



LATE 2022 EDITION

GUIDE FOR
**SUSTAINABLE
& ACCESSIBLE**
--- WEBS



It is time to bring down the wall that prevents us from seeing how things really are on the Internet. Words like “cloud”, “digital” and “code” makes us think that websites and applications are made of an intangible material, some sort of ether. Things might be virtual, like a meeting, but they require energy to take place. Electricity, for example, to make our machines work, or servers (that also function with electricity) to hold the necessary information to make the magic happen.

Not knowing all these things in 2022 is a luxury that we, the advertising industry, can no longer have. That’s why we’ve created this guide. **Because the clock is ticking and we must take a step forward.**

But that’s not all. This is not just a revolutionary act. And it is not about morality either. This way of doing things according to what we know for sure, is a tendency. It is the path that many brands are taking. So, if we are smart enough, we’ll be leading the change.

Fellows, welcome to this guide! Let’s do things right once and for all.



WHY

Digital Responsibility > Build more conscious brands and consumers

The digital world also has its impact on the environment.

We must change our perspective to focus on efficiency if we want to design a sustainable internet.

+ **DATA** = + **ENERGY** =
+ **EMISSIONS**



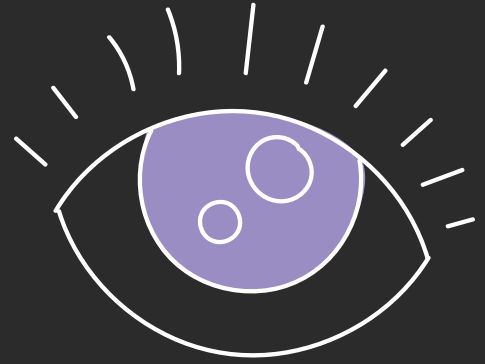
HOW & WHAT

Lighter and more accessible websites that use renewable energy.

A more sustainable alternative with positive impact.

A commitment to and for the world: Minimizing the digital carbon footprint.

HOW TO READ THIS GUIDE



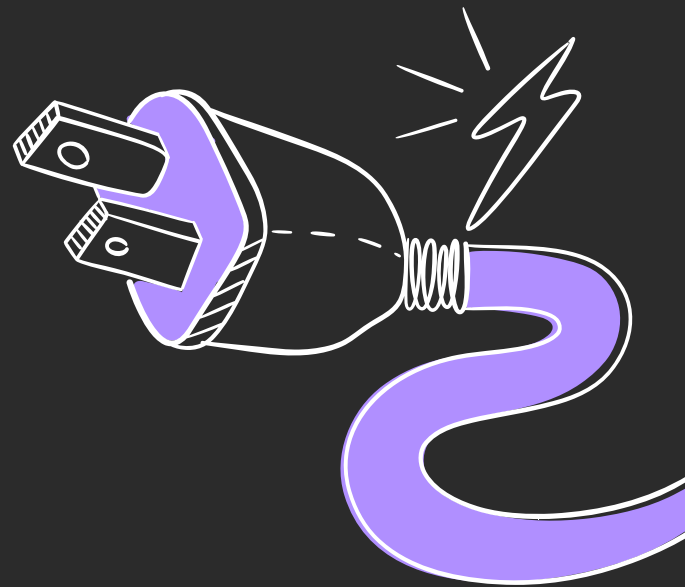
- **As a Guide, not a Rulebook:**
Apply the recommendations as much as possible.
- **Self-assessment:**
Keep track of the applied strategies.
- **Improvement opportunities:**
What could not be applied in one project, maybe in the next one it will be possible!

CONTEXT

Data centers are one of the fastest growing sources of electricity demand globally.¹

Data centers contain millions of servers running 24/7 while also being constantly cooled, therefore consuming enormous amounts of energy.

The formula is simple: the more data that is transferred, the higher the energy consumption and, with most energy coming from fossil fuels, the higher CO2 emissions. If we want to tackle the root problem, we must focus on developing light and sustainable webs that don't consume countless bytes when loading.



1. Tom Greenwood (Feb. 2021) Sustainable Web Design (Book)

WAYS TO MINIMIZE THE DIGITAL CARBON FOOTPRINT OF A WEBSITE & CREATE A MORE ACCESSIBLE INTERNET

#Note | Our designs don't have to be simple and rigid but we must justify the existence of every detail.

→ The decisions we make will be justified under these 4 key points:

1. HOSTING
2. DESIGN EFFICIENT SITES
3. DEVELOP EFFICIENT SITES
4. GUARANTEE ACCESSIBILITY

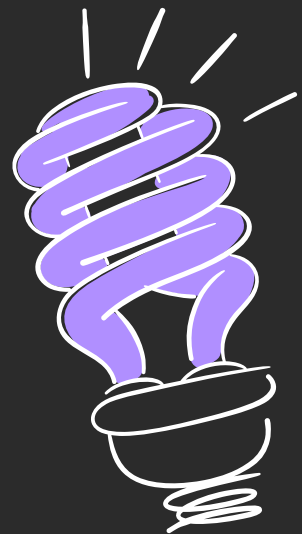
1. HOSTING

Use a hosting powered by renewable energy, or that offsets energy consumption from fossil fuels by purchasing renewable energy credits, including their offices and their data centers.

HOW

[Green Web Foundation](#) provides a list of hosting services that meet this requirement. In the case of Planta, we use GreenGeeks Web Hosting.

Select a data center as close as possible to the main user base. For websites that have an international user base, using a content delivery network (CDN) helps reduce the distance that large files travel across networks to reach each user, and therefore reduces power consumption.



2. DESIGN EFFICIENT SITES

Reduce the space occupied on the server and lower the requests to it.

Consider if the images, videos, icons and animations you have planned for your website add real value. In general, the fewer pages there are to load, the fewer data requests are sent.

HOW



→ Design with a UX/UI vision:

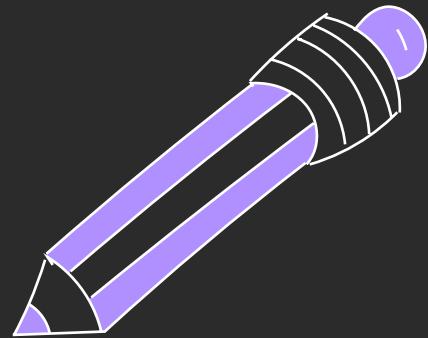
Reduce unnecessary page loads, the number of words on each page, and the number of pages on each site. Schedule a short user journey (flowchart), for example by providing all specific subcategories in a dropdown menu and providing a nice search box. Identify the main objectives of the users and make sure that they can fulfill them with the least amount of clicks as possible.

→ Image optimization:

Use image compression tools like [Shortpixel](#), [TinyPNG](#) and [ImageOptim](#) that minimize file size without affecting their capacity to make a strong visual impression.

→ Vector images:

Use SVG y CSS files if possible.



→ Color:

Most modern smartphones and a growing number of devices such as laptops and smart TVs use OLED screens that light each pixel individually. Since **each pixel is a small LED**, using darker colors is a viable technique to reduce energy for the end user. Black is the most efficient color in OLED displays since the pixel is off, white is the most energy-intensive color, and **darker colors generally use less energy**.

→ Blurring images:

Could be nearly halved in size by blurring the foreground, with almost no noticeable difference to the viewer.

→ Animation & Videos:

Ask yourself the end-purpose of the video to justify its use compared to other alternatives.

Avoid auto-play, such as in background videos. This also benefits those with sensory issues and movement sensitivity.

Use videos hosted on carbon neutral platforms such as Youtube.

→ Typographies:

The most efficient option is to use system local fonts. I.e., fonts pre-installed on the devices (Google fonts or Apple/Android system fonts), as **they take zero server requests and zero data transfer to use.**

→ Page reduction:

If you design the site so the users can access the information they need by only visiting two pages instead of three, for example, you'd significantly reduce the carbon footprint of their visit, as well as create a better experience.

Avoid the yo-yo effect, caused when a user has to always go back, for example, to the home page in order to access other pages, resulting in unnecessary navigation and wasting time.

If the design avoids the user getting lost, we can reduce these back-and-forth travels around the site.

3. DEVELOP EFFICIENT SITES

Writing fewer lines of code makes the files lighter, but also reduces the server's amount of processing on each page load or visit.

HOW



→ Efficient languages:

Researchers found that the three least energy intensive programming languages were C, Rust, and C++. Popular web programming languages JavaScript and PHP ranked 17th and 21st respectively. However, JavaScript and PHP were two of the most energy efficient languages when manipulating strings with regular expression.

Although JavaScript might be more energy efficient in processing algorithmic problems, it will always be less efficient than serving static files in cases where we don't need our code to "solve problems." The most common example is the use of JavaScript for animating elements in a web design. CSS can now achieve many animation effects with far more efficiency than JavaScript because it minimizes the need for the CPU to "think"—and can generally be achieved with far smaller file sizes, minimizing the energy used to transfer data.

→ We should think carefully about

whether the code we use is necessary:

Is it correct to use a front-end framework like React, Vue, or Angular for websites that don't have any requirement for them? Likewise, the functionality that JavaScript adds tends to consume energy and slow websites down, without adding value for the user.

Do we gain anything else besides time by using these tools? Does this supposedly gained time have value compared to the unnecessary data that these frameworks accumulate in our web, making it much more energy consuming? We strongly advise you to take the time to write your own code to make sure that each character has a real value within the total code.

→ If you can, choose and use HTML:

With the introduction of content management systems (CMSes), the actual HTML files no longer exist on the web server; the server has to generate these files dynamically every time someone requests a web page. Despite the incredible power CMSes have given people without technical knowledge of web development to take control of websites' content, the technology enabling this control is a lot less efficient, resulting in higher energy consumption and slower load times. However, sometimes we just need to use CMSes such as WordPress and that is OK. Fortunately, **there are techniques that can significantly reduce the server processing of requests, such as Cache.**

→ Cache:

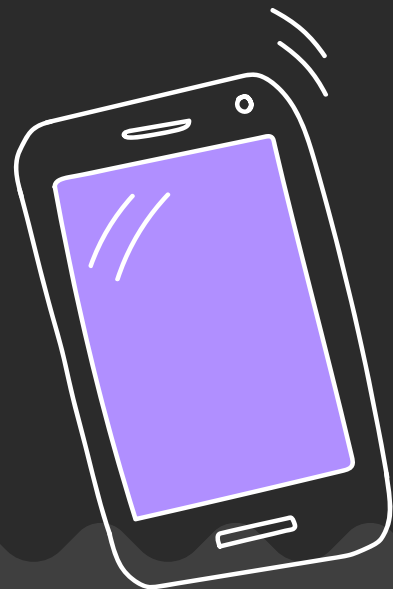
Use cache memory systems to generate static versions of the website and **prevent the server from having to load it on each visit**, thus reducing server energy consumption and improving page load times.

→ SEO:

Improve the positioning of our website in search engines so that users **find the information they are looking for more quickly and easily**. In this way they spend less time browsing and less energy is consumed.

→ Mobile First:

With mobile first, **content is optimized for smartphone use from the start**. Optimized pages don't have to load large media on mobile devices. They are programmed in such a way that the desktop styles do not have to be overwritten, but are designed directly for the smaller device.



→ Animations:

In case the design includes animations, the development team should determine which is the best approach to implement them. **Whereas GIFs and videos might be the fastest approach for development**, they are separate files that need to be stored in the server and later sent to every user. Should it be possible, JavaScript and CSS animations are preferable as they use the browser's internal engine to generate the animations. It is also worth mentioning that CSS can now achieve many animation effects with far more efficiency than JavaScript and that it does so with far smaller file sizes, **minimizing the energy used to transfer data.**

→ Images:

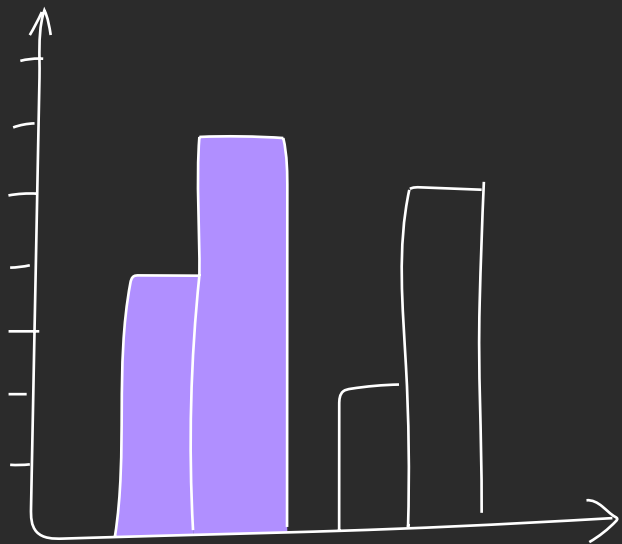
Instead of uploading several images in the exact sizes they will be displayed, generate different versions to adapt to different layouts. We can use `srcset` and `sizes` attributes in responsive markup to help browsers know which images load. **The simplest approach is to estimate the sizes for the most common display users will need.** We can upload images at scale instead of relying on CSS to resize them. If you're using WordPress, this is not as important as it supports responsive images.

→ Lazy Loading:

Use Lazy Loading so **images are loaded only if and when they are needed.**

→ Page size:

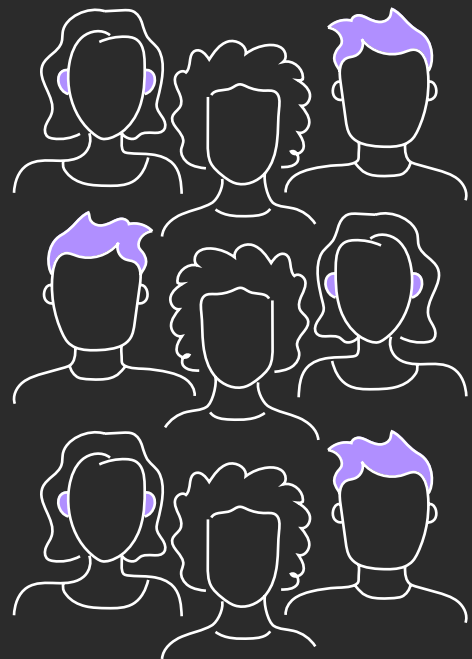
Set yourself the challenge of making a website at least 20% lighter than your competitors'. Use a website size and speed tester to make a benchmark and start designing with a maximum weight goal.



4. GUARANTEE ACCESSIBILITY

When websites and web tools are properly designed and coded, **people with disabilities can use them.** However, currently many sites and tools are developed with accessibility barriers that make them difficult or impossible for some people to use.

International web standards have already defined what is necessary to guarantee that a website's content reaches all audiences.



This is not a “nice to have” feature to be added eventually. **It is a requirement that has to be met in the first version publicly available** and it is one of the metrics used by Search Engines to measure the quality of a website.

HOW



→ Font size control:

Or use of large and readable fonts. For small fonts, be careful about using italics, fine, and serif fonts.

→ Easy reading:

Avoid justified, centered, and right-aligned text for longer paragraphs.

→ Meaningful Link Text:

It is important that each link has meaningful text describing its purpose without relying on the surrounding text. **Screen readers already inform users that they have encountered a link so there is no need to include language such as “link to” or “click the link” in link text.**

Instead of:
Click Here to see today's weather.

Use:
Today's weather



→ Check Contrast:

Between texts and their background.

→ Sensory Clues:

Do not rely solely on sensory characteristics such as color, shape, size, visual location, orientation, or sound when providing instructions for understanding or interacting with content.

→ Alt-Text:

Alternative text provides a textual description of images for those who may not be able to see them. Screen reading software will read the alt text aloud.

→ Videos:

Multimedia files with a transcript, captions and sign language interpreters.

Visit [WAVE](#) for more information on Web Accessibility

USEFUL TOOLS

Green Hosting

thegreenwebfoundation.org

Size and Speed Testers

tools.pingdom.com/#60386c9c39c00000

gtmetrix.com

pagespeed.web.dev

Website carbon footprint tester

websitecarbon.com/website

ecograder.com

greenframe.io

Mobile energy consumption

mobile-efficiency-index.com

Accessibility

wave.webaim.org/

webaim.org/resources/contrastchecker/

myusf.usfca.edu/digital-accessibility

colors.co/contrast-checker/

Tools libraries

sustainablewebdesign.org/strategies/

ecometer.org/rules/

SOURCES

Tom Greenwood (Feb. 2021)

[Sustainable Web Design \(Book\)](#)

[Greening the Web: How We Can Create Zero Carbon Websites](#)

[Dark Mode Explained](#)

[Green Software Foundation](#)

[Resources from the Green Software Foundation](#)

[How to set a Page Weight Budget](#)

[Example of Sustainable Web Design: Low Impact Website - Organic Basics](#)



www.weareplanta.com