Web performance and quality report

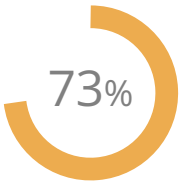
https://leite5.github.io/

This report is provided by Dareboost, **an online tool for web performance and quality analysis and monitoring**.

Don't hesitate to check out our offers or to contact us:
contact@dareboost.com

Table of contents

Summary



9 Issues

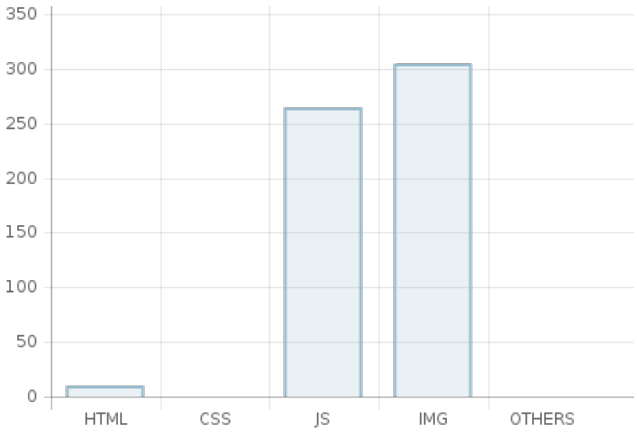
7 Improvements

61 Successes

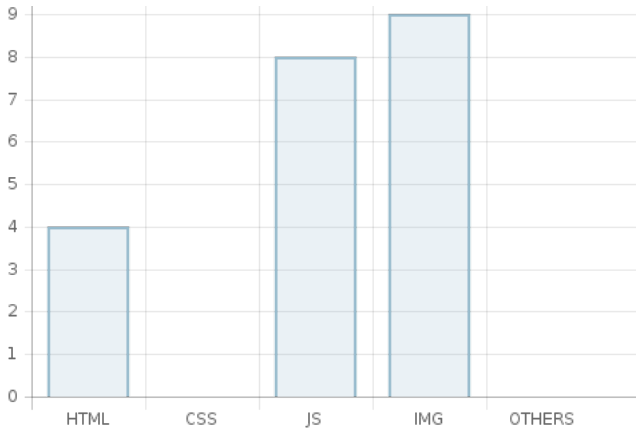
SIMULATED VISITOR: Chrome London 10.0/2.0Mbps (Latency: 28 ms)

Largest Contentful Paint	Total Blocking Time	Cumulative Layout Shift	Speed Index
0.35 sec	8 ms	0.73	2112 ms

Weight by resources type



Requests by resources type



Detected technologies

Font Awesome · OWL Carousel · Twitter Bootstrap · Varnish · jQuery

Tips and best practices

Accessibility

Your Optimization Priorities

0/100

#2530

Users should be able to specify www in the URL, or not

Some users are accustomed to access a website by adding www, and others do not.

We advise you to use a (permanent) redirect to point one of the addresses to the other.

How to fix the issue?

You should perform a 301 redirect from `https://leite5.github.io/` to `https://www.leite5.github.io/`.

0/100

#2381

Explain the purpose of each form field

Clarify the purpose of each field will facilitate the user experience on your website.

A form is composed of several fields that must be the most explicit possible for the user to quickly understand their function.

Define a label

You should prefer using the `label` tag:

Example

```
<label for="name">Fill your name:</label>
<input id="name" type="text" name="name">
```

Otherwise, you can use the `aria-label` or the `title` (not supported by all screen readers) attributes. Read more. Note that using the placeholder attribute is insufficient.

This page contains **4 fields without any explanations**:

- `<input class="contactus" placeholder="Full Name" type="text" name="Full Name">`
- `<input class="contactus" placeholder="Email" type="text" name=" Email">`
- `<input class="contactus" placeholder="Phone" type="text" name="Phone">`
- `<textarea class="textarea" placeholder="Message" type="text" name="Message">`

Did you know?

#72

No <noscript> tag is detected

When a web page uses scripts, it is advised to set at least one `noscript` tag. It is required to display a message when JavaScript is disabled by the user.

Example

```
<script  type="text/javascript">
   document.write('Hello World!')
</script>
<noscript>Your browser does not support JavaScript!</noscript>
```

Well done, these best practices are respected

100/100

#2569

✅ No empty element detected

Some tags, such as `<p>`, `<li>`, `<button>`, `<legend>`, `<caption>`, `<figcaption>` and `<quote>` elements are not allowed to be empty. Empty tag make it difficult for some screen readers to understand the page.

Either add `aria-hidden` attribute to the empty element or remove it from your page entirely.

Example

```
<p aria-hidden="true"></p>
```

100/100

#2334

✅ This page defines a lang

Perfect. Your page defines a `lang` attribute that will allow screen readers to correctly understand your website.

100/100

#2395

✅ No 'refresh' <meta> tag

It is not recommended to automatically force a page refresh. This behavior disturbs users who are not able to control it.

This page does not use `meta refresh` tag. This is a best practice.

100/100

#2410

✅ No empty "src" attribute detected

The "src" attributes link the current page to other resources. There is no justification for using them with empty values.

All your `src` attributes precise a target. That's a best practice.

100/100

#2423

✅ You specify a consistent label on your links

A link is more attractive if the text describes what is behind it. You can also take the opportunity to use keywords in these texts, to improve your page's SEO.

Describe the link in your `<a>` tag, rather than indicating the link itself. Example:
`<a href="http://mylink.com/">My description</a>`

The content of your `<a>` tags are different from the link itself. That's a best practice.

100/100

#2468

✅ The main title of the page is the first stated title

Facilitate the work of screen readers by indicating your main title (`<h1>`) before any other title.

We recommend you to keep a coherent hierarchy among your titles (h1, then h2, then h3, etc).

100/100

#2501

✅ Each form defines a submit button

HTML forms are used to send data. For accessibility purposes (e.g. using a screen reader), all your forms must include a submit button.

How to specify a submit button?

You can send the form data using two kind of elements:

- `button`
- `input`, with the `type` attribute using one of these values: `submit`, `image` or `button`

All the forms contain a component to send the data.

100/100

#2523

✅ All labels refer to an element

The `for` attribute associates the label to another element of the page and help screen readers to better interpret your content.

Label and for attribute

A label describes an element (a text to fill, a checkbox, etc.). When a user clicks on a label associated with a radio button, the option will be directly selected, improving the user experience.

How to use a label?

Associate the label to an element of the page by indicating the ID of the element. Example:

Example

```
<form action="/action">
  <label for="myId">
  <input type="radio" name="myOptions" id="myId" value="1" >
```

Your Optimization Priorities

0/100

#2405

! Avoid http-equiv <meta> tags

HTTP headers are more efficient than the http-equiv meta tags.

The <meta http-equiv=""/> tags

The `http-equiv` meta tags provide the web browser with information similar to HTTP headers. For example, defining the meta <meta http-equiv="content-type"/> is the same as sending the HTTP Content-Type header.

There are two disadvantages in using http-equiv meta tags:

- Going through the meta requires to interpret the beginning of the HTML page, which is slower than going through the HTTP headers in terms of performance
- If the HTTP header is already present, the meta is ignored

In which cases are the <meta http-equiv=""/> useful?

Only one case can justify the presence of these meta tags: if you don't have access to the configuration of your server, and that is to say to the HTTP headers.

However, we recommend that you use a configurable server so that you can establish the most efficient site possible.

This page contains 1 http-equiv meta tag. If possible, you should replace it:

- X-UA-Compatible

Well done, these best practices are respected

100/100

#2566

✓ Your HTML response is not too heavy

Why reduce the code amount of a page?

Before a web page can be displayed, the browser must, among other things, download it, parse it and model it into a document that can be understood by the rendering engine. If the amount of code contained in the page is too large, these steps are slowed down and the rendering is delayed.

How to reduce the amount of code?

Your HTML response should contain only the information that is immediately necessary to display the visible area of the page. Move inline information to external files (JS for scripts, CSS for styles, asynchronous queries for additional content) and simplify the HTML structure of your page.

100/100

#2575

✅ No Mutation Events detected in your scripts

To capture DOM events, do not use Mutation Events. Alternatives exist.

Good concept, bad implementation

When developing complex JavaScript applications, you may need to know when the DOM node tree has changed. Introduced in 2000 in the DOM, Level 2 specification to provide a solution to this need, Mutations Events are browser-initiated events that let you know when a DOM node is added, removed, or deleted.

Mutation Events, however, present major performance problems. First, they are synchronous, i.e. they prevent other events in the queue from being fired (if those events are used to update the UI, this will cause some lag). Second, they are implemented as browser events, thus traverse the DOM tree from the targeted HTML element to the parent element which listens for the event, clogging the JavaScript thread along the way.

Mutation Events have been deprecated in 2016 in the DOM, Level 3 specification.

Mutation Observers to the rescue

If you need to watch for changes being made to the DOM tree, you should use the `MutationObserver` interface (DOM4 Living Standard). Mutation Observers are asynchronous, processed in batches, and observe specific or all changes to a node. They are more efficient in terms of CPU usage than browser events and therefore cause fewer to no UI freeze.

Learn how to use Mutation Observers (Mozilla Developer Network).

100/100

#2353

✅ Your JavaScript resources don't block your page loading

JavaScript can significantly slow down a page display, especially if it is necessary to download an external script.

Defer the use of JavaScript as much as possible to provide a faster start for the page display.

How can I fix this?

First of all, distinguish what portions of your JS is critical and must be loaded as soon as possible, and put them in a specific external file. Keep this file as streamlined as possible, and defer the parsing or execution of all other JS files.

Use one of the methods below to defer parsing for external JavaScript files:

- use the async attribute;
- use the defer attribute;
- append the script to the DOM in JavaScript during the onload event;
- make sure your scripts are placed at the bottom of the page (ideally at the end of the body).

100/100

#2356

✅ You specify a character set in the response HTTP Header

Specify the character set used in the **Content-Type** HTTP header allows the browser to parse immediately the page.

When the browser receives bytes from your server, it needs to identify the collection of letters and symbols that were used in writing the text that was converted into these bytes, and the encoding used for this conversion, in order to reverse it. If no information of this kind has been transmitted, the browser will try to find recognizable patterns within the bytes to determine the encoding itself, and eventually try some common charsets, which will take time, delaying further processing of the page.

How to specify a charset in the Content-Type header?

In the following explanation, we will consider UTF-8 as the targeted character set but please remember that the character set declared in your Content-Type HTTP Header must reflect the character set used to encode the file, which may not be UTF-8.

On **Apache 2.2+**, the configuration of UTF-8 as a default character set for your `text/plain` and `text/html` files involves the `AddDefaultCharset` directive:

Example

```
AddDefaultCharset utf-8
```

For other types of files, you'll need the `AddCharset` directive:

Example

```
AddCharset utf-8 .js .css ...
```

On **nginx**, you'll need to make sure that the `ngx_http_charset_module` is loaded, then use the `charset` directive.

Example

```
charset utf-8;
```

Here too, it is possible to refine the scope so that other types of files than text/html are delivered in utf-8, using the directive charset_types:

Example

```
charset_types text/html text/css application/javascript
```

100/100

#2370

✅ You do not use CSS @import

Using CSS @import allows to add external stylesheet. In fact, browsers cannot download them at the same time, this may add a delay to the rendering of the page. It is better to use the `link` tag. See more information.

100/100

#2416

✅ This page uses an appropriate number of DOM elements

The number of DOM elements influences the complexity of the webpage and DOM access in JavaScript.

A well-designed webpage can offer rich content while maintaining a reasonable number of DOM elements. Read more about this here.

We recommend creating pages that contain less than 1000 DOM elements.

There are 283 DOM elements on this page.

100/100

#2445

✅ You don't execute the same script several times

A library or an external script is usually intended to be called once per page. However, the use of widgets can lead to several useless executions.

Duplicate scripts

It is common to see scripts used multiple times on the same page. The most common cause is the integration of social network widgets. It can be useful to find them several times on the page. That is not a bad practice. However, you should be careful that those scripts don't slow down your page.

What happens when a script is included 2 times in the code? How many times is it loaded? Parsed? Executed?

Most modern web browsers download only once a script included 2 times. An exception persists: Firefox, which will load the resource as many times as mentioned if no effective caching policy is configured.

Apart from this exception, performance issues come during the parsing and execution of the scripts. Indeed, if a script is placed three times in the code, it will be parsed and executed 3 times, on all browsers.

Do not hesitate to read this article on that topic.

How to fix it?

There is a solution to use a script several times without parsing and executing more than once. You need to write some JavaScript code that checks if the script is present. If the script is already included, it just uses it, otherwise it injects it and uses it.

Consider the following example with the Facebook widget, described in the article. Whenever you want to integrate this functionality into your page, it is necessary to include the following code:

Example

```
(function(d, s, id){
 var js, fjs = d.getElementsByTagName(s)[0];
 if (d.getElementById(id)) {return;}
 js = d.createElement(s); js.id = id;
 js.src = "//connect.facebook.net/en_US/sdk.js";
 fjs.parentNode.insertBefore(js, fjs);
}(document, 'script', 'facebook-jssdk'));
```

The bold line checks the presence of the script in the document. So the script is included, parsed and executed only during the first call in the page. Other calls will fall in the case of the bold line, and therefore will just use the script that is already included and executed.

100/100

#2539

✅ The page doesn't use client-side redirection

No client-side redirection (window-location, meta refresh...) has been triggered to display the page. Client-side redirections are to be avoided in any way possible.

100/100

#2550

✅ Your scripts seem to be injected efficiently

The script injection through the `document.write` instruction delays the rendering of your page and/or the interactivity for the user.

Do you need to inject a script?

`document.write` is sometimes used to inject a script. That is a bad practice. Example:

Example

```
document.write('<script src="' + src + '" type="text/javascript"><\/script>');
```

As described in this article, you should prefer the "createElement-insertBefore" pattern:

Example

```
var sNew = document.createElement("script");
sNew.async = true;
sNew.src = "http://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js";
var s0 = document.getElementsByTagName('script')[0];
s0.parentNode.insertBefore(sNew, s0);
```

On this page, no `document.write` injects a script.

Cache policy

Your Optimization Priorities

0/100

#2437

Set a far future cache policy in 17 requests

Defining several days of cache retention for your static resources will reduce the load on your server.

The Expires header explained

Some of your resources use the `Expires` HTTP header to get an effective caching policy—this is a best practice. However, you should consider improving its configuration to make the most of the caching mechanisms. Here is an example of the `Expires` HTTP header:

Example

```
Expires: Thu, 25 Dec 2014 20:00:00 GMT
```

When you **deploy a new version of your website, remember to rename static resources** that have been modified. If you do not change their names, your users will keep resources corresponding to the old versions stored in their caches, and they may find themselves on an unstable version of your page. For example:

Example

```
myresource.min.20140101.js
```

Read the Yahoo! guidelines on this subject.

Recommended Expires header setting

We recommend setting the `Expires` HTTP header, so the date is between 2 days and 1 year.

This page contains 17 resources that **do not have a far expiration date**:

- https://leite5.github.io/images/loading.gif
- https://leite5.github.io/images/logo.png
- https://leite5.github.io/js/jquery.min.js
- https://leite5.github.io/js/popper.min.js
- https://leite5.github.io/js/bootstrap.bundle.min.js
- https://leite5.github.io/js/jquery-3.0.0.min.js
- https://leite5.github.io/js/plugin.js
- https://leite5.github.io/js/jquery.mCustomScrollbar.concat.min.js
- https://leite5.github.io/js/custom.js
- https://leite5.github.io/js/owl.carousel.min.js
- https://leite5.github.io/images/img.png
- https://leite5.github.io/images/service1.png
- https://leite5.github.io/images/service2.png
- https://leite5.github.io/images/service3.png
- https://leite5.github.io/images/why2.png
- https://leite5.github.io/images/why1.png
- https://leite5.github.io/images/ask.jpg

Well done, these best practices are respected

100/100

#70

✅ You do not use too long inline scripts

Any script with a significant size should let the browser cached them in order to reduce loading time/improve performance of your returning visitor.

Inline scripts / cache policy

"inline" scripts allow to integrate easily small portions of scripts directly in the HTML code. Example:

Example

```
<script type="text/javascript">
   (function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']...,'/analytics.js','ga');
   ga('create', 'UA-11111111-1', 'mywebsite.com');
</script>
```

By doing so, you avoid making a request to the server to retrieve the resource. So inline scripts represent a performance gain if you want to integrate small scripts.

However, once a script has a fairly substantial size, we advise you to outsource it and perform a request to retrieve it. So you will benefit from the cache mechanism.

What should I do?

Outsource your scripts with more than 1500 characters in one or more separate files.

100/100

#2352

✅ The 'Vary: Accept-Encoding' header is defined

The `Vary: Accept-Encoding` header allows to cache two versions of the resource on proxies: one compressed, and one uncompressed. So, the clients who cannot properly decompress the files are able to access your page via a proxy, using the uncompressed version. The other users will get the compressed version.

100/100

#2430

✅ You specify resource cache expiry headers

Cache headers (`Cache-Control` , `ETag` , formerly `Expires`) are essential for an effective cache policy, and will greatly impact the loading time of your pages during future visits.

The `Cache-Control` header

Each resource can define its caching rules via the `Cache-Control` HTTP header. The `max-age` property defines the duration of the caching (in seconds), and can be accompanied by instructions for caching resources on proxy servers, located between the browser and the server issuing the resource.

The following header indicates that the response can be cached on proxy servers and on the browser (`public` , as opposed to `private` where only the browser is allowed to do the caching) for two hours:

Example

```
Cache-Control: private, max-age=7300
```

If some of your resources do not need to be cached, you can also indicate this explicitly:

Example

```
Cache-Control: no-store
```

The `Expires` header

`Expires` is the earliest HTTP header for managing resource caching, and will help you manage the cache for browsers that do not support `Cache-Control` .

When using the `Expires` header, you can define an expiry date for each resource: as long as the date has not expired, the browser will either store or use the resource stored in the cache.

The expiry date of the resources is set using the `Expires` HTTP header:

Example

```
Expires: Thu, 25 Dec 2014 20:00:00 GMT
```

You can specify a long expiry date for static resources (maximum 1 year), and a shorter expiry date for resources that may change (minimum 48 hours).

Reassessment of resources when the cache expires

If no cache policy is set for a resource or the duration of its caching is exceeded, the browser makes a new request to download a required resource.

To prevent the browser from downloading a resource that has not been modified since it was cached, use the `ETag` HTTP header. Each version of a resource can be associated with a validation token. When a resource's cache expires, the browser will ask the server again for the resource, passing this token with the `If-None-Match` HTTP header containing the token value. The server will compare its version of the token with the one provided. If the resource has not been modified, the server will allow the browser to renew the caching of the resource without downloading the resource again, via an 304 "Not Modified" HTTP response.

Deliberate cache invalidation

When releasing a new version of your site, remember to rename static resources that have been modified (versioning), in order to force browsers to download these new versions, instead of using cached resources, to prevent users from finding themselves in an unstable version of your page. For example:

Example

```
maressource.min.20140101.js
```

To learn more about HTTP caching, please see Google's recommendations.

Congratulations! Your resources are cached.

Well done, these best practices are respected

100/100
#80

✅ No frameset, frame and noframes tags detected

These tags are obsolete, due to several issues related to the navigation consistency, SEO or browsers' bookmark features for example.

None of these tags is detected on this page.

The use of the iframe tag is prefered.

100/100
#82

✅ No Java applets detected

Java applets are considered obsolete in 2015. HTML5 is powerful and more widely supported. Using Java applets can lead to compatibility issues and may send negative signals to your users (e.g. the browser indicating that content was blocked because it could be dangerous).

Congratulations, this page doesn't contain Java applets.

100/100
#89

✅ Do not use <bgsound> tag

No `bgsound` tag detected. This is a good practice: this element is not a HTML standard. See more information.

Use the audio tag to deliver audio content on your page.

Example

```
<audio src="my-audio-file.ogg" autoplay>
   Your browser doesn't support the audio element.
</audio>
```

100/100
#99

✅ You do not use links to Word documents

.doc and *.docx* documents do not guarantee compatibility with all major operating systems. It's recommended to use PDF documents.

This page contains only standard links.

100/100
#2382

✅ No deprecated attributes are detected in the <body> tag

Some layout attributes are deprecated in HTML 5: `alink` / `background` / `bgcolor` / `link` / `text` / `vlink`

Congratulations, this page doesn't use deprecated attributes in the `body` tag.

Prefer using CSS instructions instead.

100/100

#2424

✅ No BOM (Byte Order Mark) detected

Some parsers are not able to interpret a page with a BOM in it.

What is the BOM?

The BOM is a hidden character located at the beginning of the page, aiming at helping to determine what encoding the page uses. But the best practices of the web prompt the use of the HTTP `Content-Type` header to define the encoding used by the page. The BOM has no reason to be in this context.

Apart from the fact that this type of indicator is useless on the web, it can lead to a certain number of issues. This is the case for example of the W3C validation that is going to try to interpret the first character which corresponds to the BOM. Then, the document will not be valid.

No resource uses a BOM.

100/100

#2454

✅ Your characters are encoded in UTF-8

Your content is readable by the largest number of web users.

Encoding of characters

The encoding of characters indicates to the web browser how to interpret the bytes of the web page in order to convert them in readable characters by the user. Lots of encodings exist so that all the characters specific to each language are represented. Be aware that the supported groups of encoding are different from an engine to another one. If you ever use an encoding too specific, a lot of web users won't be able to accurately interpret the page.

Why choosing the UTF-8?

The UTF-8 is known for being an encoding supported by almost every web user, and taking into account a large range of characters. Its universality is then particularly well suited to the web environment.

100/100

#2488

✅ No Flash resource detected

Flash is considered obsolete in 2015. HTML5 is more powerful and more widely supported. Using Flash can lead to compatibility issues and may send negative signals to your users (eg the browser warning that content was blocked because it could be dangerous).

Moreover, Google shows directly in its mobile search results that the page may not work on the user's device.

Congratulations, this page doesn't contain Flash resources.

100/100

#2493

✅ Silverlight plugin is not used

Silverlight plugin is considered obsolete in 2015. HTML5 is more powerful and more widely supported. Using Silverlight plugin can lead to compatibility issues and may send negative signals to your users (e.g. the browser indicating that content was blocked because it could be dangerous).

Congratulations, this page doesn't use Silverlight plugin.

Your Optimization Priorities

90/100

#2388

☑ Minify JavaScript

Compacting JavaScript code can save many bytes of data and speed up downloading, parsing, and execution time.

Minify JavaScript for the following resources to reduce their size by 3.7KiB (3% reduction).

- Minifying leite5.github.io/j[...]n.js could save 3.7KiB (3% reduction).

There are many tools to minify JavaScript files. You can try YUI Compressor or JSMin, recommended by Google.

90/100

#2389

☑ Optimize your images

Properly formatting and compressing images can save many bytes of data.

Optimize the following images to reduce their size by 3.9KiB (50% reduction).

- Losslessly compressing leite5.github.io/i[...].png could save 2.3KiB (50% reduction).
- Losslessly compressing leite5.github.io/i[...].png could save 1.6KiB (50% reduction).

Images may contain data unnecessary for their use on the web. This data can increase their size significantly. Some tools automatically remove this unnecessary data without loss of quality and thus reduce your image sizes.

Many image optimization algorithms depend on each image format. Some of them are included in graphic software like Photoshop or GIMP:

- PNG: Zopfli-png, PNGOUT, OptiPNG, AdvPNG, PNGCrush, PNGQuant...
- JPG: JPEGOptim, MozJPEG, Jpegtran, Guetzli...

FileOptimizer (Windows), ImageOptim (Mac) or Trimage (Linux) are software that combine several algorithms in one place. They will find the best possible optimization and encoding for every image, with or without quality loss.

Did you know?

#2443

This page does not load too much data (579kB)

A too high page weight slows down the display, especially on low-speed connections. This can lead to frustration for users paying for data (see whatdoesmysitecost.com).

Evaluate the Weight of my Web Page

According to HTTPArchive, in July 2019, the average weight of a web page is 1,95MB.

How to reduce the weight of my page?

You can report to our "Data amount" category to discover the possible optimizations in your case. Images are often involved.
Moreover, make sure to build your web pages to load data that is essential to the user experience (rendering optimization of the critical path).
For other content (social networking plugins, advertising, content at the bottom of the page ...), it is better to delay the loading (asynchronous, lazy-loading ...), so they don't override priority contents.

We have established the weight distribution of the page by resource type:

- **Images :** 52,65% of total weight
- **JavaScript :** 45,62% of total weight
- **Texts :** 1,73% of total weight

Here is the weight of the 10 heaviest resources over the network, and that are necessary to load the page:

- https://leite5.github.io/js/plugin.js (174kB)
- https://leite5.github.io/images/img.png (107kB)
- https://leite5.github.io/images/why1.png (71kB)
- https://leite5.github.io/images/why2.png (53kB)
- https://leite5.github.io/images/loading.gif (36kB)
- https://leite5.github.io/js/jquery.min.js (31kB)
- https://leite5.github.io/images/ask.jpg (22kB)
- https://leite5.github.io/js/bootstrap.bundle.min.js (21kB)
- leite5.github.io/js/jquery.mCustomSc[...]ar.concat.min.js (13kB)
- https://leite5.github.io/js/owl.carousel.min.js (12kB)

Well done, these best practices are respected

100/100

#2325

All your resources are served from a consistent URL

Resources with identical content should be served from the same URL to avoid duplicate downloads and additional requests.

Different URLs for the same resource

If you use different URLs to serve identical content, browsers have to request this content from the server several times, and your page will trigger unnecessary requests for data. If the same URL is used, the browser will only send one request and will use the response wherever the resource is requested. You should also use the same URL to reference the same content on several pages to benefit from the cache mechanism.

How to fix this?

Serve each resource from a unique URL. If you have to request the same resource several times, the same URL should be used to retrieve it every time.

100/100

#2364

You do not need to minify the HTML resources

Your HTML is minified. Learn more about minifying HTML.

You get 100/100, while your resources are not minified? It means that we considered that the gains provided by the minification process were not significant enough to be reported.

100/100

#2384

✅ Compression is enabled

You have compression enabled. Learn more about enabling compression.

100/100

#2387

✅ Your CSS resources are minified or don't need to be

Your CSS is minified. Learn more about minifying CSS.

There are many tools to minify CSS files. You can try YUI Compressor or cssmin.js, recommended by Google.

100/100

#2421

✅ All your resources have a size < 1MB

Too heavy files should not be loaded on a web page. Be sure you need this resource to load the page.

100/100

#2436

✅ You serve scaled images

If your images are larger than their display area, the browser will download unnecessary data (and perform unsupervised resizing).

Avoid resizing images on the browser side

Resizing images on the browser side to reduce their rendering size is not recommended.

When the browser needs to display an image on your page, it does everything it can to adapt it to its rendering surface. If the image is too large, it will reduce it.

Provide images adapted to the display dimensions to prevents unnecessary data from being sent over the network, which reduces page loading time.

And because embedded browser algorithms are not as good as those of image manipulation tools, you will get a more satisfying visual result by resizing your images upfront, rather than letting the browser do it.

Serve Responsive Images

Several methods exist, to serve images adapted to the browser regardless of screen resolution or device pixel density. We recommend reading the following resources:

- "Responsive images" on the Mozilla Developer Network
- Picturefill, to start using the <picture> element in browsers that do not support it
- RICG, group of developers working on responsive images

The downloaded images match their display area: performance and visual quality are preserved.

100/100

#2446

✅ Lazyloading is set (or not required)

Load images above the fold line first, that means all the images that are visible without any scroll of the page by the visitor.

Webpage and images

Images commonly represent more than 60% of the total weight of webpages. By loading initially the only images that are visible without page scrolling, you'll reduce bandwidth consumption as for your server and your visitors. If an image has to display after any visitor's action (as scrolling), then you'd better load it on demand, when necessary (lazyloading).

How to set up lazyloading?

You can get information concerning the features and plugins proposed by your page's frameworks/CMS. You should find some easy-to-set solutions.

If you have to implement lazy-loading yourself, the simplest way may be to activate the native feature introduced by Chrome in 2019: on all targeted images, add `loading="lazy"`. The feature is implemented by recent browsers or will be in the near future. In the meantime: on browsers that don't support native lazy-load, the attribute will have no effect.

Example

```
<img src="image.jpg" alt="my image" loading="lazy" width="50" height="50" >
```

Native lazy-loading, however, is the solution that offers the least control, and is sometimes far too eager, loading images that do not require to be loaded so soon (read more).

Some Javascript libraries could help you implement lazyloading with more granularity.

With jQuery, you could use a plugin like jQuery Lazy Load, dedicated to images lazyloading.

100/100

#2450

✅ This page does not send cookies > 100kB

Keep the size of cookies as low as possible to minimize the impact on the loading time.

HTTP cookies are used to track a user to customize the page according to their profile. They are sent as an HTTP header from the web server to the browser. Then, each time the browser accesses to the server, it sends a request containing the cookie received at the first response. See more information.

100/100

#2453

✅ This page does not send too many cookies

Keep the size of cookies as low as possible to minimize the impact on the loading time.

HTTP cookies are used to track a user to customize the page according to their profile. They are sent as an HTTP header from the web server to the browser. Then, each time the browser accesses to the server, it sends a request containing the cookie received at the first response. See more information.

100/100

#2461

✅ 7 images use the PNG format in the right way

The choice of the right format for an image allows to reduce its weight.

The PNG format

The PNG image format is intended to the images requiring the transparency, or else to the small images having little details and colours.

Prefer the JPEG format...

The main problem of PNG format is to not support quality loss. Indeed, a format such as JPEG offers to "downgrade" the quality of the image without being perceived by the user. Doing so, you can reduce the quality of the image of about 25% whithout the user realizes it.

An image with an consequent weight will be therefore better compressed by using the JPEG format.

...or the PNG-8 format

In the case where your image necessarly requires using the transparency mecanism, not borne by the JPEG format, you should convert your "standard" PNG image, into PNG-8. This format, based upon a 256 color palette maximum allows to decrease the weight of the image whithout significantly affecting its overall quality. Tools such as pngquant or else tinypng will suport you in this procedure.

At last, if the quality given by the PNG-8 format is not appropriate for your image, you can get information on the possible advanced technics to get this behaviour without having an image in PNG format. For example, it is possible to halve your image in 2 JPEG images, one including the transparency data and the other one including the data related to the colours, and to gather the image on the customer's side with a CANVAS element.

For further information, please visit this article about image compression.

The choice of the PNG format on the images of this page doesn't have negative effect.

100/100

#2465

✅ 1 image(s) use the GIF format properly

Choosing a suitable format can dramatically reduce the weight of an image.

The GIF format

The GIF format is suitable for small animated images (< 100kb). If you do not use animation, the PNG and JPEG formats will be more suitable and lighter, once optimized. Replace each animated content with an `<video>` element containing an MPEG-4/H.264 `<source>`.

Exemple

```
<video loop="loop" poster="video_poster.jpg">
  <source src="video.mp4" type="video/mp4">
</video>
```

The GIF images of this page have not a negative impact.

100/100

#2486

✅ Redirects are lightweight

An HTTP redirect's content is not used by web browsers. So, it provides an unnecessary weight that should be as small as possible.

Redirects on the web

Redirects can be temporary (302 HTTP code) or permanent (301).

This mechanism is for example used for secondary domains (.net to .com), the language detection mechanisms, etc.

How to fix the issue?

First, ensure that the redirect is unavoidable. In this case, it is necessary to reduce or completely remove the contents of the server responses for redirects.

A redirect should be < 1kB.

There isn't too heavy redirect.

Your Optimization Priorities

0/100

#2344

2 resources are unreachable

You should avoid requesting unreachable resources.

Warning notifications that some requests related to your page are encountering errors can be due to:

- errors in your HTML, CSS, or JavaScript resources;
- an unhandled error on the server side;
- or a problem with a service used by your page.

The following HTTP codes were returned:

- 404 (net::ERR_ABORTED):
  - leite5.github.io/cdnjs.cloudflare.com/a[...]ry.fancybox.min.js
  - leite5.github.io/cdnjs.cloudflare.com/a[...]ry.fancybox.min.js

These errors can affect content on and behaviors of your website and cause unnecessary network traffic, which affects the loading time of your page.

Did you know?

#2543

Resources distribution by domain

This page loads data from 1 domains. This best practice retrieves the following metrics for each of these domains:

- Loading Time (Cumulative): total time spent to load all the resources
- Server Time (Cumulative): total time spent to retrieve the responses from the server (TCP connection + wait for first byte)
- Weight: data amount loaded
- Number of requests

Here is the list of all the domains used by the page:

Domain	Time (ms)	Server Time (ms)	Weight (kB)	Requests
leite5.github.io	4570	1890	578	21

Well done, these best practices are respected

100/100

#2339

✅ No redirect detected

The redirects trigger avoidable roundtrips on the network and increase the page loading time.

HTTP redirects

The HTTP redirects inform the browser that the desired content is accessible from a different URL. They trigger a new HTTP request to retrieve the target resource and return an HTTP code between 300 and 399. See the specifications of HTTP redirects.

How to solve the issue?

Allow the user to directly access your content without redirects, or determine and improve what causes these excessive loading times on your redirects. See recommendations from Google.

This page does not use redirects before accessing the right content.

100/100

#2403

✅ You do not use too much "prefetch" on the links

No resource is prefetched on this page.

You should limit the number of prefetched requests to avoid network congestion. Identify your users' behavior to prefetch the most requested resources.

Note that some browsers automatically block the number of prefetched resources. For instance, Internet Explorer 11 limits this mechanism to 10 requests.

100/100

#2466

✅ None of your images uses only one color

One request to an image composed of a unique color unnecessarily increases the data amount transmitted on the network.

Fetching an image composed of a single color is useless. CSS styles can achieve the same result in a more efficient way.

For instance, to draw a simple circle, you can use the following code:

Example

```
#myElement {
 background:#ff0000;
 border-radius:50%;
 width:160px;
 height:160px;
}
```

All the images of this page are composed of several colors.

Your Optimization Priorities

0/100

#2491

One ID is duplicated within your HTML

Using the same ID on several elements can have side effects, especially during JavaScript executions or when applying CSS rules.

IDs explained

Each element of a web page can be identified thanks to the `id` attribute:

Example

```
<p>
  <span id="mySpan1"></span>
</p>
```

These IDs allow you to manipulate your elements with CSS or JavaScript instructions.

How to properly use IDs?

You must ensure identifiers are not duplicated within the page. If you want to share a property or a behavior between multiple items, you have to use the `class` attribute, which is dedicated to this purpose:

Example

```
<p>
  <span class="mySpans"></span><span class="mySpans"></span>
</p>
```

The following ID is used multiple times within your page:

- `box_ho`, used 2 times

0/100

#2379

Provide a favicon

No favicon found on this page. You should put one in your `head` tag as shown below:

Example

```
<link rel="icon" type="image/png" href="/path/favicon.png" />
<!--[if IE]><link rel="shortcut icon" type="image/x-icon" href="/path/favicon.ico" /><![endif]-->
```

Favicon is a small image providing an icon to a website. It's located in the root of your server and the browser will always request it. It is better not to respond with a 404 HTTP code (not found).

Moreover, this file will be asked on every requested web page, so make it cacheable: the client will request it only once. See more information.

Did you know?

#71

No HTML code is commented

Comments allow you to detail a portion of code and help you navigate more efficiently in the DOM. However, make sure no sensitive information is exposed in your comments.

Well done, none of your comments contains HTML code.

Well done, these best practices are respected

100/100

#2383

✅ The CSS styles are separated from the HTML tags

Separating HTML tags and CSS directives improves code readability and promotes factorization.

How to define CSS styles

CSS styles are used to format the page. You can use one of three main methods to define them:

- declare styles in a specific CSS file;
- declare "inline" styles (<style> tag in your HTML template);
- declare styles with the "style" attribute of a HTML tag.

How can I improve my page?

We recommend grouping your CSS styles in `<style>` tags or in separate files. That way, the HTML is only responsible for providing the structure of the page, and its layout is outsourced. The `<style>` attribute should only be generated by some JavaScript code (e.g., if you need to know the screen size).

This page does not use the `style` attribute.

100/100

#2398

✅ You do not gzip/deflate PNG images

Compression has a cost, on the server and the client browser. You should enable it only if it is effective.

PNG and compression

PNG images does not support gzip compression. It is useless to perform compression operations on server side, and decompression on client side for this kind of files.

This page does not compress images. This is a best practice.

100/100

#2448

✅ The extensions of your resources are consistent

The extension of a resource allows to identify easily its content type. You have no reason to indicate a different extension of the actual content type of the resource.

For instance, the file `resource.js` must use a `Content-Type` HTTP header equal to `application/javascript`. You can check the `Content-Type` HTTP header value using a debugger tool (e.g.: DevTools on Chrome).

100/100

#2449

✅ All resources define their content type

Each resource should define its content type in order to facilitate their interpretation by web browsers.

You have no reason to hide the type of a resource.

We advise you to set the "Content-Type" HTTP header on every resources of the page.

Your Optimization Priorities

0/100
#84

The <meta> 'description' should not be empty

The page should define a unique description.

Description in search engines

The description of the page may be directly displayed in search engine results pages (SERP):

It allows you to control at best the entry preview in search engines, and to improve the click rate to your page. Learn more.

How to define a page's description?

Use `<meta name="description" content="page description">` and place it in the `<head>` tag.

The <meta> `description` tag is empty.

0/100
#69

Add alt attribute on <img> tags

Moreover, the `alt` attribute is also an important criterion for SEO. Indeed, search engines crawlers cannot parse graphic contents. That is why they use the alternative text to return consistent results, like in Google images.

Example

```
<img src="product.jpg" alt="My product description"/>
```

The `alt` attribute is used in several cases unrelated to SEO:

- When a screen reader is in use for accessibility purposes;
- While image is loading, particularly for slow connections;
- When the image file is not found.

You have 9 `img` tags, but the following tag does not define the `alt` attribute:

- `<img src="images/img.png" />`

If nothing seems appropriate for describing an image, you might set an empty text. We advise you to make sure the majority of your images define a relevant text. Read the W3C recommendations here.

0/100
#2503

robots.txt file should be defined

Indicate to web crawlers which URLs should be explored on your website.

The robots.txt file

Place your robots.txt file in the root of the website. It will be interpreted by the robots in charge of your SEO. It delivers instructions to specify the pages to explore by robots, like Google bot.

Note that these directives are indicative only. A lambda robot will not be blocked by the restrictions specified by the file.

We have not detected the robots.txt file on this website, you should define one:

- https://leite5.github.io/robots.txt

The other tips

0/100

#2399

Your site doesn't use Open Graph properties

You can help social networks understand information related to the page by using Open Graph properties.

The Open Graph properties explained

Several properties allow social networks to learn more about the page's content. We recommend using at least the required properties:

- <meta property="og:title" content="The title" /> Example
- <meta property="og:type" content="The type" /> Example
- <meta property="og:url" content="http://url.com/" /> Example
- <meta property="og:image" content="http://image.jpg" /> Example

This information is used to improve links between your page and various social networks, including Facebook. Read more about Open Graph here.

This page does not provide information to social networks.

Did you know?

#2457

This page contains 2 links

Two kind of links exist:

- **Internal links** that refer to pages with the same domain name;
- **External links** that point to other websites (must be relevant and point towards quality content).

If you reference many links, you can ask the SEO crawlers to consider only some of them, by adding the `rel=nofollow` attribute to the irrelevant ones (e.g., advertisements).

Here is the distribution of 2 links present in the page:

- 1 internal link (50,00%)
- 1 "follow" external link (50,00%)
- No "nofollow" external link (0,00%)

Well done, these best practices are respected

100/100

#78

This page defines <h1> and <h2> tags

We recommend putting page keywords in at least the h1 and h2 tags. Search engines use the h1, h2, and h3 tags for SEO purposes.
This page contains:

- 1 <h1> element(s)
- 4 <h2> element(s)
- 20 <h3> element(s)

100/100

#81

This page uses only standard image formats

The images that use a non-standard format may not be indexed by search engines.

Only these image formats are considered standard on the web: jpeg, jpg, png, gif, svg, ico, webp. You should consider an alternative to any other format.

Moreover, remember to treat the text around your images: some search engines analyze approximately the 10 words preceding and following the image in order to add a context to the image.

100/100

#83

✅ This page specifies a <title> tag

The page should define a unique title (using a <title> tag).

Use of titles by search engines

Once properly configured, the page title can be displayed in the search engine results page:

Using a suitable title is a major criterion for SEO. It allows you to control at best what is displayed in search results pages and determine the keywords you want your site pops out.

How to define the title of a web page?

The title of the page is specified into the `<title>` tag, which must be placed into the `<head>` tag, at the beginning of the code.

This page defines the title trough the `title` tag.

Here is the page's title:

```
Radiance
```

100/100

#90

✅ You do not use query strings in the URL

This URL does not contain any parameter.

A URL should be as readable as possible. For instance, `http://example.com?userId=332&group=MyGroup` is less readable than `http://example.com/mygroup/me`.

100/100

#2345

✅ The words are well separated in this URL

You should prefer the use of dashes in the URL.

URL and SEO

The words in the URL are among the many factors impacting the SEO: if you search the words `web performance` in a search engine, one criterion used by Google will be to check if the URL contains the words `performance` and `web`.

But for Google, underscores are not word separators: If your URL contains `web_performance` it will not help to highlight the page on the query `web performance` (`web_performance` is regarded as a unique word).

This behavior is however not common to all search engines. For example, Bing does not differentiate dashes and underscores.

Please note that Google does not penalize you for using an underscore in the URL.

How to resolve the issue?

We recommend using `-` instead of `_` on your new web pages.

This is more complex fix the issue on existing web pages, such as the one analyzed here, because you can't just rename the URL (you will lose all your SEO efforts). It is then necessary to set up a permanent redirect (HTTP code 301) retaining the old URL, that forwards to the new URL. Beware: many redirects on your site can also affect the visibility of your web pages. So do not use redirects if the number of relevant pages remains limited.

This URL respects the tip.

100/100

#2444

✅ Your <title> tag is an appropriate length

This page defines one title which contains less than 75 characters (see more information):

```
Radiance
```

The longer your title is, the more your chances are to see the search engines to truncate it or even to select another one from your page content.

100/100

#2467

✅ None of your titles are empty

`<h1>`, `<h2>` and `<h3>` tags should contain keywords related to the content.

The titles included on this page provide content.

Your Optimization Priorities

0/100

#2433

The Content Security Policy is missing

Protect your website from cross-site scripting (XSS) attacks by setting up a restrictive Content-Security-Policy.

XSS attacks explained

XSS attacks are a type of attack in which malicious data is maliciously added to websites. The number of vulnerabilities allowing these attacks is quite large, which is why it is as useful to prevent them as to limit their harmful effects.

You can protect your pages against these attacks and their effects by restricting execution to code portions either legitimized by the domain to which they belong or by a unique integrity token. The code that does not match this security policy will not be executed and the user will be informed.

You can learn more about XSS attacks on the Open Web Application Security Project (OWASP) Website.

Configure a "Content-Security-Policy" (CSP) HTTP header

Set up a "Content-Security-Policy" (CSP) HTTP header to prevent or limit the damage caused by an XSS attack. To specify a security policy configure your server so the response of the first resource contains the "Content-Security-Policy" HTTP header.

Here's an example:

Example

```
Content-Security-Policy: script-src 'self' https://apis.google.com
```

In this case, only scripts coming from the current host or https://apis.google.com will be executed.

Read more about the CSP HTTP header by consulting the CSP directives specification.

Please, be careful, if the header is misconfigured, some of your content, scripts, or styles may be blocked. That could cause unwanted side effects. Moreover, the restrictions apply to all pages of the website. We recommend you test the different pages of your website before deploying this header in your production environment.

No Content Security Policy on this page: it is more easily exposed to XSS attacks.

0/100

#2480

This page is exposed to "clickjacking" type attacks

Keep malicious people from integrating your pages into their websites.

Clickjacking explained

This kind of attack happens when your page gets integrated with a malicious website via <frame> or <iframe> tags. By doing this, attackers can persuade users that they are on your own page when they are not. The unsuspecting user may enter personal information that is visible on and thus vulnerable to the malicious website.

To avoid this, always indicate which domains have permission to integrate your pages.

How to prevent clickjacking?

There are two main ways to prevent that behavior.

1/ Configure a "X-Frame-Options" HTTP header. Configure your server so the main resource response includes the "X-Frame-Options" HTTP header.

Three values may be defined:

- `DENY` to prevent any frame or iframe from integrating the page;
- `SAMEORIGIN` to authorize only frames from the same domain name;
- `ALLOW-FROM uri` to indicate the domains allowed to integrate a page into frame (however is not compatible with some browsers)

2/ Define an explicit `frame-ancestors` directive into a Content-Security-Policy HTTP Header. "frame-ancestors" directive is a newer, hence supported by fewer browsers, approach that will allow your website to authorize multiple domains instead of only the current origin. Setting this directive to 'none' is similar to `X-Frame-Options: DENY`.

Which approach to choose? If you only have the current domain to allow, do set up the two security features, for better compatibility with older browsers. If you want to allow multiple domains, you should only implement the frame-ancestors security policy.

Neither the "X-Frame-Options" HTTP header nor the "frame-ancestors" security police are configured on this page; you are more likely to be exposed to clickjacking.

0/100

#2485

Disable the auto detection of resource type

Protect yourself from malicious exploitation via MIME sniffing.

MIME-Type sniffing explained

Internet Explorer and Chrome browsers have a feature called "MIME-Type sniffing" that automatically detects a web resource's type. This means, for example, that a resource identified as an image can be read as a script if its content is a script.

This property allows a malicious person to send a file to your website to inject malicious code. We advise you to disable the MIME-Type sniffing to limit such activity.

Chrome has been working on a feature called Site Isolation which provides extensive mitigation against exploitation of these types of vulnerabilities. Site Isolation is more effective when MIME types are correct.

How to prevent MIME-Type sniffing

Configure a "X-Content-Type-Options" HTTP header. Add the "X-Content-Type-Options" HTTP header **in the responses of each resource**, associated to the "nosniff" value. It allows you to guard against such misinterpretations of your resources.

On this page, **you should configure the following resources**, that risk being misinterpreted:

- https://leite5.github.io/js/jquery.min.js
- https://leite5.github.io/js/popper.min.js
- https://leite5.github.io/js/bootstrap.bundle.min.js
- https://leite5.github.io/js/jquery-3.0.0.min.js
- https://leite5.github.io/js/plugin.js
- https://leite5.github.io/js/jquery.mCustomScrollbar.concat.min.js
- https://leite5.github.io/js/custom.js
- https://leite5.github.io/js/owl.carousel.min.js

Did you know?

#2567

SSL Certificate

Your SSL certificate will expire on 04/14/2022. Update your certificate before that date.

What happens if my certificate expires?

Letting a certificate expire can have consequences for end users who will then see many error or alert messages while browsing the site, warning them of possible frauds, identity thefts or traffic interceptions. These alerts can have a very negative impact on the user's perception of the visited domain.

#2474

21 resources on this page are for public use

By default, the browser accepts to perform AJAX requests, or to retrieve web fonts, only on the same domain name of the page. So a font provided by toto.com can only be used by the pages of toto.com. This prevents misuse of your resources by any site.

Some resources are public, and explicitly want to be available to everyone (eg Google Fonts). In this case, the HTTP header `Access-Control-Allow-Origin` can be used with the value "*". You should, however, use this property if your resource has aimed to be used by the greatest number. Otherwise, we recommend that you keep the default, or set a specific domain name in the "Access-Control-Allow-Origin" HTTP header.

You should be aware of the following resources, that use a `Access-Control-Allow-Origin: *` HTTP header. Make sure they are actually intended to be used by pages from all domain names:

- https://leite5.github.io/
- https://leite5.github.io/images/loading.gif
- https://leite5.github.io/images/logo.png
- https://leite5.github.io/js/jquery.min.js
- https://leite5.github.io/js/popper.min.js
- https://leite5.github.io/js/bootstrap.bundle.min.js
- https://leite5.github.io/js/jquery-3.0.0.min.js
- https://leite5.github.io/js/plugin.js
- https://leite5.github.io/js/jquery.mCustomScrollbar.concat.min.js
- https://leite5.github.io/js/custom.js
- https://leite5.github.io/js/owl.carousel.min.js
- leite5.github.io/cdnjs.cloudflare.com/ajax/libs/[...]5/jquery.fancybox.min.js
- https://leite5.github.io/images/img.png
- https://leite5.github.io/images/service1.png
- https://leite5.github.io/images/service2.png
- and 6 others

#2553

Do all third-party resources deliver the right content?

If this page loads resources from third parties, you should ensure their integrity.

SubResource Integrity (SRI)

Use SRI to ensure that a third-party resource has not been tampered. Add the `integrity` attribute to `<script>` and `<link>` tags loading this kind of resource. Example:

Example

```
<script src="https://exemple.com/exemple-framework.js"
 integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1kPzQho1wx4JwY8wC"
 crossorigin="anonymous">
</script>
```

The `integrity` attribute value is equal to the base64-encoded hash (SHA) of the resource. The browser compares this hash with the downloaded content in order to determine if the resource matches the expected content.

You can create the SHA thanks to several tool. In command line, you can use openssl. You can also test some online tools, as srihash.org or report-uri.io. Learn more about SubResource Integrity.

Well done, these best practices are respected

100/100

#2397

✅ This page only uses secure content

This page was transmitted over the HTTPS protocol, and all resources are fetched using the HTTPS protocol. So there is no Mixed Content vulnerability.

100/100

#2463

✅ Your server only communicate in HTTPS with your web users

Take precautionary measures against attacks like "man in the middle" by making sure to only communicate in HTTPS with the server.

The HTTP Strict Transport Security (HSTS) Header

When you communicate with a server through a secure connection, every sent request towards this server should use the HTTPS protocol. The HTTP HSTS header allows to indicate to the browser that all the requests sent to the domain concerned must be done via HTTPS. If the URL is presented under "http://...", the web browser is automatically going to replace it by "https://...".

However, we advise you to not set this header unless your entire website serves its resources in HTTPS.

This page defines a HSTS header.

100/100

#2509

✅ The secure version is used systematically

Users using the HTTP version of the page must be redirected to the secure version.

HTTPs redirect

Even if a page is secure, visitors may still use the HTTP version (via an external link, or because they have bookmarked the HTTP version of the page, for example). Always set up a redirection so that users accessing the page using the HTTP protocol are redirected to a secure version.

Use the administration interface of your hosting provider or contact the administrator to set up an automatic redirection from the HTTP version to the secure version.

Your users are automatically redirected to the secure version of the page.

Your Optimization Priorities

60/100

#2299

Avoid DOM manipulation inside loops with jQuery

Working directly with the DOM has a cost. If you have to add elements to a node, you should prefer to append them once rather than one by one.

This page contains **1 DOM manipulation method(s) inside loop(s)**:

https://leite5.github.io/js/plugin.js

- for(var o=0;o<n;o+=1){var l=s(e.createElement("div")).addClass(i.slideClass+" "+i.slideBlankClass);a.append(l)}

0/100

#2302

Avoid excessive specificity on jQuery selectors

You are using too specifics jQuery selectors. It could impact performance: see more information. Here is an example of a good use of the library:

Example

```
$( ".data table.firstClass td.secondClass" );
// Better: Drop the middle if possible
$( ".data td.secondClass" );
```

We found some too specifics selectors on your website:

https://leite5.github.io/js/plugin.js

- jQuery(".mean-nav ul ul")
- jQuery(".mean-nav ul ul")
- jQuery(".mean-nav ul ul")
- jQuery(".mean-nav ul li")

https://leite5.github.io/js/custom.js

- $(".main-menu ul li.megamenu")
- $(".main-menu ul li.megamenu")

Did you know?

#2303

Additional information about your jQuery performance

jQuery is the most used JavaScript library. Upgrade your website performance respecting the jQuery best practices. We recommend that you learn the basics of the jQuery performance, reading the following link: http://learn.jquery.com/performance/.

Well done, these best practices are respected

100/100

#2470

Only one version of jQuery is loaded

Loading jQuery several times increases the page weight and complexify the maintainability of the code.

This page uses only one jQuery version, so it follows the best practice recommandation.