adtrak

Page speed

Big Team

‹a›

What we’re going to cover...

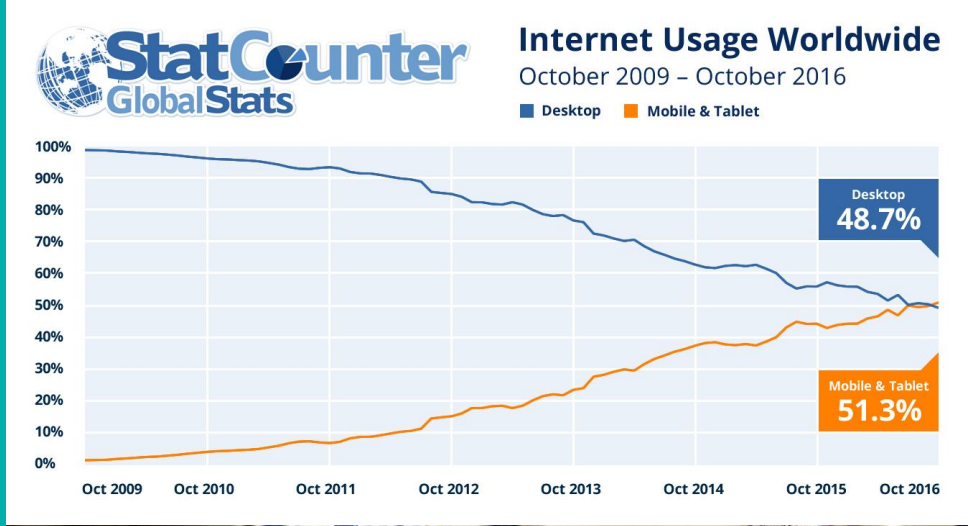
- Why does it matter?
- Tools and metrics
- Worst Offenders (and how to deal with them)
- The Future

Why does it matter?

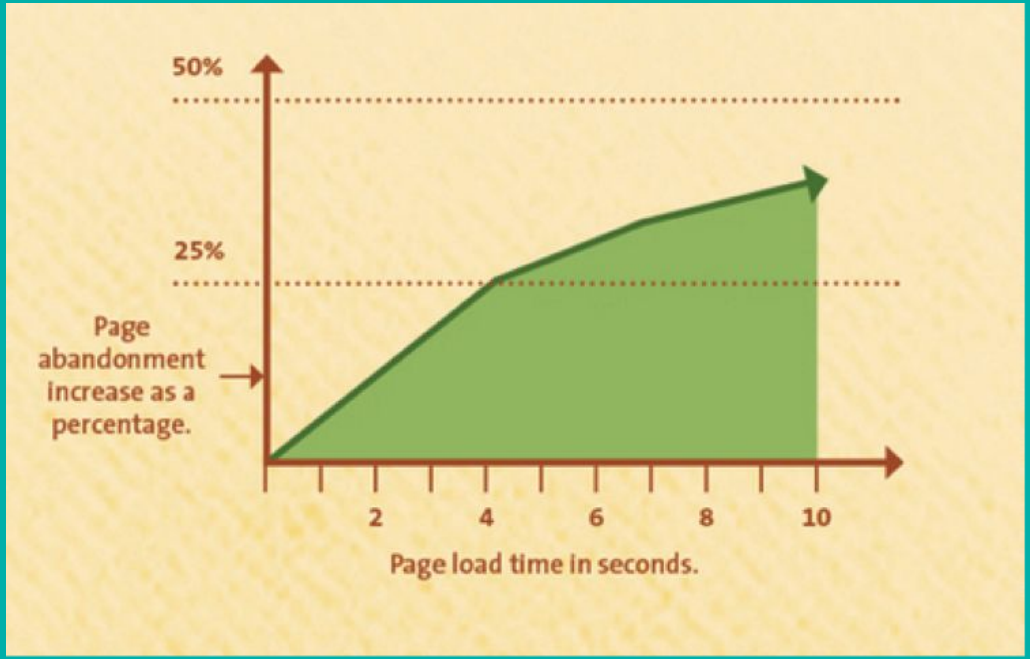
Although Internet speeds are increasing, **the way people access the web is changing**, with mobile traffic increasing day by day

This means we have to cater for users on slow, public connections

...as well as being considerate of data plans



Every Second Counts.



BBC has seen that they lose an additional 10% of users for every additional second their site takes to load

AliExpress reduced load times by 36% and saw a 10.5% increase in orders and a 27% increase in conversions for new customers

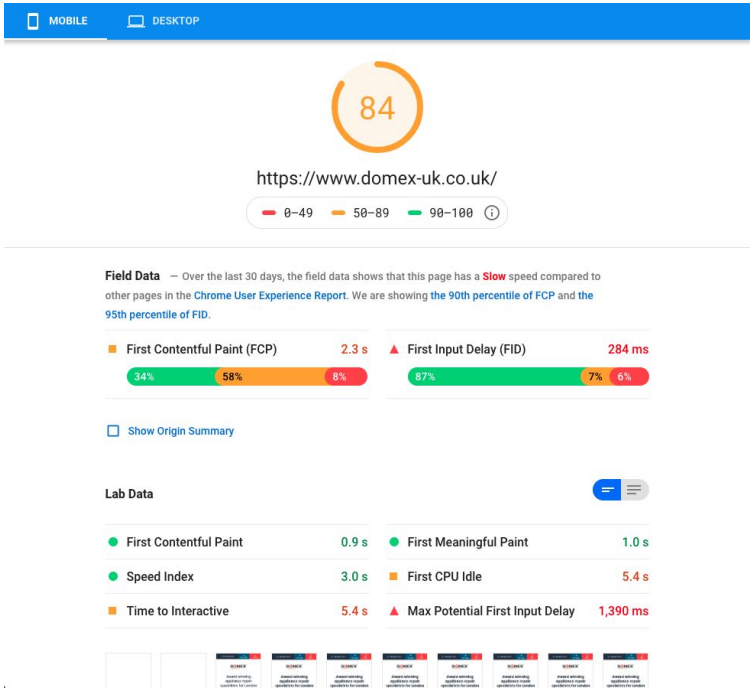
The Trainline reduced latency by 0.3 seconds across their funnel and customers spent an extra £8m a year

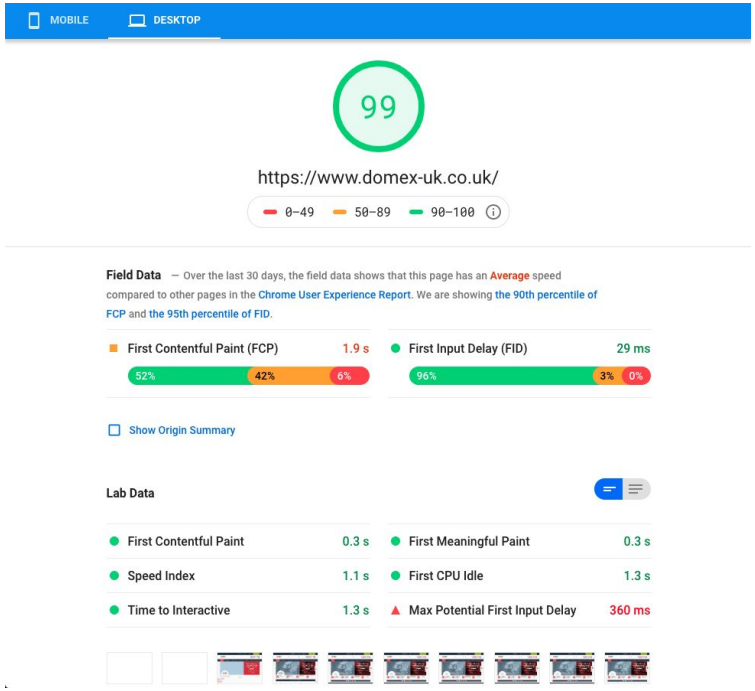
53% of visits made to mobile sites are abandoned after 3 seconds according to Google

Tools & Metrics

PageSpeed Insights

The main tool we use is Google's PageSpeed Insights (powered by Lighthouse)





PageSpeed Insights

- Overall scores will vary with each test due to fluctuating server response times
- Scores can also vary from one server to another (it's not all about how the site has been built on the front-end)

- Also important to note that you are testing a single URL
  - http://whatever.com will score worse than https://www.whatever.com because it has to perform two redirects (to the www version and to the https version)

First Meaningful Paint

- Time it takes for the “**above-the-fold**” portion of the page to render
- We need to try and **eliminate any http requests** that are required to render this section
- We do this by taking the CSS required to render this section of the page (the “**Critical CSS**”) out of the main external stylesheet, and putting it straight into the head of the page (and then loading the main stylesheet after this)

Time To Interactive

- When the whole of your page is fully able to handle user interaction
- **Limit the amount of JavaScript** used on the page
- This includes **3rd party scripts** used by things like Wistia and Google Map embeds
- If you can’t remove scripts, we can often **lazy load** them

Worst Offenders

3rd party scripts & iframe embeds

Embedding an iframe or widget on your page is quite literally like including another mini website on your page, which means your page speed is then subject to all of the resources used by that 3rd party embed

- Reviews.co.uk widgets
- Wistia
- Google Maps
- YouTube
- Facebook pixel
- Google Analytics
- Mouseflow

For a simple iframe embed (such a Google Map), we can use a lazy loading script to only load this iframe when it is in or almost in view

```
<iframe src="https://..."></iframe>
```

```
<iframe class=”lazyload” data-src="https://..."></iframe>
```

Also **WordPress plugins** - general rule is don’t use any plugins that add stylesheets/scripts to the front-end

Images

- **File size** - images should be optimised as much as reasonably possible
- **Physical dimensions** - if the image size on the page is 300x200, make sure the original image isn't something silly like 1600x1000 (Copy/IM need to be mindful of this when adding images in WP)
- **Format** - There are new image formats (mainly .webp developed by Google) that allow for much better compression; webp is quite well supported but not universally so, so we'd still need to provide jpeg/png fallbacks

```
<img src="https://..." alt="Alt Text">
```

```
<img class="lazyload" data-src="https://..." alt="Alt Text">
```

Web fonts

- Often quite heavy file sizes, so best to limit the amount used
- Need to remember that if you're including 1 font with 5 different weights/styles (such as light/bold/italic etc.), then you're including 5 webfonts, not 1!
- Should use SVG icons instead of an icon font
- We can use `font-display: swap;` on the web fonts to improve the font loading experience
    - This ensures your text is rendered in a fallback font whilst the web fonts are loading
    - PSI will kick off if we don't do this
    - Adobe Fonts (Typekit) doesn't provide any way of doing this :(

Large amounts of JS

- Should limit the number of scripts included in a site (each one is an extra http request)
- Should also limit the size of these scripts - a really large script can have a disastrous effect on the *Time To Interactive* result, which can bring the PSI score down massively on mobile
- It's generally best to combine your site's javascript into a single file
  - However, if you have large chunks of JS that are only required on specific pages, it's best to put them in their own file and then include them only on those pages

The Future

The Future

- The future of the web is **static** - this means cutting out the server processing time by delivering static html pages
- Instead of running the PHP code on the server with each page load, we take a snapshot of the page and cache it on the server, and then deliver this to the browser as a static html page
- We can do this with WordPress sites with plugins, but it’s a bit cumbersome and can break certain features on some sites
- The future of static sites on the web is through PWAs and Server Side Rendering

The End.