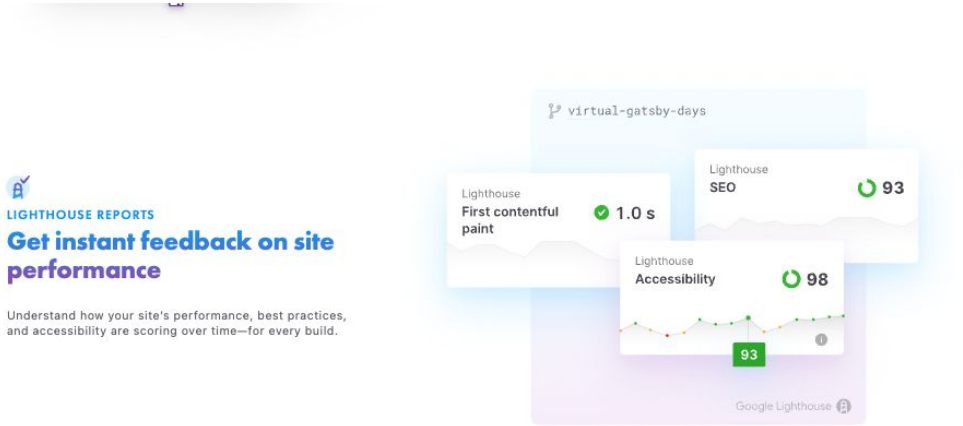
How Gatsby helps you

Performance Optimization	How to do it with Gatsby
Use a CDN	**Makes easier**. Use Gatsby Cloud, Netlify, Cloudflare, etc
Server-side render a “shell”	**Out of the box.** No work required
Inline styles	**Out of the box.** No work required
Requests don’t block page load	**Makes easier.** Reduces “chaining” problem.
Delay third-party scripts	**Makes easier**. Use onInitialClientRender in gatsby-ssr.js
Minimize JavaScript bundle size	**Makes easier**. Use gatsby-plugin-perf-pbudgets to audit
Progressively load images	**Out of the box**. Use gatsby-plugin-image.

Gatsby

Continual performance monitoring

- Use per-commit, per-PR perf monitoring to keep track of these vitals over time.



Gatsby

Summary

- Develop an intuition for yourself
- Use tools for others
- Get the 7 key types of performance optimization down

Gatsby

Questions?

Common Speed Improvements

- Enable compression
- Minify CSS, JS, + HTML
- Fonts, fonts, fonts!
- Reduce Redirects
- Remove Render-Blocking JS
- Leverage Browser Caching
- Improve Server Response Time
- Use a CDN
- Properly size and load images
- Avoid enormous network payloads
- Serve static assets with an efficient cache policy
- Avoid an excessive DOM size
- Avoid chaining critical requests