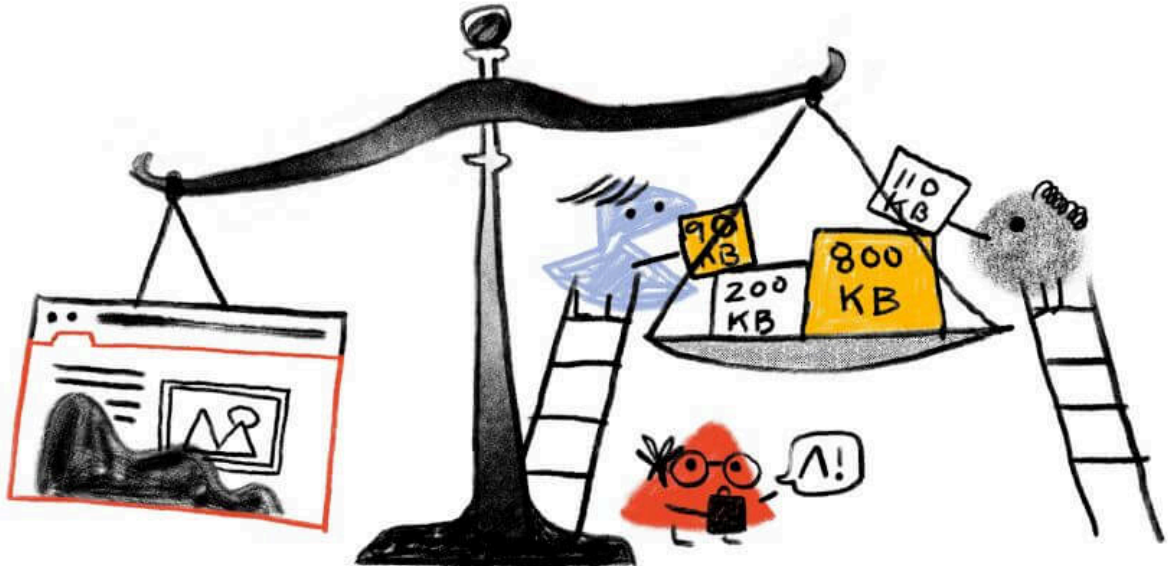


# Page Weight

Web Almanac 2020



## Content team

**Authors:** Henri Helvetica (henrihelvetica@gmail.com), ...

**Reviewers:** Tammy Everts (tammy@speedcurve.com), Boris Schapira (boris@schapira.dev), Katie Hempenius (khempenius@google.com)

**Analysts:** Paul Calvano (pacalvan@akamai.com)

Hello content team! This is your personal doc to collaborate on and plan the contents of your chapter. Click **Request edit access** above to get started.

The objective of your chapter is to write a data-driven answer to this big question:

**“What is the state of Page Weight in 2020?”**

Learn more about the [chapter lifecycle](#) and refer to your chapter’s [tracking issue](#) on GitHub for more info. Thank you all for your contributions!

# Outline

*The purpose of this section is to define the scope of the chapter by listing all of the topics to be explored. Think of this as your table of contents and list of talking points.*

**Completed by July 20**

- **Introduction**
  - #PageWeightStillMatters
  - Bandwidth: Now At A Possible Premium
  - Still Tracking The Same Assets
  - Intricate + Interactive: *Page Complexity Remains*
- **Analysis**
  - Page Weight: *Classic Data*
    - Mobile
    - Desktop
  - Time Machine: *Page Weight Over Time*
    - Mobile
    - Desktop
  - YOY Examination: *A More Granular Look At 1 Year Since The Almanac*
  - Page Requests
    - Mobile
    - Desktop
  - File Formats
    - Mobile
    - Desktop
- **Coming Times Under COVID-19 Conditions**
- **A Not So Distant Future**
- **Conclusion**

# Metrics

*The purpose of this section is to track all of the metrics needed to substantiate the chapter. Analysts can use this section to triage whether the HTTP Archive dataset on BigQuery is able to measure each metric.*

***Prepared by*** July 27

***Queried by*** September 7

***Reviewed by*** September 14

-

# Draft

*The purpose of this section is to iterate on a full draft of the chapter after all of the metric results are ready. Data visualizations for inline figures can be copy/pasted in from the results spreadsheet.*

*First draft peer reviewed by November 12*

*Data visualizations prototyped by November 26*

*Final markdown submitted for editing by November 26*

# Page Weight

Web Almanac 2020

## — introduction —

Page weight is one of the simpler metrics available. Much like stepping on a human scale to get a sense of your personal weight (well, mass really, but you get it), loading a page will provide a sense of the amount of resources collected and requested. But as the web and webpages have matured and grown, so have associated metrics — such as page weight. It can affect a page's performance much like personal weight (mass) can do the same. This chapter will take a deeper dive and peel back the layers of web pages, and see what it is that constitutes a page's weight at the possible detriment of the end user: you, I, us.

## — #PageWeightStillMatters —

#PageWeightStillMatters would almost imply that it didn't or ever mattered. It might not have mattered when text based Craigslist launched. But 25 years ago when it was founded, Mosaic 1.0 also launched the same year, and Waterfalls by TLC was a top hit. The web matured as did resources. It was just a few years back when the twitterverse was tied up discussing how the average size of web pages now equaled the size of the [original doom](#). Many of us mused about what the size the page could become in time, [including our very own Tammy](#) Everts, but the reality is startling. A page sits @ ~4 MB and 3.7 MB, desktop/mobile respectively, at the 75th percentile, and a shocking 7.4 MB and 6.7 MB at the 90th percentile. There are multitudes of implications in having such heavy pages, like the likelihood of poor user experience due to

unreliable networks. Today, despite lessons [learned a decade ago](#), we are experiencing variations of the same challenges: despite having slightly better networks, we are working with much larger resources.

## — bandwidth —

In 2016, when asked to explain why I asked an Australian tourist who is delighted with UK Internet, Google's Ilya Grigorik [had two words](#): *physics damn'it!* (whoops, that's three). The point was simple: though you might benefit from increased bandwidth, the laws of physics still prevail. An Australian is unable to escape laws of latency. In the best case scenario, at home in Sydney, this Australian was experiencing enough latency that his Internet was at times perceived as unresponsive.

Now, imagine that the same Australian, knowing that at the 75th percentile, his page is making about 108 requests (more on that later), and we still have no idea of the network protocol, the resources being requested, the level of compression or optimization. You can pursue the H/2 and compression chapters for more information on the life of a modern request.

*physics damn'it!*

## — assets —

In 25 years of modern browsing, the assets and resources have mostly not changed, other than the amount. The HTTP archive modus operandi is “how the web was built”, and that was mostly done with HTML, CSS, JavaScript and finally images.

Prior to 1995, the web's page weight was mostly predictable and manageable. But with RFC 1866, which introduced HTML 2.0, responsible for the introduction of inline images via the IMG tag, page weight would make a dramatic increase, all for the good of web development (adding images was seen as a positive experiment).

For the most part, the rule of thumb has been that images would make up the majority of page weight. It was certainly the case and a concern when in-line images were added to the web then, and remains the case today. In a separate scenario, as image data will be the greatest source of page weight, it will also be the greatest source of page weight savings (more on that later). This will be achieved from ensuring that the images are sized properly, but also making sure that the images are at the optimization sweet spot - finding the best balance of quality and file size.

Although Javascript is on average the 2nd most abundant resource on a page, we tend to have more opportunities in working with that file type: from bundling, compression and minification to name a few.

## — intricate and interactive —

The web's journey from the plain, near pedagogical platform, to the innovative, intricate and highly interactive apps it has become, the rudimentary page weight metric hid a bigger story: a ratatouille of resources, each affecting modern metrics, in turn affecting user experience.

Whenever we talk about interactivity, we are talking almost exclusively about JavaScript. Now, though we are not here to discuss interactivity in any depth, we know there are metrics which are focused and dependent on JavaScript content and execution. So weightier the JavaScript, likely to have a greater impact on interactivity metrics (time to interactive, total blocking time). We have a javascript chapter that dives a pinch more.

## — analysis —

As we post and parse the statistical results, just a reminder that the data is based on transfer sizes. Added, we are employing decompressed sizes in this analysis when possible.

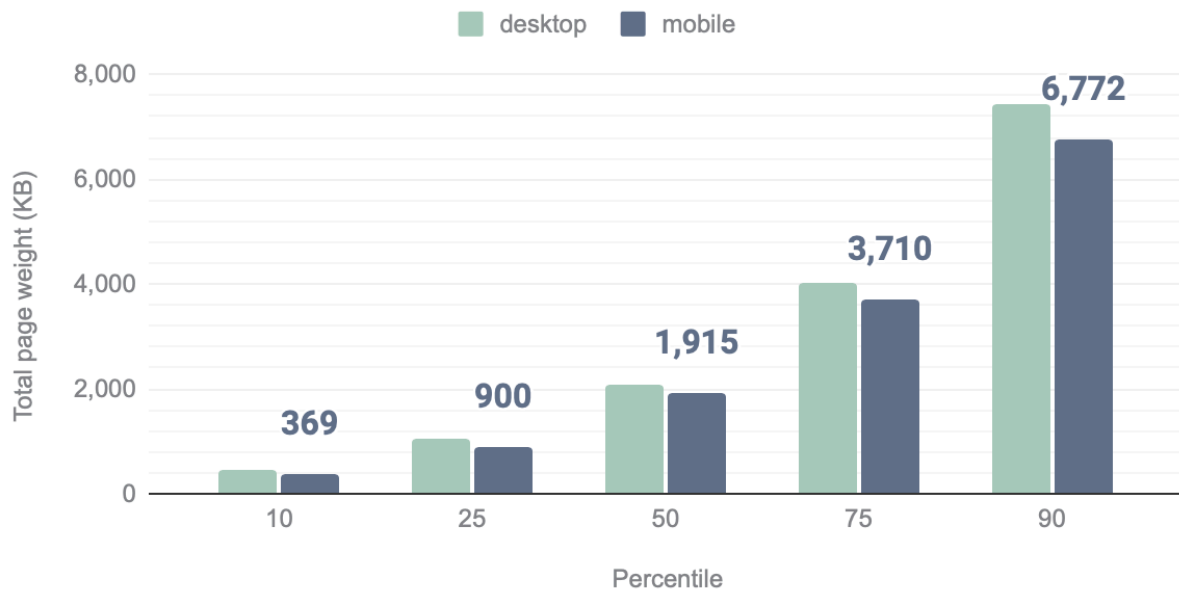
## — page weight —

let's look at the classic page weight, on both desktop and mobile. The deltas are mostly due to a few less resources transferred on mobile, a likely pinch of media management, but you can see below that at the median, the differences are not that significant between the two clients. We are highlighting the mobile results.

We can however surmise from this the following: we are closing in on 7 MB of page weight on mobile and 7.5 MB on desktop at the 90th percentile. The data is following an age old trend: growth in page weight is on the upward trajectory yet again, from the previous year. Popping the hood, we can see how things look at the median and average for each resource. One thing again remains: images are the dominant resource and JavaScript is the second most abundant, though a far second.

# Page weight distribution

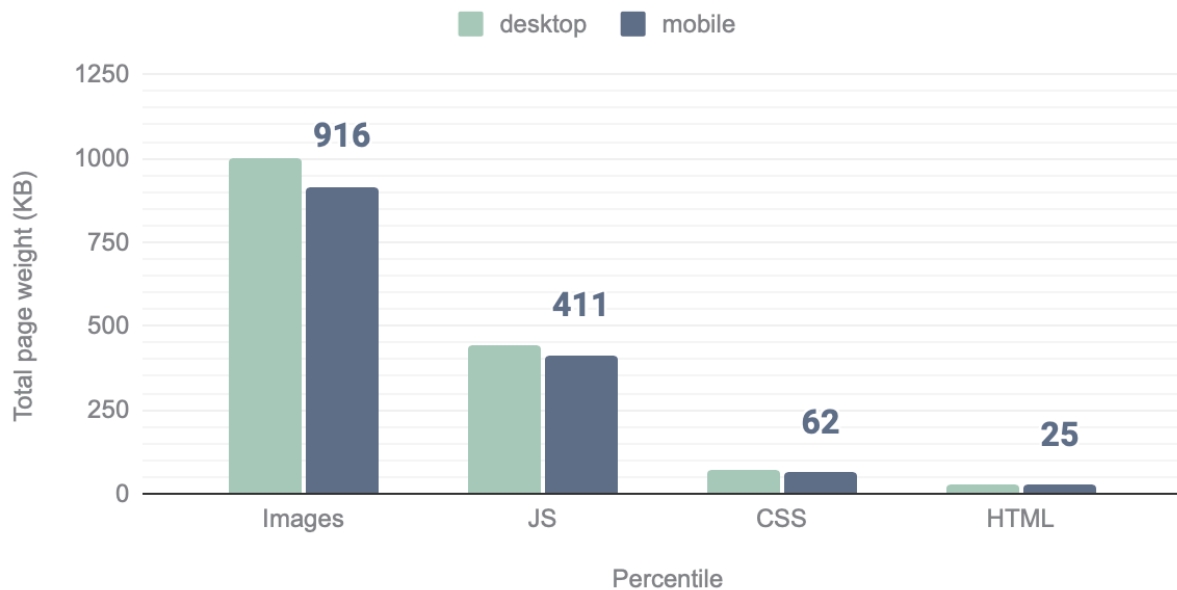
Web Almanac 2020: Page Weight



| Page Weight (KB) | client  |        |
|------------------|---------|--------|
| percentile       | desktop | mobile |
| 10               | 469     | 369    |
| 25               | 1,033   | 900    |
| 50               | 2,096   | 1,915  |
| 75               | 4,034   | 3,710  |
| 90               | 7,438   | 6,772  |

## Median page weight by content type

Web Almanac 2020: Page Weight



|        | client  |        |
|--------|---------|--------|
| Values | desktop | mobile |
| Images | 1001    | 916    |
| JS     | 444     | 411    |
| CSS    | 68      | 62     |
| HTML   | 26      | 25     |

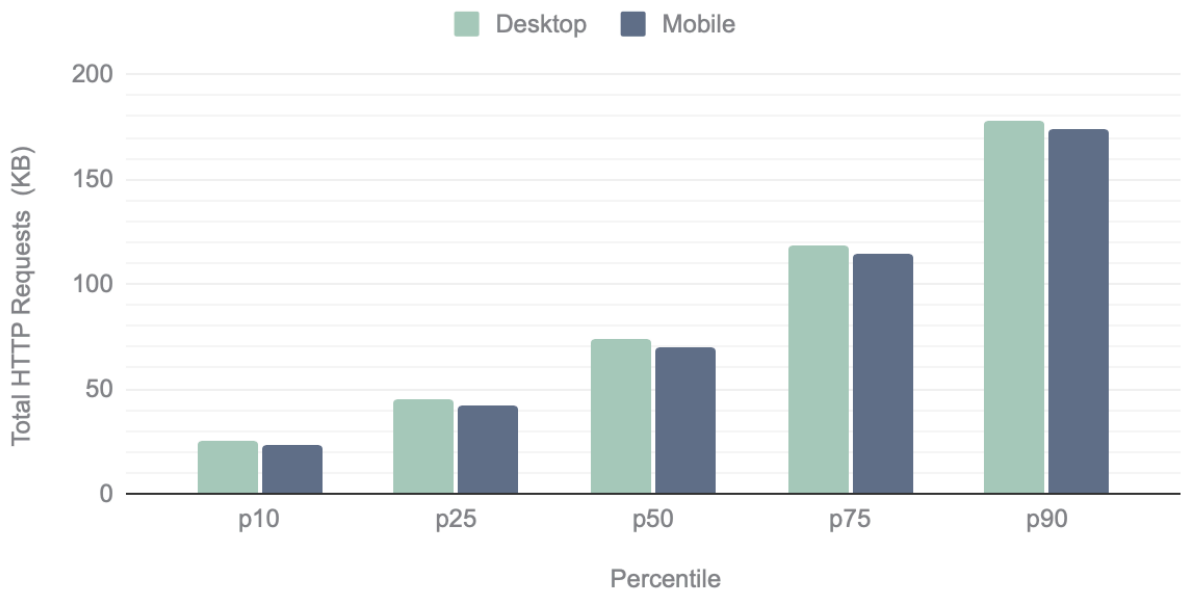
### — requests —

We have an old adage: the quickest request is the one never made. Dare we then say: the smallest resource is one never requested.

at the request level, much is the same. The weightiest resources are making the most requests. The request distribution will show that the difference between desktop and mobile is not so significant, with desktop leading the way. Something worth noting: the median request on desktop at this time is the same as last year later (74), yet the page weight has ticked up (+122kb). A simple observation, but one which confirms the trajectory we've seen over the years.

# Distribution of Requests

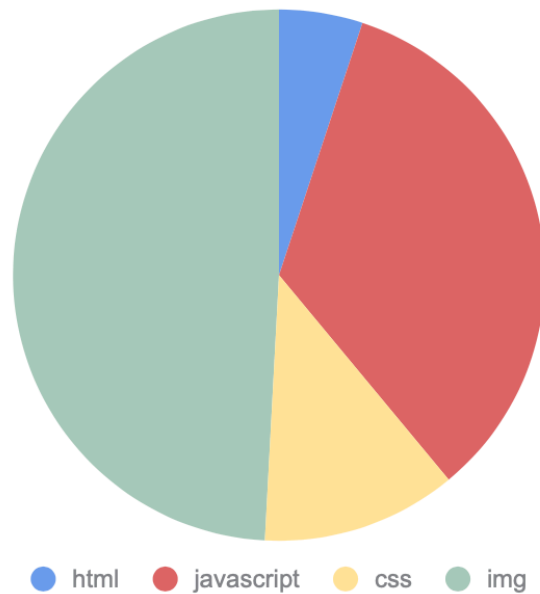
Web Almanac 2020: Page Weight



| percentile | Desktop | Mobile |
|------------|---------|--------|
| p10        | 25      | 23     |
| p25        | 45      | 42     |
| p50        | 74      | 70     |
| p75        | 118     | 114    |
| p90        | 178     | 174    |

## Median Requests by Resource Type

Web Almanac 2020: Page Weight

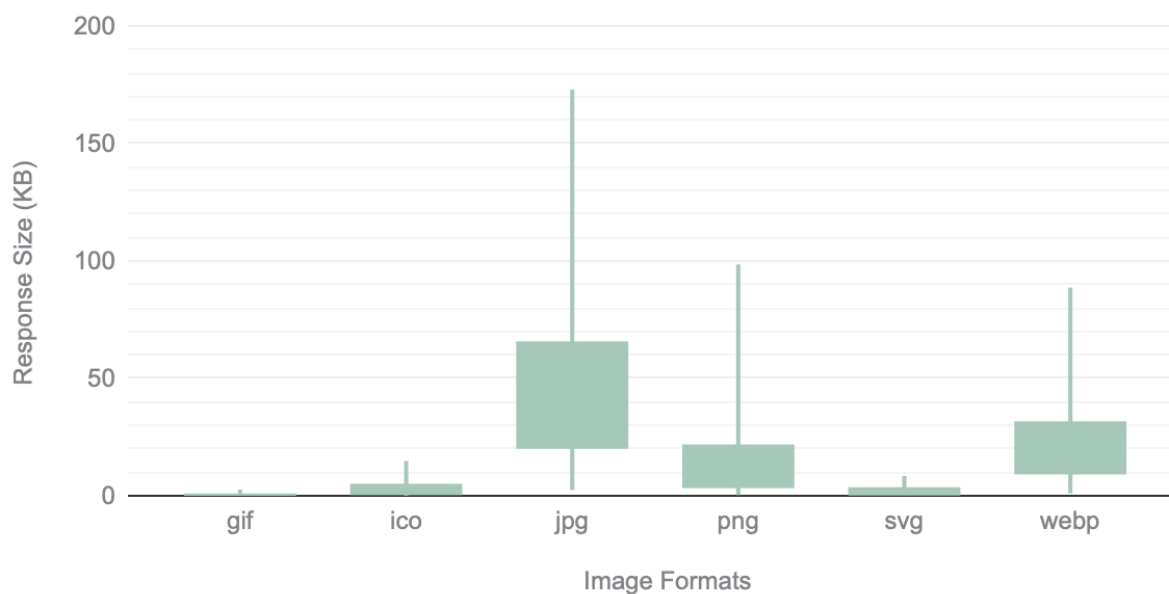


### — file formats —

We know that images are a great source of page weight. This graphic below shows us the top sources of image weight and the weight distribution. Top 3: JPG, PNG and webp. So not only is the JPG the most popular image format, it also tends to be the largest by size as well - even larger than a lossless format like the PNG. But as we noticed last year, that has to do w/ the predominant use case for the PNG - which seems to be icons and logos. Added,

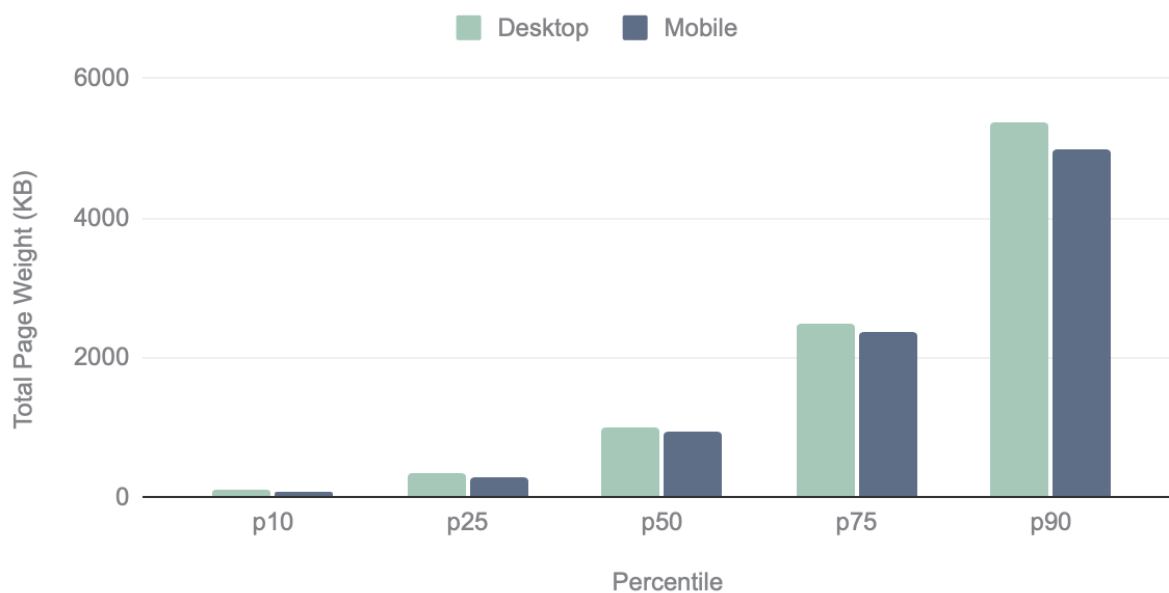
## Distribution of Image Sizes by Format

Web Almanac 2020: Page Weight



## Total Image Bytes per Webpage

Web Almanac 2020: Page Weight



| percentile | Desktop | Mobile |
|------------|---------|--------|
| p10        | 89      | 67     |
| p25        | 328     | 284    |
| p50        | 983     | 928    |
| p75        | 2,474   | 2,365  |
| p90        | 5,362   | 4,975  |

## — covid-19 —

2020 has been the most demanding of any year in Internet history. This is based on self reporting by [telco companies](#) all over the globe. YouTube, Netflix, gaming console manufacturers and many more were asked to [throttle their networks](#) due to anticipated bandwidth demands of Covid and the stay at home orders. There are now new suspects creating demands on the networks: we are now working from home, teleconferencing from home, and now schooling from home as well. In the midst of this crisis some government organizations have moved forward to optimize all aspects of the site and redesign and or updates. Two such examples of being [Ca.gov](#) ([link](#)) and [gov.uk](#). In these times, Covid has certified the Internet as an essential service, and being able to access crucial and life-saving information, Must be as friction free as possible, which includes a manageable page weight via discipline delivery of data.

If we have been married to the Internet, Covid has forced us to renew our vows. Assuring that content is delivered as efficiently as possible over the Internet, page weight must be kept at the forefront at all times.

## — a not so distant future —

We have watched for 25 years page weight grow steadily. It might have been one of the greatest stock investments — had it been one. But this is the web. And, we are trying to manage data, requests, file size and ultimately page weight. We have just combed over data, seeing how images are the greatest source of weight. This means, it will also be our greatest source of savings. 2020 was a pivotal year, a possible inflection point for HTTP Archive tracking of web data. 2020 marked the year the modern format webp was finally adopted by Safari, making this format finally supported by all browsers across the board. This means that the format could comfortably be used with little to no fall back. The most important point? The potential for significant page weight savings is unavoidable — at a possible 30%. Even more interesting is the idea of a more modern format: [avif](#). This format has burst onto the scene with enough support today for approximately 70% browser market share, creating a scenario for small image file sizes - even smaller than webp. And lastly, and possibly most distant: media queries level 5, [‘prefers-reduced-data’](#). Though in very early draft, this media feature will be used to detect if a user may have a preference for variant resources in data sensitive situations

Looking at the crystal ball, the third instalment of the Almanac and the page weight chapter could have a much different look in 2021. The big technological and engineering investments into images, might finally provide the diminishing returns we have been looking for.

## — Conclusion —

it's of no surprise that web pages have generally kept growing. We have been feeding more resources down the wire to create richer experiences, more engaging interactivity, more stunning visuals through more powerful imagery. We have created these applications at the cost of data overages and user experiences. But as we move forward and keep pushing the web to places we had never anticipated, we are also making additional advances in engineering, as mentioned earlier. We may begin to see a drop in page weight as early as next year, as modern raster image formats see more adoption, we start to manage javascript more efficiently, and deliver the data down the wire with the discipline that users demand.