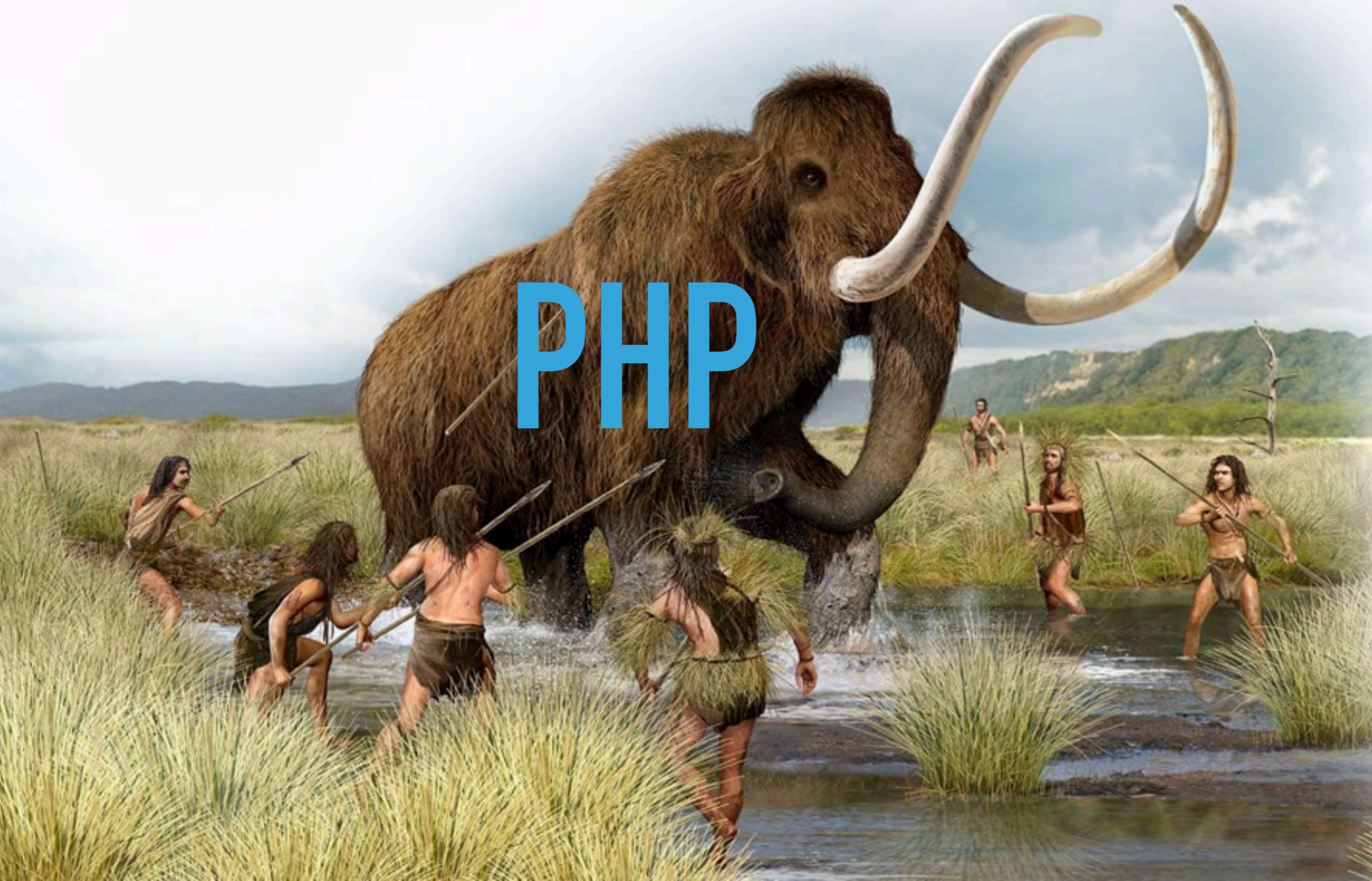


NO INÍCIO DOS TEMPOS...

PHP







*ember*



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <link rel="shortcut icon" href="/favicon.ico">
6     <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
7     <meta name="theme-color" content="#000000">
8     <link rel="manifest" href="/manifest.json">
9     <title>React App</title>
10  </head>
11  <body>
12    <div id="root"><!-- insert app here --></div>
13
14    <script src="/static/js/bundle.js"></script><script
src="/static/js/0.chunk.js"></script><script
src="/static/js/main.chunk.js"></script></body>
15 </html>
16
```









meu site lindão

All

Images

Videos

Maps

News

More

Did you mean:

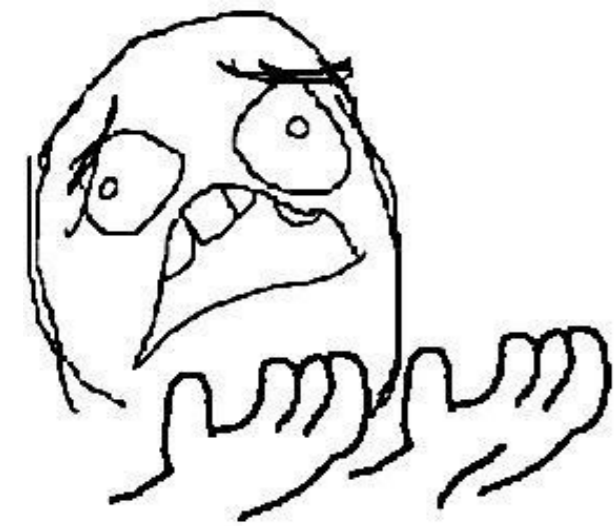
***site que não pode ser indexado direito?***

No results containing all your search terms were found.

Your search - **meu site lindão** - did not match any documents.

Suggestions:

- Make sure that all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Server side render your stuff.





OTIMIZANDO AS EXPERIÊNCIAS COM

---

**SERVER-SIDE RENDER**



EU!

# ARTHUR REIS PUTHIN

- ▶ Senior Interactive Developer
  - ▶ Isobar IWS Brazil
- ▶ Tentando terminar a faculdade
- ▶ 3ª vez no TDC!
- ▶ Gosto muito de JS. Mesmo!



aputhin



aputhin\_





MAS DE NOVO...

QUAL É O PROBLEMA DE

CLIENT-SIDE RENDER?

A.K.A. CSR

Infographics are **visual** content that captures the attention and imagination of your audience. As well as drive a successful digital marketing campaign through Search Engine Optimization

# SEO

- 1 INTEREST GROWS AS IT IS ATTRACTIVE TO VISITORS
- 2 BUILD TARGETED AUDIENCE

INTEREST IS A VERY IMPORTANT PART OF A CONTENT THAT CAN BE IN DIGITAL MARKETING TOOL. People tend to get more attracted to content with visual elements rather than plain text based content.

50% OF PEOPLE GAIN FACT WITH VISUAL INFORMATION. Most like the graphs, charts, patterns, explained facts and other graphical elements can do a lot better to capture the attention of target audience.



- 3 CONTENT GOES VIRAL
- 4 CREATING GREATER BRAND AWARENESS

If you can combine the power of text, graphics and photos with compelling infographics you will increase in volume than by writing it to viral.

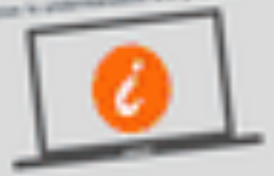
INCREASE YOUR BRAND AWARENESS BY CREATING INNOVATIVE INFOGRAPHICS. The web will make your audience understand the importance of your brand in a different way.



- 5 BETTER UNDERSTANDING OF CONTENT
- 6 PEOPLE LINK BACK WITH CREDITS

READ BY THE CUSTOMERS ARE IMPORTANT TO DISPLAY YOUR CONTENT. Infographics are also easy to consume by according to content clear and concise bits of information to understandable and graphical appealing way.

INFOGRAPHICS PLAY A MAJOR ROLE IN MAKING YOUR CONTENT VIRAL THROUGH SOCIAL NETWORKING SITES. Infographics can help you generate more likes, comments and shares and further social networking sites that other are important role in increasing your SEO efforts.



- 7 MAKE VISITORS STAY LONGER
- 8 PORTABLE OR CAN BE EMBEDDED BY FELLOW BLOGGERS.

INFOGRAPHICS ARE BECOMING TO MAKE VISITORS STAY LONGER TO INCREASE OVER SALES. Infographics will make content stick to their screens for longer time thereby increasing the bounce rate and increasing the average time duration.

INFOGRAPHICS CAN BE SHARED ON VARIOUS PLATFORMS AND WEBSITE & PROBABLY EMBEDDED WITH CODE BELOW IT. Infographics are shared by other bloggers & they then it they will give credit for the content to the page which is right to you.



- 9 PORTRAYS YOU AS AN EXPERT



# 1. SEARCH ENGINE OPTIMISATION

# SEO\*





MAIS DIFÍCIL  
mas  
POSSÍVEL (?)





## 2. USER EXPERIENCE

---

# UX

# UNCOMPRESSED JS

## WHAT DO POPULAR SITES SEND DOWN?

### DESKTOP WEB

Google Sheets **5.8MB**  
Facebook **7.1MB**  
GMail **3.9MB**  
LinkedIn **7.2MB**

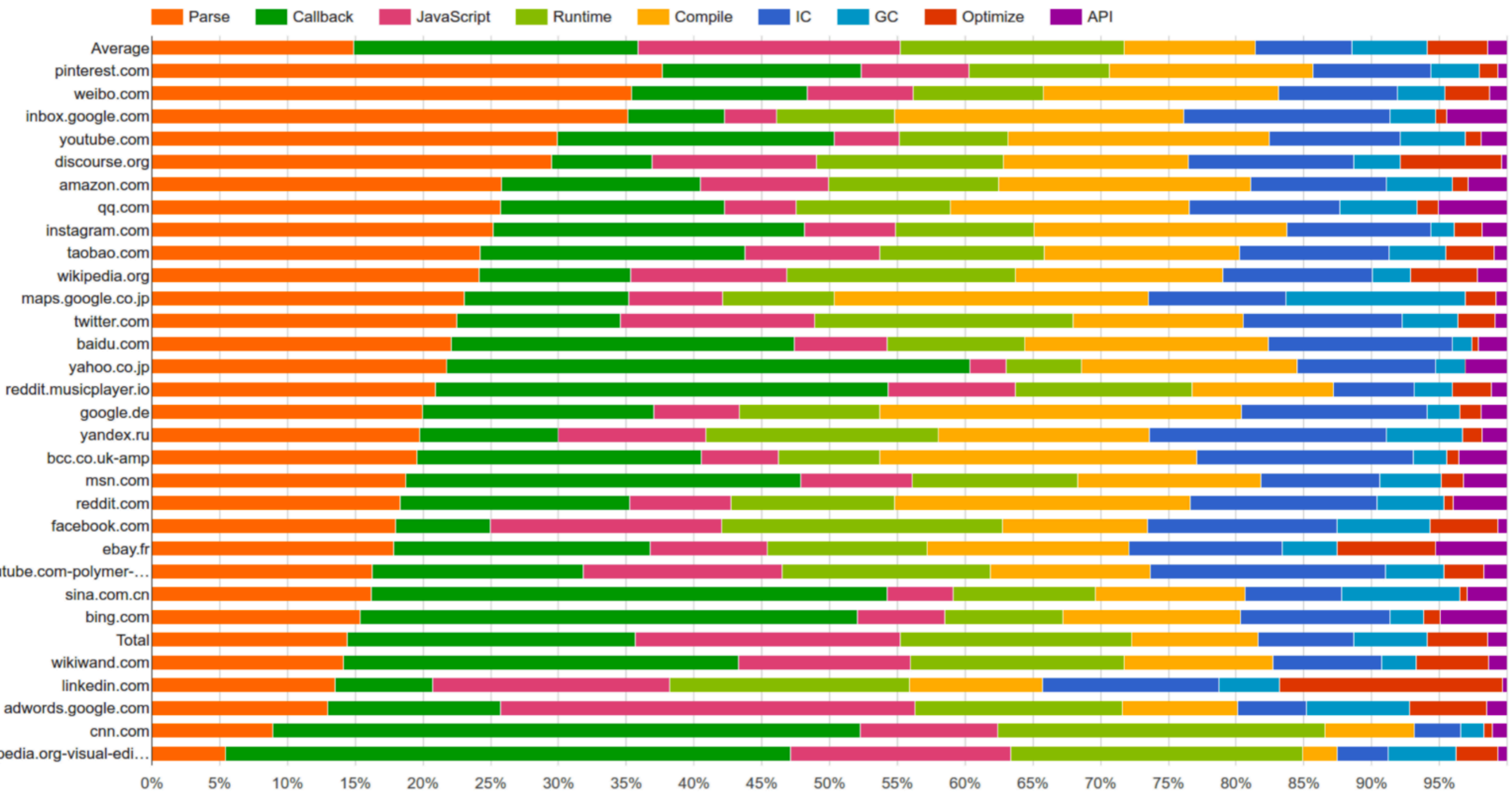
### MOBILE WEB

YouTube **1.4MB**  
Twitter **1.8MB**  
Facebook **1.8MB**  
Reddit **1.3MB**  
LinkedIn **6.2MB**



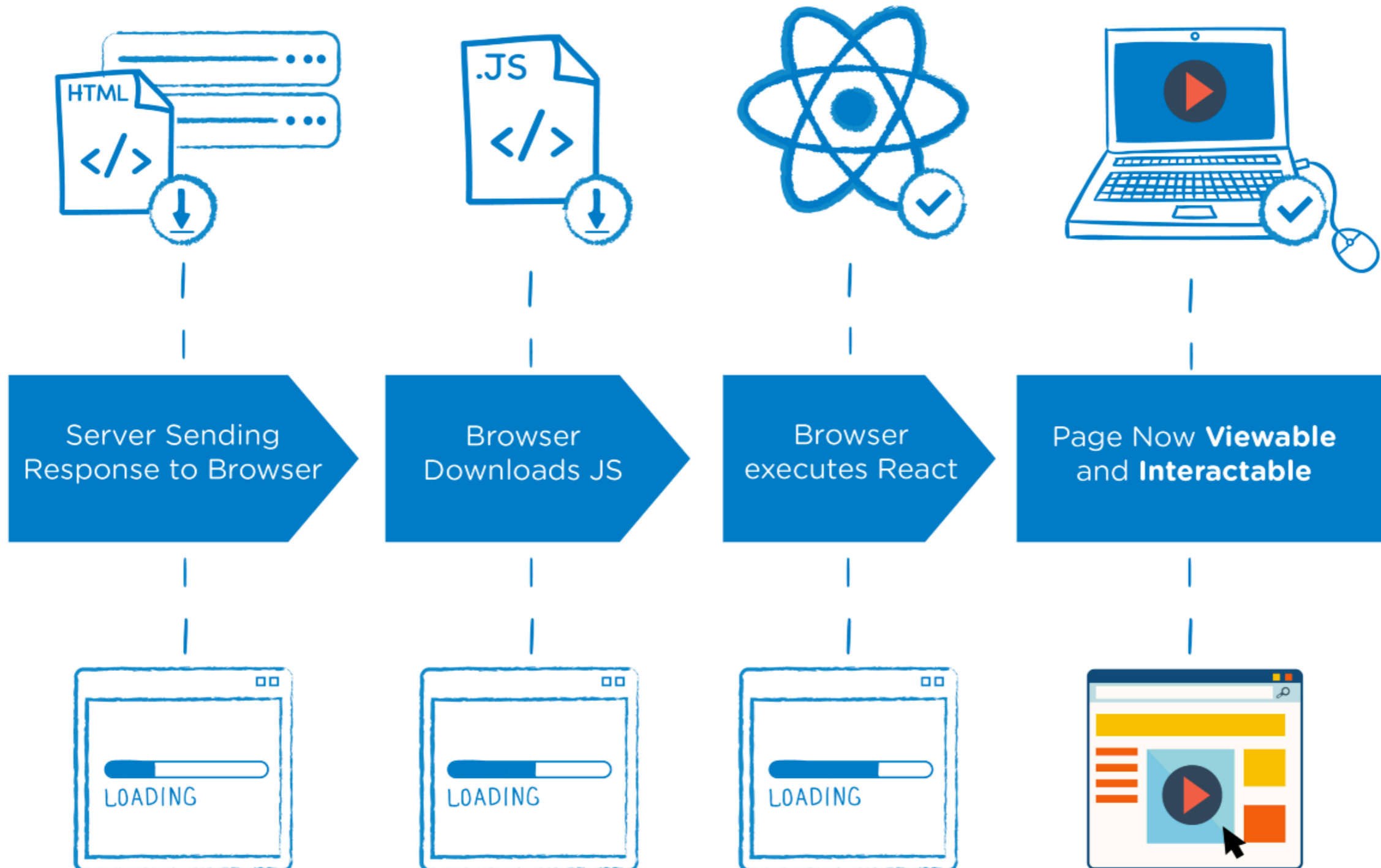
From 2017's "BlinkOn 9: Bringing Facebook.com and the Web up to speed"





**PARSE / COMPILER = 10-30% TIME SPENT  
IN V8 DURING PAGE LOAD**

# CSR





```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <link rel="shortcut icon" href="/favicon.ico">
6     <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
7     <meta name="theme-color" content="#000000">
8     <link rel="manifest" href="/manifest.json">
9     <title>React App</title>
10  </head>
11  <body>
12    <div id="root"><!-- insert app here --></div>
13
14    <script src="/static/js/bundle.js"></script><script
src="/static/js/0.chunk.js"></script><script
src="/static/js/main.chunk.js"></script></body>
15 </html>
16
```

# Promises - OTHER

Usage

% of all users 

Global

89.59% + 0.02% = 89.61%

A promise represents the eventual result of an asynchronous operation.

Current aligned Usage relative Date relative Apply filters Show all ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *
		2-26	4-31		10-18		
		27-28	32	3.1-7	19	3.2-7.1	
6-10	12-17	29-62	33-69	7.1-11.1	20-55	8-11.4	
11	18	63	70	12	56	12.1	all
		64-65	71-73	TP			





0.0s



0.3s



0.8s

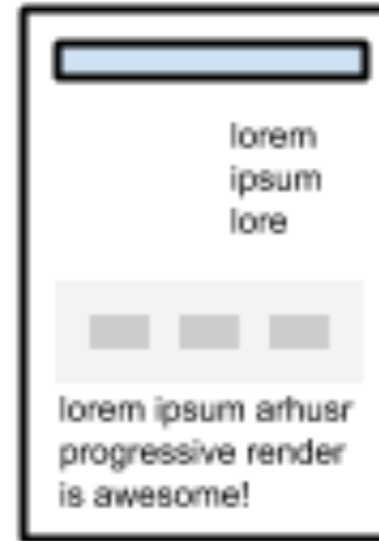
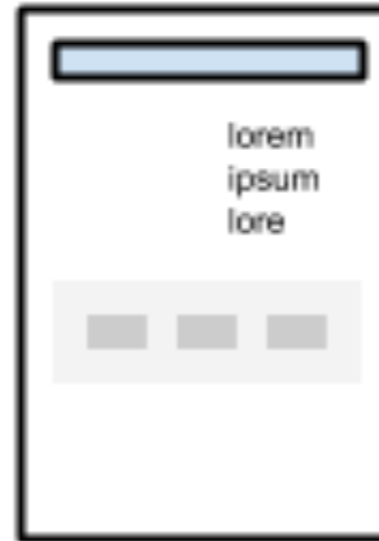
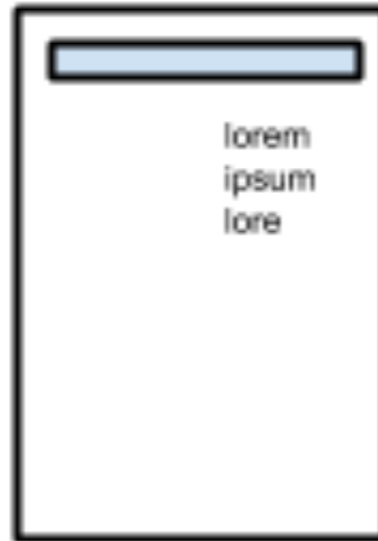


1.2s



1.5s

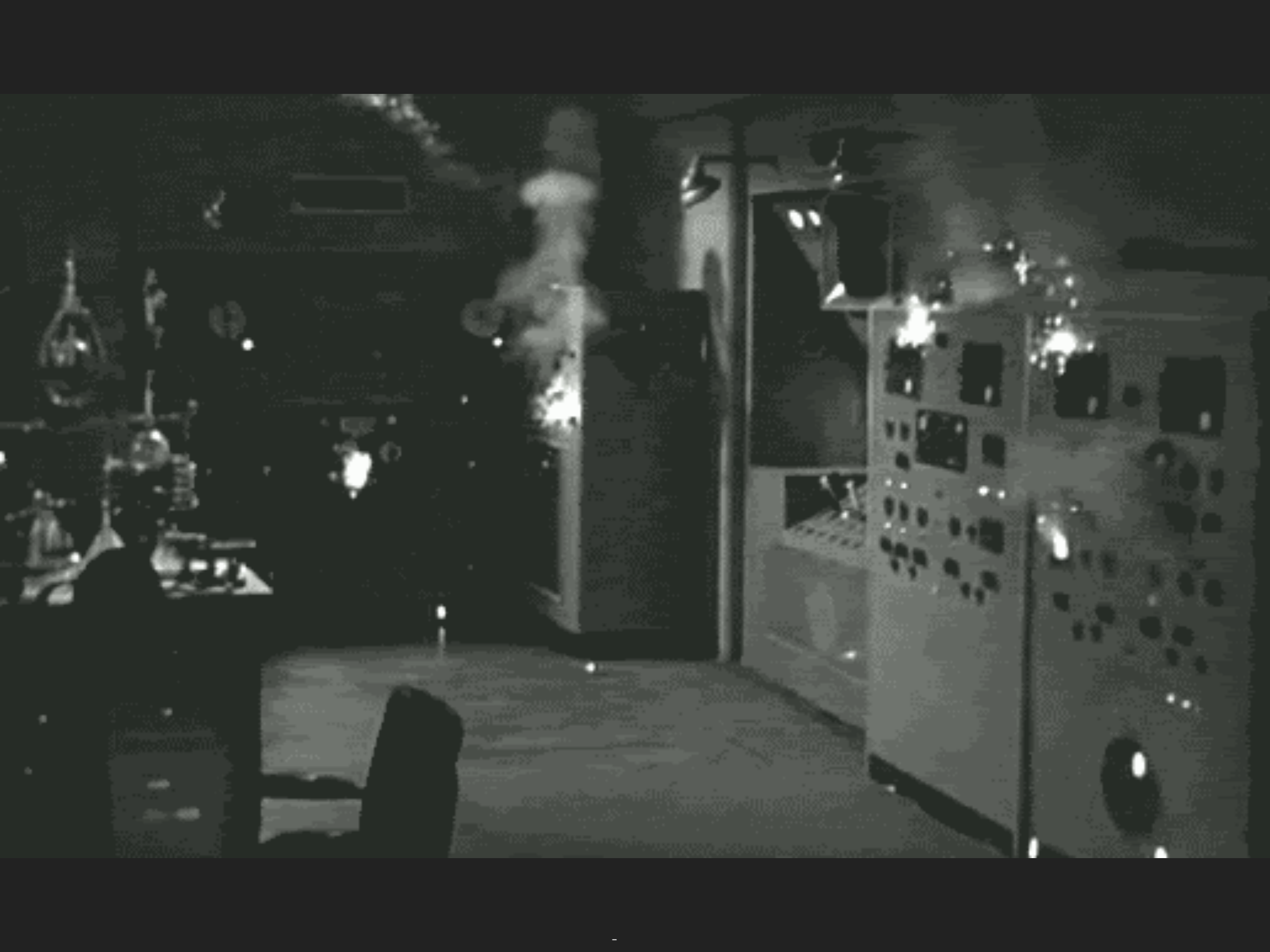
Optimized  
(progressive)  
rendering



Unoptimized  
rendering

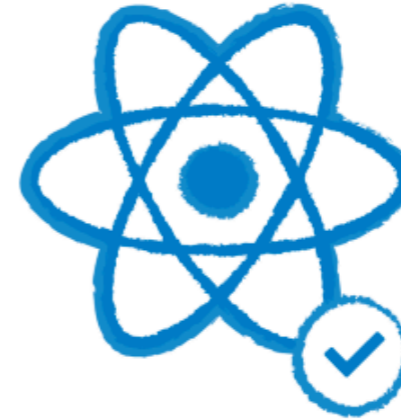
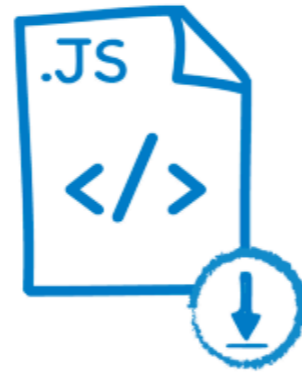
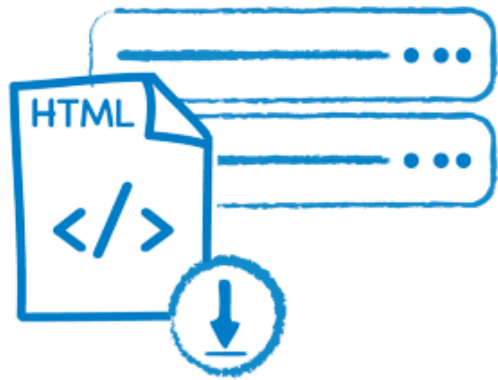


zzz





# SSR



Server Sending Ready to be rendered HTML Response to Browser

Browser Renders the page, **Now Viewable**, and Browser Downloads JS

Browser executes React

Page Now **Interactable**



# SERVER-SIDE RENDER

é bala de prata



# SERVER-SIDE RENDER

~~é bala de prata~~

ajuda a enfrentar o problema



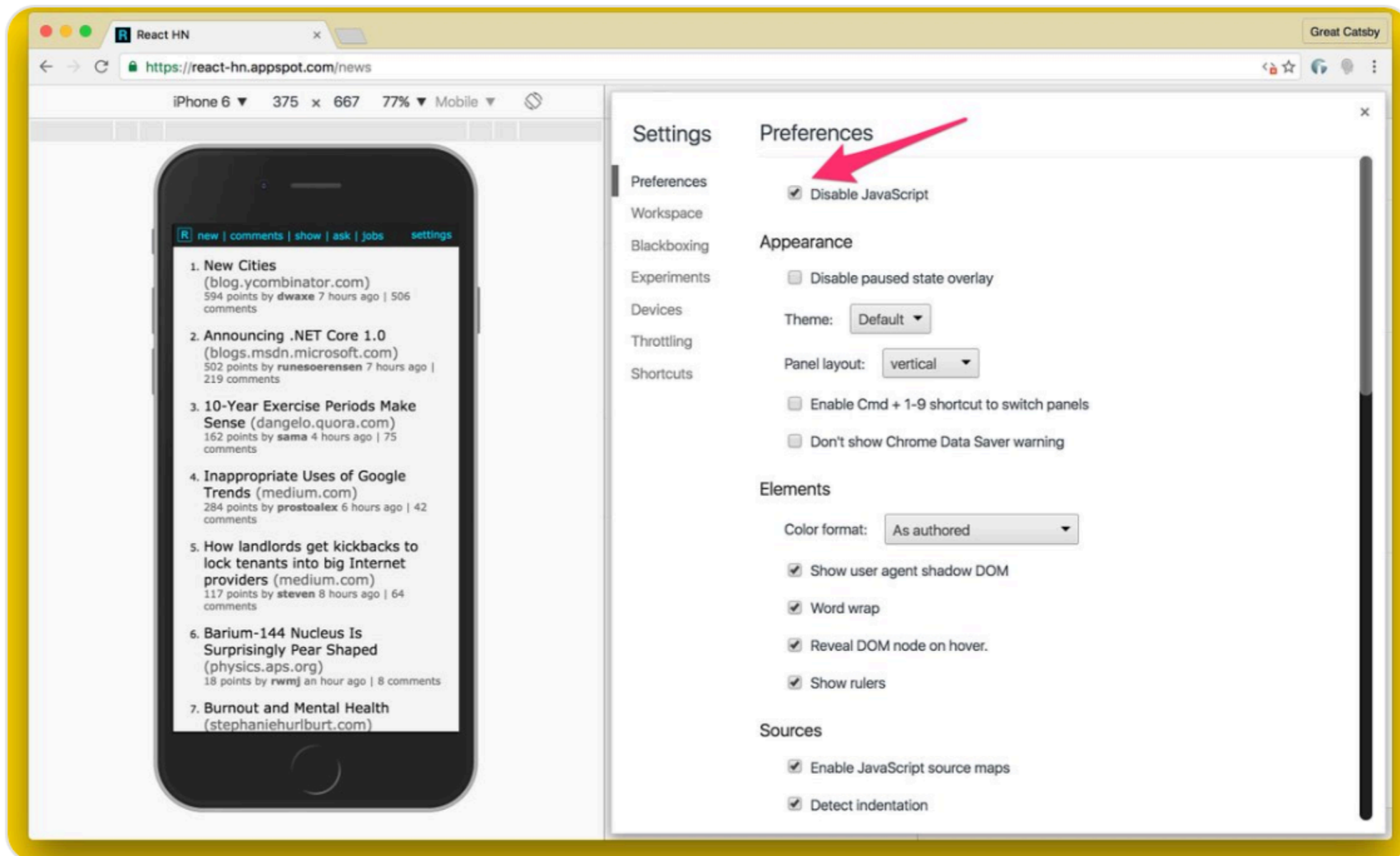
Addy Osmani

@addyosmani

Seguindo

^ Thanks to server-side rendering, top stories & comments will also work with JavaScript disabled.

Traduzir Tweet



19:41 - 27 de jun de 2016

[Tweet do Addy Osmani](#)

# SERVER-SIDE RENDER

~~é bala de prata~~

ajuda a enfrentar o problema

sem precisar fazer tudo server-side





# SERVER-SIDE RENDER

~~é bala de prata~~

ajuda a enfrentar o problema

sem precisar fazer tudo server-side

mas com um custo









**It's not about money...**

TALVEZ VOCÊ REALMENTE  
**NÃO PRECISE**  
DE SSR OU DE UM APP UNIVERSAL...

TALVEZ VOCÊ REALMENTE  
**NÃO PRECISE**  
DE SSR OU DE UM APP UNIVERSAL...



...MAS E SE PRECISAR?



$$I_0 \left( \sqrt{\frac{4K(1+K)}{P_{SF}}} P_K \right)$$

$$21 = (A + \eta)^2 + \kappa^2 \quad \text{and}$$

NÃO É COMPLEXO,

MAS ENVOLVE UM POUCO DE

~~M A G I A~~

TECNOLOGIA



# REACT

Já tem suporte a SSR direto da caixa





```
import express from 'express';
import path from 'path';

import React from 'react';
import { renderToString } from 'react-dom/server';
import Layout from './components/Layout';

const app = express();

app.use(express.static(path.resolve(__dirname, '../dist')));

app.get('*', (req, res) => {
  const jsx = (<Layout />);
  const reactDom = renderToString(jsx);
  ..
  res.writeHead(200, { 'Content-Type': 'text/html' });
  res.end(
    ..
    ..htmlTemplate(reactDom) // retorna uma string html com o app renderizado
  );
});

app.listen(2048);
```

```
import React from 'react';  
import ReactDOM from 'react-dom';  
  
import Layout from './components/Layout';  
  
const app = document.getElementById('app');  
ReactDOM.render(<Layout />, app);
```

```
import React from 'react';  
import ReactDOM from 'react-dom';  
  
import Layout from './components/Layout';  
  
const app = document.getElementById('app');  
ReactDOM.hydrate(<Layout />, app);
```

NEXT.js

Um framework para React SSR e/ou estático



```

import Layout from '../components/layout'
import fetch from 'isomorphic-unfetch'

const Post = (props) => (
  <Layout>
    <h1>{props.show.name}</h1>
    <p>{props.show.summary.replace(/<[/]?p>/g, ' ')}</p>
    <img src={props.show.image.medium}/>
  </Layout>
)

Post.getInitialProps = async function (context) {
  const { id } = context.query
  const res = await fetch(`https://api.tvmaze.com/shows/${id}`)
  const show = await res.json()

  return { show }
}

export default Post

```

```
const express = require('express')
const next = require('next')

const dev = process.env.NODE_ENV !== 'production'
const app = next({ dev })
const handle = app.getRequestHandler()

app.prepare()
  .then(() => {
    const server = express()

    server.get('/p/:id', (req, res) => {
      const actualPage = '/post'
      const queryParams = { id: req.params.id }
      app.render(req, res, actualPage, queryParams)
    })

    server.get('*', (req, res) => {
      return handle(req, res)
    })

    server.listen(3000, (err) => {
      if (err) throw err
      console.log(> Ready on http://localhost:3000)
    })
  })
  .catch((ex) => {
    console.error(ex.stack)
    process.exit(1)
  })
```

```
const express = require('express')
const next = require('next')

const dev = process.env.NODE_ENV !== 'production'
const app = next({ dev })
const handle = app.getRequestHandler()
```

```
app.prepare()  
  .then(() => {  
    .const server = express()  
  
    .server.get('/p/:id', (req, res) => {  
      .const actualPage = '/post'  
      .const queryParams = { id: req.params.id }  
      .app.render(req, res, actualPage, queryParams)  
    .})  
  
    .server.get('*', (req, res) => {  
      .return handle(req, res)  
    .})  
  
    .server.listen(3000, (err) => {  
      .if (err) throw err  
      .console.log('> Ready on http://localhost:3000')  
    .})  
  .})  
  .catch((ex) => {  
    .console.error(ex.stack)  
    .process.exit(1)  
  .})
```



# AFTER.JS



“Se Next.js e React Router tivesse um filho”

# ANGULAR UNIVERSAL

Angular no servidor! (ou onde mais você quiser)





# VUE && NUXT.JS

Sim, também dá pra fazer Vue SSR \o/



**YEAH, SO WHAT ELSE IS NEW?**



# GERAÇÃO DE SITES ESTÁTICOS



A close-up photograph of Tom Cruise sitting in the driver's seat of a car. He is wearing a blue button-down shirt and a grey seatbelt. He has a serious expression and is looking slightly to his left. The background shows a blurred view of a parking lot with several white vans and green trees.

**IS IT ALMOST OVER?**

LEMBRE-SE:

TECNOLOGIA

É PARA

HUMANOS





aputhin



aputhin\_