

RENDERING IS COOL

(again)

{Tame**the**Bots}

Hello, I am **Dave**

and I

- Have mucked around on the internet since last century.
- Am a web developer & tech seo at {Tame**the**Bots}.
- Am strangely fascinated with web rendering.

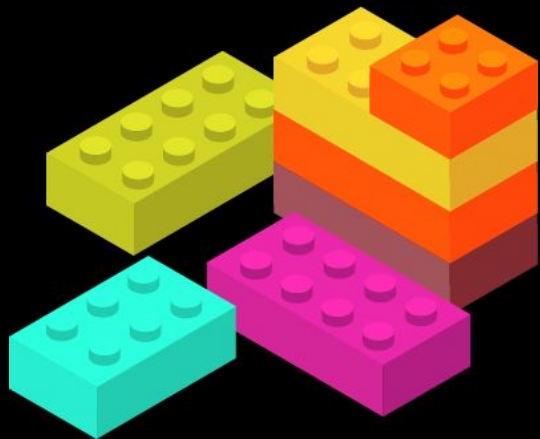


HELLO DAVE

What is **rendering** anyway?

It's the conversion of all the bits of html, CSS, JavaScript into a web page.

It's this.



Becoming this.

It's a few **different** steps.

1. **Loading**
The resources need to be downloaded across the network, or retrieved from cache
2. **Parsing**
Converting HTML & CSS into the Document Object Model (**DOM**) and the CSS Object Model (**CSSOM**)
3. **Construction**
The DOM and CSSOM are used to construct the **Render Tree**, the elements that need to be displayed
4. **Layout**
Based on the render tree, the size and position of each element is calculated
5. **Paint / Composition**
Colours, styles, and images are applied, assembled into layers and displayed to the user.

But what about JavaScript?

JavaScript can change the DOM and CSSOM, meaning all the following steps need to happen again to reflect the change.

The JavaScript needs to be (*like HTML and CSS*):

- Downloaded
- Parsed

And then it needs to be:

But what about JavaScript?



EXECUTED

(really, this just means running the script)



But what about JavaScript?

After execution, if any changes are made to the DOM or CSSOM,
The rest of the rendering pipeline happens again.



Why do we focus so much on JavaScript?

For us SEO types, JavaScript is the thing that can add the content we want to:

- Be indexed for
- Be found for

*JavaScript dominates the conversation,
because it's the bit that impacts the job we do the most*

Server-Side vs. Client-Side

Server-Side Rendering (SSR)

The html with all the content is sent from the server complete.



Client-Side Rendering (CSR)

Some or all of the content is sent separately and JavaScript puts it together.



Server-Side vs. Client-Side

Server-Side Rendering (**SSR**)

- Can be static html files
- Could be more dynamic, but put together on the server before being sent over (e.g. WordPress)

Client-Side Rendering (**CSR**)

- Can be all the content, with just a minimal shell sent across
- Can be just bits of content on a page

Google are pretty good at **CSR!**

- Google switched to 'evergreen' chromium (keeps up-to date with actual chrome) in 2019
- Before that it was stuck on an old (version 41) version
- Have a whole bunch of documentation and videos about it
developers.google.com/search/docs/crawling-indexing/javascript/javascript-seo-basics
- Have a **Martin Splitt!**
linkedin.com/in/martinsplitt/

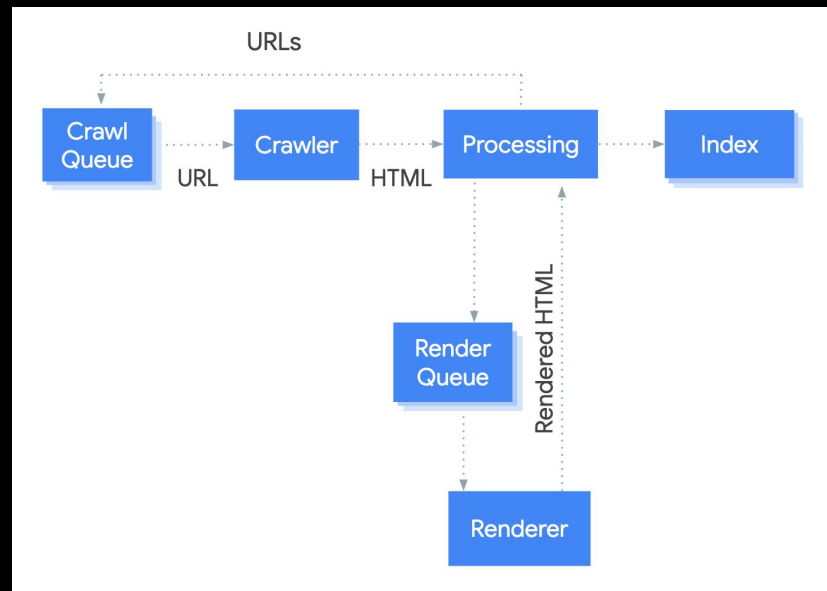


Google are pretty good at CSR!

They pass pages through their

Web Rendering Service

- It's *ALL* pages, not just JavaScript ones
- The delay is mins/hours, not days/weeks
- A major difference between a browser and the WRS is it uses the crawling infrastructure as a network layer, and relies heavily on their caching



Google are pretty good at CSR!

There are still gotchas and foibles:

- Be careful blocking api endpoints and JavaScript resources with robots.txt
- Doesn't use more esoteric things like service workers, and doesn't play nicely with others like web workers
- There are more moving parts, and more moving parts can mean more chance to break.

Bing are OK at **CSR!**

Bing are also pretty competent at CSR

- Render using an evergreen(ish) version of Edge, a chromium based browser
- Less documented process, but seems to work in a similar way to Google
- Seems a little slower and more fragile, (*anecdotally!*)
- Same gotchas and increased fragility apply

So it's been okish for 7 odd years?

Kinda.

Usually.

Wait, that's not sounding cool again...

So it's been okish for 7 odd years?

OK-ish is doing some heavy lifting.

Whilst Google and Bing are mostly ok, when it goes wrong, it can go REALLY wrong.

Those gotchas can mean your site is empty to a Bot

Test with:

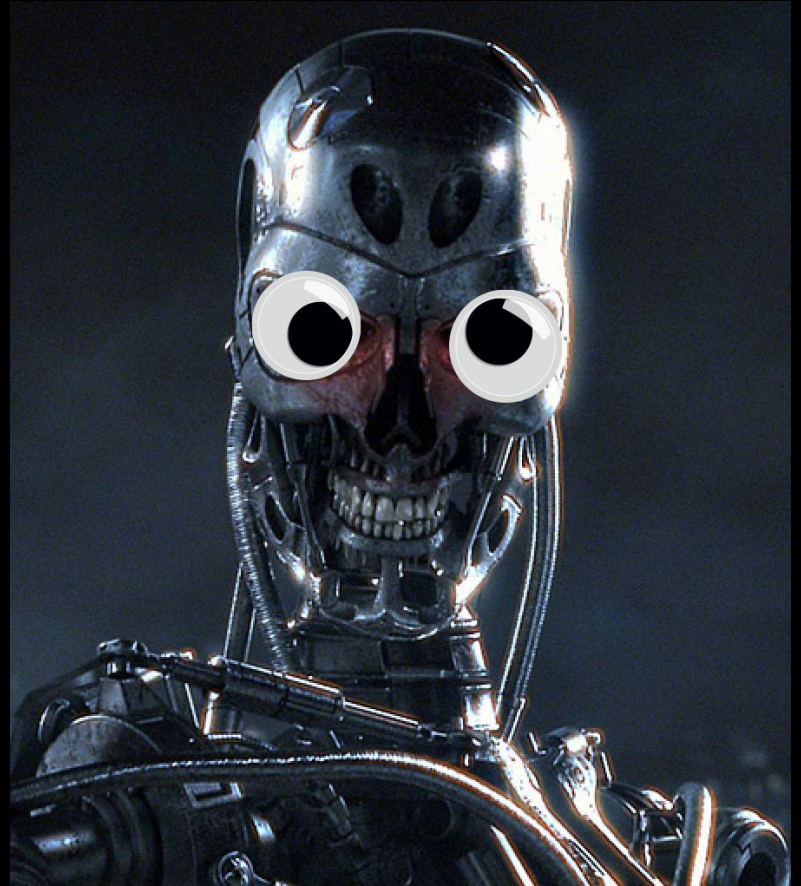
- URL inspection tool
- Bing's URL inspector
- Searching for the content

Rise of the machines

OH NO. A.I.

Our new, smart, but also very dumb,
overlords.

*If John Connor was inserted via JavaScript, we'd all have
been safe anyway.*



LLMs fetch content too

Three major functions that fetch your pages:

- **Training**
Gathering content to train their models on
- **Retrieval Augmentation (RAG) / Grounding**
To improve a generated answer by fetching up-to-date content that might not be in the training
- **Analyse**
Fetch a URL to analyze if someone asks specifically about it; "*Please provide a summary of this page*"

LLMs fetch content too

But with the exception of Google*:

No JavaScript was executed
in the making of this generated answer

**Google use their indexing pipeline, which can and does render JavaScript*

LLMs fetch content too

If you want your content to surface and be usable by the LLMs

You *NEED* SSR

Not all SSR is equal

Next.js introduced streaming

- The content is in the initial HTML
- But it can be in a different order to where it actually goes on screen
- Requires JavaScript to shuffle into order

Can change the overall structure of a page for consumers that don't execute JavaScript

<https://tamethebots.com/blog-n-bits/out-of-order-streaming-and-llms>



The new hotness vs. reality

How important leadership think LLMs visibility is:



How important it might actually be:



A **tiny** proportion of the traffic

According to Cloudflare's Radar report, between 7th and 14th June 2026

For visits that had a referrer:

- 89% of referrals were from Google
- 0.23% were from ChatGPT
- 0.1% were from Claude
- 0.01% were from Perplexity

<https://radar.cloudflare.com/explorer?dataSet=bots.crawlers&groupBy=referrer&timeCompare=1>

A **tiny** proportion of the traffic

SSR could simultaneously be:

- The Right Solution
- Not "move the needle" much

When it comes to traffic from the chat bots.

A tiny, but **important** proportion of the traffic?

The juice might still be worth the squeeze

- Not all sites are equal, you might get a greater proportion of your traffic from these.
- Look at your log files and analytics for referrers.

And:

- SSR is generally more robust for 'traditional search' anyway.
- It's generally more robust for actual users.
- You might find it easier to get budget by tying it to A.I. performance.

Turn **off** the JavaScript to test

Most major browsers allow you to disable JavaScript i.e.

<https://developer.chrome.com/docs/devtools/javascript/disable>

You can disable JavaScript in my Fetch & Render tool:

<https://tamethebots.com/tools/fetch-render>

A.I. is doing **more** though

Vibe-Coding is no longer just something done when putting together the Ann Summers Catalog.

Name *	Jumbo Sexy Fun Time Toy
SKU *	EYEWATER-3000

A tendency towards JavaScript frameworks

The major coding agents often reach for well documented frameworks like React to create the pages and site you ask for.

They might be:

- CSR apps
- Meta frameworks like Next.js
- Some completely weird custom thing.

A tendency towards JavaScript frameworks

Many of these will be newer sites

- Built by founders with minimal developer experience
- No significant developer oversight until now
- May well face bigger issues than the SSR / CSR debate

What might have been WordPress or a builder like Wix in the past might now be more complex, where rendering *might* be an issue again.

We're **BACK** baby!

With new:

- Breed of Bots & Consumers
- Site owners and founders
- Methods of SSR

Rendering **IS** cool again



Thank you very much!

I have been Dave

You can find me at:

- tamethebots.com
- <https://bsky.app/profile/tamethebots.com>
- <https://www.linkedin.com/in/davewsmart/>