https://rpires71.github.io/milestone1/index.html

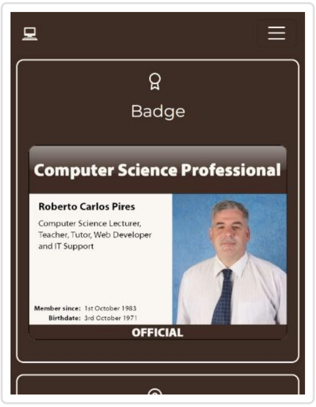
11/12 Performance

8/8 Best Practices

11/12

Performance

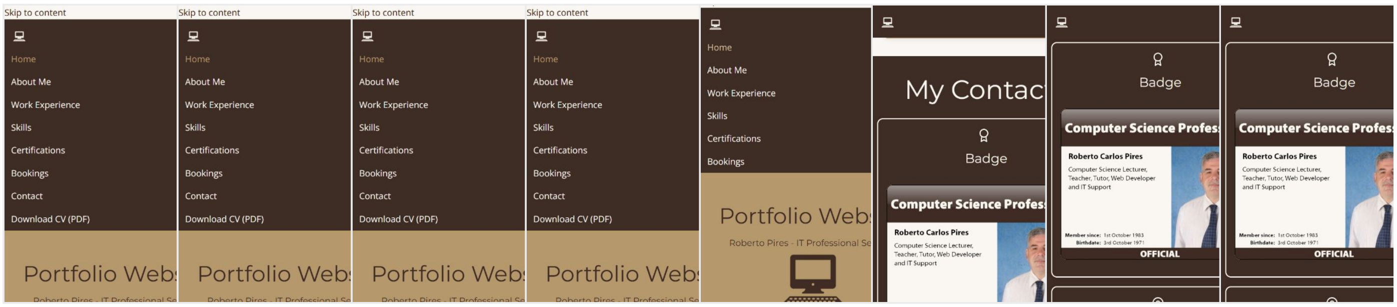
▲ 0–49   50–89   90–100


Badge
Computer Science Professional
Roberto Carlos Pires
Computer Science Lecturer, Teacher, Tutor, Web Developer and IT Support
Member since: 1st October 1983
Birthdate: 3rd October 1971
OFFICIAL

METRICS   Expand view

Total Blocking Time 0 ms	Cumulative Layout Shift 0.061
Interaction to Next Paint 40 ms	

View Treemap   View Trace



Later this year, insights will replace performance audits. Learn more and provide feedback here.

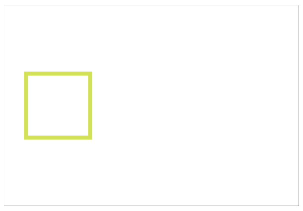
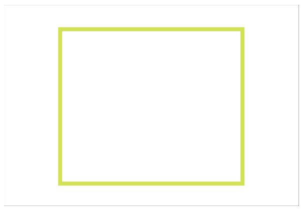
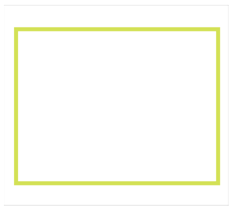
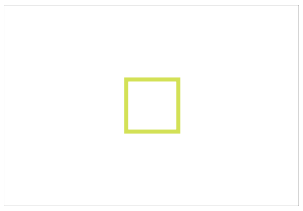
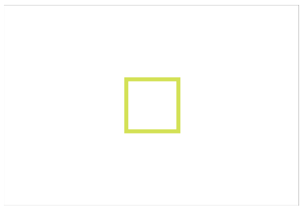
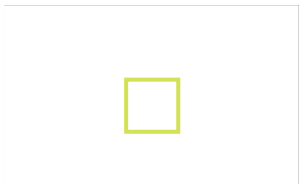
Try insights

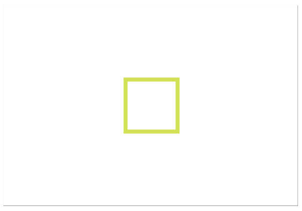
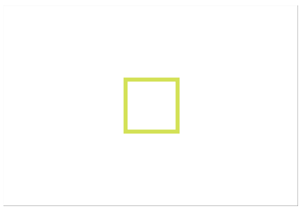
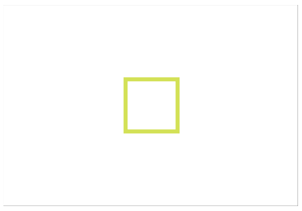
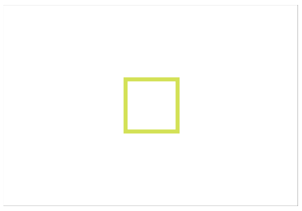
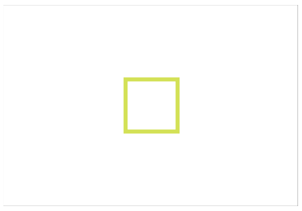
Show audits relevant to: All CLS INP

DIAGNOSTICS

Image elements do not have explicit `width` and `height`

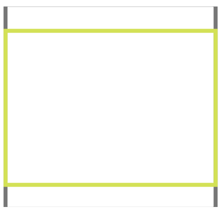
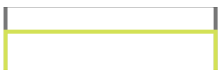
Set an explicit width and height on image elements to reduce layout shifts and improve CLS. Learn how to set image dimensions CLS

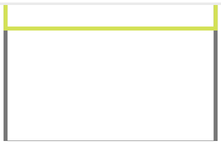
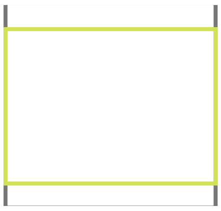
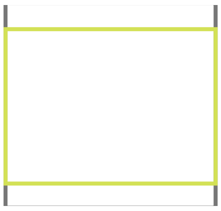
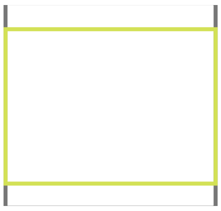
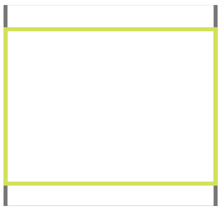
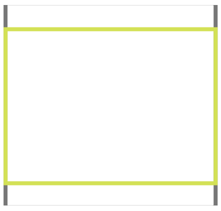
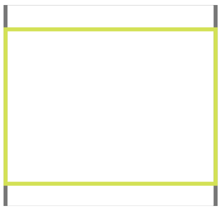
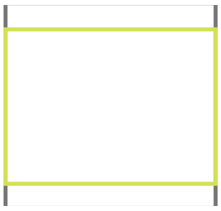
		URL
	img.logo.logo-navbar	...images/logodbg2.webp (rpires71.github.io)
	img.logo.logo-header.img-fluid	...images/logolbg3.webp (rpires71.github.io)
	img.img-fluid	...images/badgev2.webp (rpires71.github.io)
	img.footer-icon.me-2	...icons/award.svg (rpires71.github.io)
	img.footer-icon.me-2	...icons/map-pin.svg (rpires71.github.io)
	img.footer-icon.me-2	...icons/send.svg (rpires71.github.io)

		URL
	img.footer-icon.me-2	…icons/phone.svg (rpires71.github.io)
	img.footer-icon.me-2	…icons/message-circle.svg (rpires71.github.io)
	img.social-icon.me-2	…icons/linkedin.svg (rpires71.github.io)
	img.social-icon.me-2	…icons/facebook.svg (rpires71.github.io)
	img.social-icon	…icons/instagram.svg (rpires71.github.io)

Avoid large layout shifts — 9 layout shifts found

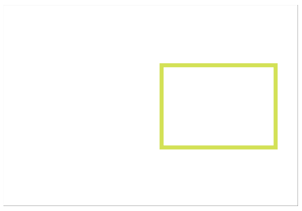
These are the largest layout shifts observed on the page. Each table item represents a single layout shift, and shows the element that shifted the most. Below each item are possible root causes that led to the layout shift. Some of these layout shifts may not be included in the CLS metric value due to windowing. Learn how to improve CLS CLS

Element		Layout shift score
	header.header-background.py-4	0.014
	header.header-background.py-4	0.011

Element		Layout shift score
		
	header.header-background.py-4	0.009
	header.header-background.py-4	0.007
	header.header-background.py-4	0.006
	header.header-background.py-4	0.005
	header.header-background.py-4	0.004
	header.header-background.py-4	0.003
	header.header-background.py-4	0.002

○ Avoid non-composited animations — 4 animated elements found

Animations which are not composited can be janky and increase CLS. Learn how to avoid non-composited animations CLS

Element	Name
a#nav-contact.nav-link.active	
Unsupported CSS Property: color	color
 button.navbar-toggler.collapsed	
Unsupported CSS Property: box-shadow	box-shadow
div#navbarNav.navbar-collapse.collapse	
Unsupported CSS Property: height	height
a.nav-link	
Unsupported CSS Property: color	color

○ Minimizes work during key interaction — 40 ms spent on event 'mousedown'

This is the thread-blocking work occurring during the Interaction to Next Paint measurement. Learn more about the Interaction to Next Paint metric. INP

Event target
a#nav-contact.nav-link.active

Phase	Total time	Script evaluation	Style & Layout	Rendering
Input delay	14 ms			
/milestone1/work-experience.html (rpires71.github.io)	9 ms	0 ms	2 ms	1 ms
Unattributable	4 ms	0 ms	0 ms	0 ms
Processing duration	0 ms			

Phase	Total time	Script evaluation	Style & Layout	Rendering
Presentation delay	25 ms			
/milestone1/work-experience.html (rpires71.github.io)	10 ms	0 ms	2 ms	2 ms
/milestone1/index.html (rpires71.github.io)	5 ms	3 ms	2 ms	0 ms

More information about the performance of your application. These numbers don't directly affect the Performance score.

PASSED AUDITS (21) Hide

○ Properly size images

Serve images that are appropriately-sized to save cellular data and improve load time. Learn how to size images.

○ Minify CSS

Minifying CSS files can reduce network payload sizes. Learn how to minify CSS.

○ Minify JavaScript

Minifying JavaScript files can reduce payload sizes and script parse time. Learn how to minify JavaScript.

○ Reduce unused JavaScript

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. Learn how to reduce unused JavaScript.

○ Efficiently encode images

Optimized images load faster and consume less cellular data. Learn how to efficiently encode images.

○ Serve images in next-gen formats

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. Learn more about modern image formats.

○ Enable text compression

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. Learn more about text compression.

Use HTTP/2

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. Learn more about HTTP/2.

○ Use video formats for animated content

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. Learn more about efficient video formats

○ Remove duplicate modules in JavaScript bundles

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity.

○ Avoid serving legacy JavaScript to modern browsers

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. Consider modifying your JavaScript build process to not transpile Baseline features, unless you know you must support legacy browsers. Learn why most sites can deploy ES6+ code without transpiling

Avoids enormous network payloads — Total size was 0 KiB

Large network payloads cost users real money and are highly correlated with long load times. Learn how to reduce payload sizes.

Uses efficient cache policy on static assets — 0 resources found

A long cache lifetime can speed up repeat visits to your page. Learn more about efficient cache policies.

○ User Timing marks and measures

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. Learn more about User Timing marks.

JavaScript execution time — 0.0 s

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. Learn how to reduce Javascript execution time. TBT

URL	Total CPU Time	Script Evaluation	Script Parse
/milestone1/work-experience.html (rpires71.github.io)	425 ms	9 ms	0 ms

URL	Total CPU Time	Script Evaluation	Script Parse
`Unattributable`	188 ms	4 ms	0 ms

Minimizes main-thread work — 0.6 s

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. Learn how to minimize main-thread work TBT

Category	Time Spent
Other	274 ms
Rendering	259 ms
Style & Layout	79 ms
Script Evaluation	24 ms

○ Minimize third-party usage

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading. Learn how to minimize third-party impact. TBT

Uses passive listeners to improve scrolling performance

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. Learn more about adopting passive event listeners.

Avoids `document.write()`

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. Learn how to avoid document.write().

○ Avoid long main-thread tasks

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay. Learn how to avoid long main-thread tasks TBT

Page didn't prevent back/forward cache restoration

Many navigations are performed by going back to a previous page, or forwards again. The back/forward cache (bfcache) can speed up these return navigations. Learn more about the bfcache

8/8

Best Practices

PASSED AUDITS (8) Hide

Uses HTTPS

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding mixed content, where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. Learn more about HTTPS.

Avoids deprecated APIs

Deprecated APIs will eventually be removed from the browser. Learn more about deprecated APIs.

Avoids third-party cookies

Third-party cookies may be blocked in some contexts. Learn more about preparing for third-party cookie restrictions.

Displays images with correct aspect ratio

Image display dimensions should match natural aspect ratio. Learn more about image aspect ratio.

Serves images with appropriate resolution

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. Learn how to provide responsive images.

No browser errors logged to the console

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. Learn more about this errors in console diagnostic audit

No issues in the `Issues` panel in Chrome Devtools

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

Page has valid source maps

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. Learn more about source maps.

Captured at Aug 20, 2025, 12:36 AM GMT

Emulated Moto G Power with Lighthouse 12.6.1

Single page session

User interactions timespan

Slow 4G throttling

Using Chromium 139.0.0.0 with devtools

Generated by **Lighthouse** 12.6.1 | File an issue