



We're moving to Discord!

In a few days, we will be leaving Slack in favor of Discord 🎉

👉 **Click here for more information (/concepts/100033)**

The trinity of front-end quality

*Have you noticed that this concept does not contain any link? By now, you should be able to find your own sources, based on the proper technical terms we'll keep giving you in the concepts, usually introduced **in bold font**.*

When talking about front-end quality, there are 3 main end goals that front-end developers keep striving for:

- **Accessibility**
- **Performance**
- **SEO**

SEO

SEO (Search Engine Optimization) refers to making sure your website is ranked relevantly high on search engines's **SERPs**, like Google's.

Because the exact rules checked by search engines are not entirely public, and competition for popular **queries** is so fierce, SEO is often referred to as black magic... and well, sometimes it does feel like it is! But there are very clear best practices to apply to be respected and taken seriously by Google and its friends.

Of course, making your markup as semantic as you can will help the search engine understand your webpage; but by now, most websites do that right, so it is definitely not going to be enough.

More current strategies involve the content you put in your website, the URLs, the internal and/or external linking, the website's performance, ... The list of things that can be done to improve one's SEO is endless!

To monitor SEO results, people usually use monitoring tools, which are going to keep track of search engine rankings continuously for them in various countries. Most of those tools are paid, but you'll find a few free ones with the most limited features. Many of those tools also meant to help define the right SEO strategy in the first place, when you try to figure out which queries you should be optimizing for (a step called **keyword research**). For instance, if the tool tells you that a query doesn't involve many competitors fighting for it, and yet Google Trends tells you it is a query that is heavily being ran by Google users, then you might be sitting on a goldmine!



Another important note to keep in mind is that a SEO strategy will of course contain the **targeted queries** of someone actively looking for you (for instance, Safeway probably works hard to be ranked high on the “safeway” query), but also the **untargeted queries** of someone who is not specifically looking for you, but whose attention you seek (for instance, Safeway probably works even harder to be ranked high on the “san francisco grocery store” query, or one even more competitive: just “grocery store”, if it’s relevant for them). Deciding which queries make sense to pursue is a critical component of SEO strategy.

Accessibility

Stéphane Deschamps is a mentor and a hardcore internationally recognized web accessibility expert. He once told me the story of how he starts his accessibility training sessions.

First thing: he’d start by asking everyone to stand up. Then he’d ask people wearing glasses to sit down; then people who had back pain sometimes; then people who already had had a broken arm; etc. At the end of his list, he’d tell the people still standing that they could ignore the next few hours, because they will be about all of the other people in the room. The catch is that by then, there would only be at most one person standing, most often none. He now had the room’s attention.

In real life, it is estimated that a website may lose as high as 15% of its visitors to bad accessibility. Think: someone with poor eyesight who couldn’t read the menu button properly because of their low contrast, and gave up; or someone who worked hard with their hands today, and can’t handle the mouse anymore to fill in a long form, and the keyboard navigation won’t work, etc.

Having a semantic and valid HTML markup will already go a long way to make one’s website accessible; but it is definitely not enough, and issues are hidden in the details (images, colors, ...).

Also, for complex applications for which you feel like some bits of the page should be more explicit in their purpose, you can give that extra information with **WAI-ARIA** (often referred to as just “ARIA”).

The specific case of network fault-tolerance

Another kind of loss of accessibility is when resources (such as CSS files, images, JavaScript files) are inaccessible, such as for network reasons. Without CSS, without JavaScript, without images, your website should still work. It may not have as much styling and interactivity, but people must still be able to post that micro-blogging status, for instance.

Browsers often have built-in tools (or plugins) that allow you to see your website without images, without CSS, without JS, ...

Front-end performance

Did you know that over 90% of the time spent by a user waiting for your webpage is happening in the front-end, rather than the back-end? The reasons are pretty obvious when you think about it: you can host your code on faster servers, or scale it on a higher number of them; but there’s nothing you can do to make the user’s

browser environment better. To ensure all browsers can interpret it, you have to send your user code that respects web standards (which are sometimes verbose), and you can't make that code smaller than the standards allow. It has to go through the network, be entirely loaded and run by the browser, you can't have the browser "preload" anything. Well, actually...

There are many pitfalls and tricks to address web performance. All ways attempt to either make each file smaller, or to lower the number of files (it is magnitudes more expensive to download two files of 50kB each, than one file of 100kB).

Some pitfalls:

- Forgetting to enable GZip and set Cache expirations on your web server's configuration
- Using many external JS libraries, including some you absolutely didn't need
- Using huge images displayed smaller
- Leaving the metadata in your images
- Etc...

Some tricks:

- Images sprites
- No CSS or JS inside the HTML (.js and .css files will be cached on first call, won't have to be downloaded next time).
- Ajax calls that can be done with GET HTTP verb should use it (that way, they are cached)
- CSS and JS files are concatenated into one single file each
- CSS and JS files are "minified"
- Images are recompressed

As usual with quality, you will want to measure. But there are many ways to measure a front-end's load time. Should it be the **Time To First Byte**? **Page Load Time**? **Time To First Paint**? You should define your strategy, and monitor the measures you feel are relevant.

Typically, the way measures are done is:

- The cache is cleared, the page is loaded, and the measurements are noted. The operation is done 3 times. If one is very significantly apart to the others, it is removed, to account for potential network issues unrelated to the website. The final result is the average of remaining calls.
- The page is loaded without clearing the cache, and the measurements noted. Again, 3 times, removing very significantly wrong measures, and the final result is the average. How to measure, you say? Just use the performance measuring tools that are likely to be in your browser.

This is Google Chrome's diagnostic of the stackoverflow.com website, with cache cleared. As you can see, the back-end performance of generating the HTML page is 180ms (actually way less, if you remove the round-trip the data is doing) out of 1.30s to finish. Load time is 1.11s, which is not too bad!

