

WUNDERLANDMEDIA

10 Things You Can Delete From Your Website Today

The cheapest performance work there is. No refactor, no migration, no rebuild. Measure it, delete it, done before lunch.

By Kemal | wunderlandmedia.com

10

Deletions

280 KB

Unused JS on the Median Page

1

Afternoon

Measure First: Prove the Problem Before You Delete It

697 KB

JavaScript shipped by the median desktop page in July 2025, on a total page weight of 2.86 MB. Of that JavaScript, 280 KB was downloaded and parsed but never ran. *Source: HTTP Archive Web Almanac 2025, Page Weight.*

None of the ten items below is a rule. Each one is a hypothesis you can confirm in about ninety seconds with tools that are already on your machine. Run these four checks first, write down the numbers, and the delete list writes itself.

CHECK 1 — CHROME DEVTOOLS COVERAGE (WHAT LOADED BUT NEVER RAN)

☐ **Open Coverage**

In Chrome DevTools press Cmd+Shift+P (Ctrl+Shift+P on Windows), type "Coverage", and reload the page with recording on.

☐ **Read the red bars, not the score**

You get a per-file breakdown of how much CSS and JS was downloaded versus how much actually executed. Red is dead weight.

☐ **Sort by unused bytes and note the top five files**

Coverage will not tell you what is safe to remove. It tells you where to look, which is the part people skip.

☐ **Compare against the 280 KB median**

Page-builder sites sit well above it, because they load every module's CSS and JS on every page in case you used that module.

CHECK 2 — THE NETWORK PANEL (WHAT EACH THING ACTUALLY WEIGHS)

☐ **Filter by Font and count the files**

Then count the distinct font-weight values in your CSS. The gap between those two numbers is your font delete list.

☐ **Filter by Media and look at the hero video**

There is no vendor study for this one. Your own number is the argument, and it is usually bigger than you expect.

☐ **Sort by size, third-party domains first**

Anything served from a domain that is not yours is a candidate. Note the transfer size next to each name.

☐ **Disable cache and throttle to Slow 4G**

Test what a first-time mobile visitor gets, not what your warm desktop cache gets.

CHECK 3 — LIGHTHOUSE / PAGESPEED INSIGHTS (THE DELETE LIST WITH THE ARGUING DONE)

☐ **Run it before and after every deletion**

Not for the score. For the third-party table, which lists every external domain with transfer size and main thread blocking time.

☐ **Open "Reduce the impact of third-party code"**

Lighthouse names the offenders and estimates the savings, so you are not guessing at which vendor is expensive.

☐ **Open "Lazy load third-party resources with facades"**

This is item 10 on your list, pre-diagnosed. Every embed it flags is a facade candidate.

CHECK 4 — CONFIRM THE ITEM APPLIES TO YOU

☐ **Name an owner for every third-party script**

Write next to each one the name of the person who looked at its data this quarter. Any script without a name gets deleted.

☐ **Check usage data in the vendor dashboard, not your memory**

Conversations in the last 90 days, campaign spend in the last 12 months, recordings actually watched. Numbers, not vibes.

☐ **Grep the codebase before you dequeue anything**

Especially on WordPress, where a plugin can depend on a library your theme does not, and break silently when it disappears.

☐ **Pause before you delete, wait a week, then delete properly**

Reversible is the whole point. If nothing broke in seven days, nothing was using it.

Why this page exists

Some of the ten items will not apply to you. The point is that four or five of them probably will, and measuring first is what turns this from a list of opinions into a list of deletions you can defend to whoever asks.

1 The Homepage Carousel

1%

of visitors clicked the carousel at all, and about 84% of those clicked slide one. Slides two to five split a rounding error. *Source: Erik Runyon, University of Notre Dame homepage click data, 2013.*

WHAT IT COSTS

- ☐ **60 to 150 KB of script and CSS on a typical WordPress build**
A slider ships a JavaScript library and usually its own stylesheet, before a single image loads.
- ☐ **Several hundred KB of hero-sized images nobody sees**
Four or five full-width images load whether or not anyone waits for them to rotate past.

HOW TO CONFIRM YOU HAVE THE PROBLEM

- ☐ **Look for click events on slides two and up**
If the carousel has no event tracking at all, that is the answer: nobody has ever looked, so nobody has ever cared.

WHAT TO REPLACE IT WITH

- ☐ **Keep slide one as a static hero**
One headline, one button. That is what people saw anyway.
- ☐ **Move slides two to five into real sections further down**
They usually contain things that deserve to be read instead of glimpsed.

2 The Live Chat Widget Nobody Answers

500-750 KB

of JavaScript loaded by the heaviest chat widgets (DebugBear). A separate test of ten live chat tools clocked script parse and execution at 129 ms to 820 ms of main thread time (CueForGood).

HOW TO CONFIRM YOU HAVE THE PROBLEM

- ☐ **Count conversations in the vendor dashboard for the last 90 days**
Most small business widgets were installed during a growth push and abandoned three weeks later.
- ☐ **Check your median first response time**
"A handful" and "eleven hours" means you are paying half a megabyte for a worse experience than a contact form.

WHAT TO REPLACE IT WITH

- ☐ **A phone number and email in the header, or a plain form**
Zero KB, answered by the same person who was not answering the chat.
- ☐ **If you genuinely staff chat, load it on interaction only**
Put it on the pages where conversations actually start, and init on click. Most vendors document a manual init for exactly this.

BEFORE

```
<script src="//widget.chat.com/w.js">
</script> on every page, on load
```

AFTER

```
btn.addEventListener('click', loadChat) on /contact and /pricing only
```

3 The Third Analytics Tool Nobody Reads

90%+

of pages include at least one third party, and the median third-party inclusion chain is three deep, meaning the scripts you added are loading scripts you never chose. *Source: Web Almanac 2025, Third Parties.*

WHAT IT COSTS

- ☐ **A separate connection, script and main thread cost per tool**
GA4 through Tag Manager, plus a heatmap tool, plus a session recorder, plus whatever the last agency added.
- ☐ **Session recorders never stop working**
They listen to mouse movement, scrolls, clicks and DOM mutations on every page for every visitor, forever, so you can watch nine recordings once.

HOW TO CONFIRM YOU HAVE THE PROBLEM

- ☐ **List every third-party script and name its owner**
Write next to each one who looked at its data this quarter. Any script without a name gets deleted. Brutal in a good way.

This is a security list too

Every third-party script is code you did not write, executing on your domain with full access to your pages. Fewer scripts is fewer things that can turn hostile without you noticing.

WHAT TO REPLACE IT WITH

- ☐ **One analytics tool**
Pick the one someone actually opens.
- ☐ **Heatmaps as a two-week study, then removed**
Heatmap data is a study, not a permanent installation.

4 Tracking Pixels for Campaigns That Ended

50-200 KB

of script per pixel, plus its own network requests. Meta, LinkedIn Insight, TikTok, Google Ads remarketing, and one from an affiliate program that closed in 2023.

HOW TO CONFIRM YOU HAVE THE PROBLEM

- ☐ **Sort Tag Manager by last modified, look at anything over a year old**
The tags live in a web interface someone else set up, so grepping your theme files finds nothing.
- ☐ **Cross-check spend in the ad platform itself**
A pixel receiving events with no campaign spend in twelve months is collecting data for nobody.

WHAT TO REPLACE IT WITH

- ☐ **Pause the tag, wait a week, then delete it**
If you are nervous, that week is the whole safety net you need.
- ☐ **Keep pixels only where you are actively spending**
When you restart a campaign you re-add the tag. It takes four minutes.

5 jQuery, Loaded for Two Selectors

87 KB

minified, about 30 KB over the wire compressed. On a very large number of sites it exists to support a mobile menu toggle and a smooth scroll. Both have been native for years.

HOW TO CONFIRM YOU HAVE THE PROBLEM

- ☐ **Search your theme and custom JS for \$(and jQuery (**
Under thirty hits, this is an afternoon. Three hundred hits, leave it alone. This list is about easy wins.
- ☐ **Check the plugin list before you dequeue anything**
On WordPress, plugins may depend on jQuery even if your theme does not, and it will break them silently.

WHAT TO REPLACE IT WITH

JQUERY

```
$('.menu').toggleClass('open')
$.ajax({ url: '/api' })
$(document).ready(fn)
```

NATIVE

```
document.querySelector('.menu')
.classList.toggle('open')
fetch('/api')
<script defer src="app.js">
```

- ☐ **The same logic covers animation libraries**
Most sites load one to do a fade and a slide, both of which CSS does natively.

6 Four of Your Six Font Weights

139 KB

median font weight per desktop page in 2025. Usually two families at three weights each, or one family in Light, Regular, Medium, SemiBold, Bold and Black because the embed code offered them all. *Source: Web Almanac 2025.*

HOW TO CONFIRM YOU HAVE THE PROBLEM

- ☐ **DevTools Network tab, filter by Font, reload, count the files**
Each unused weight is a separate file, a separate request, and another chance for a flash of unstyled text.
- ☐ **Grep your stylesheet for font-weight and count distinct values**
Most sites use two. The gap between files loaded and values used is your delete list.

WHAT TO REPLACE IT WITH

- ☐ **Two weights, one family, WOFF2, self-hosted**
Set font-display: swap and preload the face used for your largest text.
- ☐ **Want a third weight for headings? Use a variable font**
One file, the whole range, no extra requests.

7 Half of Your Cookie Banner

1.43s → 3.61s

LCP in a DebugBear consent platform test, because the banner text became the largest contentful element. OneTrust alone ships upward of 184 KB of JavaScript; Cookiebot's script is around 34 KB and loads synchronously.

HOW TO CONFIRM YOU HAVE THE PROBLEM

- ☐ **Ask whether your consent platform matches your business model**
A brochure site with a contact form and GA4 is running the compliance machinery of an ad-tech company with dozens of vendors.
- ☐ **Compare the banner's weight to the tracking it consents to**
When the banner is heavier than the trackers, that is compliance theatre and worth questioning.
- ☐ **Check whether the banner is your LCP element**
Lighthouse names the LCP element. If it is the consent text, the banner is now your performance problem.

Not legal advice

I am not a lawyer. If you are running real ad targeting in the EU, you need the real thing. This item is about the sites that are not.

WHAT TO REPLACE IT WITH

- ☐ **Reduce what you collect first**
Drop the dead ad pixels from item 4 and your consent requirement shrinks with them.
- ☐ **Then a small self-hosted banner, or in some setups none at all**
The lightest consent implementation is having less to consent to.

8 The Icon Font

130 KB

for Font Awesome's full WOFF2, for a set you use nine icons from. Nine inline SVG icons are typically 100 to 300 bytes of markup each: under 3 KB total, less once compressed.

HOW TO CONFIRM YOU HAVE THE PROBLEM

- ☐ **Search your markup for fa- or icon- and count unique classes**
Under about thirty unique icons, inline SVG wins clearly.
- ☐ **Block the font request and reload**
Icon fonts fail as invisible icons or empty boxes. Inline SVG renders with the HTML, so there is nothing to fail.

WHAT TO REPLACE IT WITH

- ☐ **Copy the SVG paths you use into a sprite or straight into components**
No font load, no flash of missing icons, no request at all.
- ☐ **Attached to the icon font workflow? Subset it**
Subsetting to only the glyphs you use usually takes the file under 10 KB.

9 The Auto-Playing Background Video

No study to cite here, and that is the point

A hero video is normally the single heaviest asset on a site, frequently several megabytes, playing behind text that would read perfectly well over a still image. Open your own Network tab, filter by Media, and use that number as the argument.

HOW TO CONFIRM YOU HAVE THE PROBLEM

☐ **Network tab, filter by Media, read the transfer size**

It is usually larger than you expect, and it competes with everything else the page needs in the first second.

☐ **Load the page on a real phone**

Mobile browsers may refuse to autoplay it anyway, so half your visitors already get the poster frame. Which tells you the poster frame was sufficient.

WHAT TO REPLACE IT WITH

☐ **Use the poster frame as a static hero image**

Same composition, same message, a fraction of the bytes, and it is the LCP element you actually control.

☐ **If the video genuinely sells the thing, move it down behind a play button**

Which is also the honest signal that it is worth watching.

10 Every Embed That Loads Before Anyone Looks at It

1.2 MB → 30 KB

Zach Leatherman measured a single standard YouTube embed at almost 1.2 MB. A facade such as lite-youtube-embed does the same job at roughly 30 KB until someone clicks play: a 97% reduction, ten minutes of work, identical to look at.

HOW TO CONFIRM YOU HAVE THE PROBLEM

☐ **Run PageSpeed Insights and read two audits**

"Reduce the impact of third-party code" and "Lazy load third-party resources with facades". Lighthouse names the offenders and estimates the savings.

☐ **Inventory every embed, not just video**

Google Maps, Instagram feeds, Twitter timelines and review widgets all load third-party JS on every page view for a small fraction of visitors.

WHAT TO REPLACE IT WITH

BEFORE

```
<iframe src="youtube.com/embed/ID">
</iframe> ~1.2 MB on every page view
```

AFTER

```
<lite-youtube videoid="ID">
</lite-youtube> ~30 KB until click
```

☐ **For maps, a screenshot linking to Google Maps**

On mobile people want directions in their maps app, not a tiny pannable rectangle.

The Ten-Minute Audit: Run This Today

In order. Nothing here requires an install, a login, or a meeting. Work top to bottom and stop whenever you run out of afternoon: everything above where you stopped is already a smaller page.

MINUTES 0-3 — BUILD THE EVIDENCE

1 Run PageSpeed Insights on your homepage and save the result

This is your "before". Screenshot the third-party table specifically, not the score.

2 Open DevTools Coverage and reload with recording on

Note the top five files by unused bytes. Biggest red bar is almost always a file you can load conditionally instead of globally.

3 Network tab: filter Font, then Media, then sort by size

Three numbers written down: font file count, hero video weight, largest third-party transfer.

MINUTES 3-6 — THE FREE DELETIONS (NOTHING CAN BREAK)

4 Start with the third-party table. Delete what nobody can name an owner for

This is the fastest single win on the list and it needs no code change.

5 Open Tag Manager, sort by last modified, pause everything over a year old

Cross-check ad spend. No spend in twelve months means the pixel collects data for nobody.

6 Check chat widget conversations for the last 90 days

If it is not staffed, remove it now and put a phone number in the header instead.

7 Kill the session recorder and any second or third analytics tool

Keep the one someone actually opens. Heatmaps come back as a two-week study when you have a question.

MINUTES 6-9 — THE TEN-MINUTE CODE CHANGES

8 Delete the carousel, keep slide one as a static hero

Deleting a carousel has never once made a website worse.

9 Swap the YouTube embed for a facade

Ten minutes, looks identical, 97% lighter until someone presses play.

10 Replace the hero video with its poster frame

One attribute change on most builds. Move the video below the fold behind a play button if it earns its place.

11 Cut the font weights you never use down to two

Edit the font URL or the @font-face block, keep WOFF2, add font-display: swap.

MINUTE 9-10 — VERIFY, THEN QUEUE THE REST

12 Re-run PageSpeed Insights and compare the third-party table

Fewer rows, smaller transfer, less blocking time. That is the whole scoreboard.

13 Click through your key pages once on a phone

Menu, forms, video, maps. Five minutes of clicking catches anything the deletions touched.

14 Queue jQuery, the icon font and the cookie banner for next week

These three need a dependency check first, so they are not today's job. Everything above is.

15 Come back in seven days and delete what you only paused

Nothing broke, nothing was using it. Now remove it properly.

Do it once and it becomes a habit

The same way running a pre-publish checklist stops the same five mistakes shipping every time. The third-party table is your delete list with the arguing already done.

Quick Reference: All Ten at a Glance

Item	Typical cost	How to check	Replace with
1. Homepage carousel	60-150 KB JS/CSS + hero images; ~1% ever click (Runyon)	Analytics click events on slides 2+; no tracking = no interest	Static hero, slide one only
2. Unstaffed chat widget	500-750 KB JS (DebugBear); 129-820 ms parse/exec	Conversations in last 90 days; median first response time	Phone + email, or load on interaction
3. Third analytics tool	Extra connection, script, main thread; 90%+ of pages carry third parties (Almanac)	Name the person who read its data this quarter	One tool; heatmaps as a 2-week study
4. Dead tracking pixels	50-200 KB each plus own requests	Tag Manager by last modified; ad platform spend	Pause a week, then delete; re-add on campaign
5. jQuery for two selectors	87 KB min, ~30 KB compressed	Grep for \$(and jQuery(; check plugin deps	querySelector, fetch, defer
6. Surplus font weights	139 KB median font weight per page (Almanac)	Network tab filter Font vs. distinct font-weight values	2 weights, WOFF2, self-hosted, or a variable font
7. Oversized cookie banner	OneTrust 184 KB+; LCP 1.43s → 3.61s (DebugBear)	Is the banner your LCP element? Is it heavier than the tracking?	Collect less first, then a small self-hosted banner
8. Icon font	~130 KB WOFF2 for nine icons	Count unique fa- / icon- classes; under ~30 wins	Inline SVG (<3 KB), or subset under 10 KB
9. Autoplay hero video	Often several MB, the heaviest asset on the site	Network tab, filter Media; test autoplay on a real phone	Poster frame as static hero; video below fold
10. Eager embeds	~1.2 MB per YouTube embed (Leatherman)	Lighthouse: third-party impact + facade audits	Facade (~30 KB); screenshot linking to Maps

"Deletion is free, it is reversible, and it is the only performance work with no downside. Every item on this list makes the page smaller and leaves one less thing that can break, get exploited, or need updating. Most of them also make the page clearer, which is the part that shows up in conversions rather than in Lighthouse."

Kemal, Wunderlandmedia

Be honest about what this buys you

Removing a slider and two pixels will not turn a slow site fast. If your stack is fundamentally heavy, deletion buys a few hundred milliseconds, not a new site. Start with the third-party table, delete what nobody can name an owner for, then do the carousel. If the numbers still look bad afterwards, the problem is deeper than the widgets.

Read the Full Post

wunderlandmedia.com/things-you-can-delete-from-your-website-today

Helping people remove website bloat and then fix what is left is a fair chunk of what I do. I am happy to tell you when deleting is enough and when it is not. wunderlandmedia.com

Sources: HTTP Archive Web Almanac 2025 (Page Weight, Third Parties) · DebugBear (chat widget performance, cookie consent banner performance) · CueForGood (live chat load speed test) · Erik Runyon (carousel interaction stats, University of Notre Dame) · Zach Leatherman (YouTube embed weight).