Engineer

Website

related web pages published together for a common purpose

What it is

A website is a group of related web pages published together for a common purpose by an owner or group: a company's pages, a shop, a set of documentation, a personal blog. The pages often share a domain or a part of one, a navigation, a look and somebody who answers for them. "Site" is the everyday short form, and "web site", as two words, is the older spelling.

The useful line runs between the site and the page. WCAG defines a web page as a non-embedded resource fetched from a single URI over HTTP, together with everything a user agent renders with it, such as images, style sheets and scripts. Its word for a website is a *set of web pages*: pages that share a common purpose and come from the same author, group or organisation. That second definition is the more honest of the two, because it ties a site to purpose and authorship rather than to a domain name.

The examples attached to the definition show how far the idea reaches. A shop's checkout, stripped of the navigation the product pages share, is no part of the set those pages form. A blog on a subdomain with its own navigation and its own authors is a separate set from the pages on the main domain. And different language versions of the same pages count as different sets. Three domains can therefore hold one website, and one domain can hold several.

Three neighbours get confused with a website. A *web server* is the program that answers requests; one server can hold dozens of sites, and one large site can be spread across many servers. A *domain* is the name in the address, which a site may share with others or spread across several. A *web application* is a site that provides software through the web, with work done in the browser, on the

server or both; WCAG's own example is a webmail program that lives at a single address yet shows an inbox, contacts and a calendar, which makes one URI feel like many pages.

Sites also differ in when their pages are made. A *static* site is generated ahead of time into files that a server hands out unchanged. A *dynamic* site builds each page when it is asked for, usually from a database and a program running on the server. A reader can rarely tell which is which from the page itself. The difference shows up in what has to keep running.

Use the concept in a decision

Put "What is a website?" to work in a real decision. We can help relate the concept to the system you are building or buying. A site that is quick, findable and editable by your own staff, without a bill for every change of wording.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Developed 100 web applications using HTML/HTML5 — 100 web applications built

Developed 100 web applications using HTML/HTML5 Get in touch →

Why it matters

People judge sites, not pages. A reader who lands on one article from a search result decides whether to trust it from what the menu, the neighbouring pages and the contact details say. A strong page on a site that looks abandoned inherits the abandonment.

Search engines read the site as a whole. They index pages one at a time but learn about them through the site: its links, its structure and the list of addresses it publishes. The sitemaps protocol, the format a site uses to hand that list to crawlers, allows up to 50,000 addresses and 50MB uncompressed per file, and a larger site splits its list across several files gathered in an index.

Some accessibility requirements only exist across pages. Three WCAG 2.2 success criteria at level AA are written about a set of web pages. Multiple Ways (2.4.5) asks for more than one route to each page, except a page that is a result of, or step in, a process, such as a checkout. Routes can be a menu and a search, or a table of contents and links between related pages. Consistent Navigation (3.2.3) asks that navigation repeated on several pages keep the same relative

order each time, unless the user initiates a change. Consistent Identification (3.2.4) asks that components with the same job be identified consistently. Every page can pass a check on its own while the site fails all three.

Addresses are a promise. Each published address ends up in bookmarks, other people's links, search indexes, emails and printed material. A site's list of addresses is the part outsiders build on, and it outlives any design.

Something has to keep running. A dynamic site is a program with a database behind it, to be patched, watched and paid for every month. A static site is a set of files and a server that hands them out. That choice sets the attack surface and the running bill far more than the visual design does.

What it looks like in practice

This website is a working example of the definitions above. Built as a multilingual static site in Hugo, at the stage documented in August 2026 it had 108 template files, 53 of them partials and 18 shortcodes, generating three languages from one content tree with no runtime backend. Every page went out in four representations: HTML, a Markdown twin, a Gemini document and a Gopher menu entry. The same tree was built seven more times, once per focused subdomain, by pointing the generator at a different content directory. Counted by addresses, that was eight sites; counted the way WCAG counts, with each language a separate set of web pages, the three languages alone were three. The page is plain about the price of having no runtime backend: apart from one scoped JavaScript enhancement for the portfolio filter, interaction was solved at build time or left out, and a search index was costed and parked for that reason.

The earlier work shows the other two halves of the job: running sites and building them from a design. On the hosting side, forty websites on Ubuntu Linux hosting servers ran on a mix of Apache and Nginx and were kept available and fast through configuration, performance tuning and maintenance. The measure there was behaviour under real traffic, since a site that only holds up on a quiet night has been left alone rather than looked after.

On the building side, thirty websites designed from Photoshop files were converted to HTML by hand and held to two standards at once: faithful to the spacing, type and details the designer intended, and still maintainable once the content started to change. A site that matches its mockup exactly and breaks at the first edit was built for the launch, not for the years after it.

Where it goes wrong

- **A redesign changes the addresses.** Every inbound link, bookmark and search result that pointed at the old structure lands on an error page the day the new site goes live. A permanent redirect from each old address to its new home, kept for years, preserves what other people built on.
- **Navigation drifts from section to section.** Templates written at different times grow their own menus in different orders, and the same page is called “Contact” in one place and “Get in touch” in another. Readers lose their bearings and the site fails Consistent Navigation and Consistent Identification while every page passes alone. One shared navigation template, used everywhere, removes the drift at its source.
- **The site is treated as finished at launch.** Content goes stale, dependencies age, certificates expire and the one person who knew how to publish moves on. A site lasts when it has an owner, a publishing routine and checks that run whether or not anyone remembers them.
- **A page can be reached by one route only.** A page buried three menus deep and left out of every index is found by its author and by very few others. A second route, whether a search, a site map or links from related pages, is what Multiple Ways asks for on pages outside a process, and what readers reach for anyway.
- **A second language is bolted on after launch.** Addresses, dates, number formats, string lengths and menus were all shaped around one language, and the new one has to fight every template. A site meant to be multilingual is cheaper to build that way from its first page.

Further reading

- NIST CSRC Glossary: website — a one-line definition from NISTIR 7693: related web pages prepared and maintained as a collection in support of a single purpose.
- WCAG 2.2: set of web pages — the standard’s definition of a site by purpose and author, with examples covering checkouts, subdomains and language versions.
- Understanding SC 3.2.3: Consistent Navigation — why navigation repeated across pages should keep its order, and the failure that breaks it.
- Understanding SC 2.4.5: Multiple Ways — the routes a site can offer to each of its pages, and the exception for steps in a process.
- sitemaps.org: Sitemaps XML format — the format a site uses to list its own addresses for search engines, with its size limits per file.
- W3C: Mobile Web Application Best Practices, Web Application — the Recommendation’s scope includes applications using server-side or client-side processing, rather than requiring browser-only execution.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is a website?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/website/