**Engineer**

# Landmark

a region of a page a screen reader can jump to

## What it is

A landmark, also called a landmark region or an ARIA landmark, is a part of a web page that the markup names by its purpose, such as the site header, the main content, the navigation or the footer, so that assistive technology can list those parts and move straight to any one of them. A page usually has several landmarks, and together they describe its layout to someone who never sees the layout.

WAI-ARIA 1.2 defines eight landmark roles: `banner`, `navigation`, `main`, `complementary`, `contentinfo`, `search`, `form` and `region`. A ninth, `landmark` itself, is the abstract parent in the specification's role model, and authors are told not to use it. Most pages never need the `role` attribute to get the others, because HTML elements carry them already:

- `main` is the `main` landmark, and `nav` is a `navigation` landmark.
- `header` and `footer` become `banner` and `contentinfo` only when they belong to the page as a whole. Placed inside an `article`, a `section` or `main`, they are ordinary page structure and not landmarks at all.
- `aside` is `complementary` at the top level of the page or inside `main`. Inside other sectioning content it counts only when it has a name.
- `section` becomes a `region` landmark only when it has an accessible name, from `aria-label` or from a heading referenced with `aria-labelledby`. A nameless `section` is just a group.
- `form` maps to the `form` role, and the specification requires authors to label it. The newer `search` element maps to `search`.

The specification asks assistive technologies to let people reach these regions quickly, and screen readers do. In NVDA, `D` moves to the next landmark and the elements list shows all of them at once; VoiceOver's rotor has a landmarks list of

its own. A user arriving on a page can open that list, hear that there is a banner, a navigation, a main region and a footer, and know the page's shape before reading a word of it.

Use the concept in a decision

Put "What is a landmark?" to work in a real decision. We can help relate the concept to the system you are building or buying. Interfaces that load quickly, work on a phone and behave the same in every browser your customers happen to use.

**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. Directed full-stack GIS map development — full-stack mapping, from database to browser

**Directed full-stack GIS map development** Get in touch →

## Why it matters

The decision behind landmarks is whether every part of a page sits somewhere a listener can name and reach. On a typical page most of the content is repeated: the header, the menu, the footer and often a sidebar are the same on every page of the site, and the part that is new sits somewhere in the middle. A sighted visitor's eyes go straight there. Without landmarks, a screen reader user has to listen past the same header and menu on every page, or guess how many headings to skip.

WCAG's own technique for landmarks, ARIA11, is listed as a sufficient way to meet three success criteria. One of them asks for a way to skip blocks of content repeated across pages; another asks that the structure a sighted reader takes from the layout is also available to software. The cost of meeting them this way is small when the page is built from `header`, `nav`, `main` and `footer` in the first place: the semantics come with the element, and there is nothing extra to maintain. Retrofitting landmarks into a layout made of nested `div` elements costs more, but the work is mostly mechanical.

There are two limits worth knowing. The first is that the specification leaves landmark navigation optional for mainstream browsers, so a keyboard user without a screen reader usually gets nothing from landmarks on their own, and a visible skip link still earns its place. The second is that more landmarks do not make a page easier to use. Each one is an entry in a list somebody has to hear, and a page where every block is a named region reads like a table of contents

fifty lines long. The specification keeps `region` for content important enough to appear in a summary of the page, and asks authors to use it only when none of the other roles describes the purpose.

## What it looks like in practice

The brand mark on every inner page sat outside every landmark, and nothing on screen gave that away, which is typical of a defect only the structure can show. It came to light during the work to take the whole site to WCAG 2.2 level AA. The automation runs an accessibility rule engine with its best-practice rules enabled, not only the conformance ones, which is a thirty-rule difference, and one of the extra rules failed the moment it was switched on.

The guards explain how it had gone unnoticed. Among eleven defects the browser had called healthy, two were in the guards rather than the pages, and one of those was that the rule engine had been running seventy of its hundred and five rules. The rule that would have caught the missing landmark was simply not in the set. The lesson is short: a rule set that does not contain the rule cannot enforce it.

Outside the main site, the landing page in front of the company's own git forge had been a hand-maintained copy of the main site's design, and every difference ever found between the two turned out to be an accident rather than a decision. One of the five was that no page had a main landmark or a top-level heading, and not one of the five was visible in a screenshot. The copy was deleted, and the landing page is now built by the main site's own templates, so it inherits the same structure instead of imitating it.

## Where it goes wrong

Landmark mistakes are silent on screen and plain to hear in a screen reader. These are the common ones.

**Content lands outside every landmark.** A logo, a cookie notice or a language switcher is placed before the `header` or after the `footer`, and a listener moving by landmark skips past it. The axe rule `region` reports any content not contained by a landmark, but it is tagged as best practice rather than as a WCAG failure, so a run limited to the WCAG tags never executes it. Turn the best-practice rules on, and move the stray element inside the landmark it belongs to.

**The same role appears twice with the same name.** Two `nav` elements, one in the header and one in the footer, both announce as “navigation”, and the listener has to guess which is which. Give each a short, distinct label with `aria-label` or `aria-labelledby`, and leave the role out of the label: a navigation labelled “Site navigation” is read as “Site navigation navigation”.

**A second main or banner appears.** A `div` is given `role="main"` or `role="banner"` to make a component feel important, or a template wraps an inner page in a `main` of its own, and the page now announces two of a role it should have once. The specification asks for at most one `banner`, one `main` and one `contentinfo` per page, and axe has a rule for each. Check the landmark list after every layout change, and let the native elements carry the roles wherever they can.

**Every block becomes a region.** A component library gives every `section` an `aria-label` by default, and the landmark list fills up with card titles. Name a `section` only when a reader would want to jump to it from anywhere on the page.

A page shaped like the sketch below gives a listener five stops, each one distinct:

```
<header><!-- logo, search --></header>
<nav aria-label="Site"><!-- menu --></nav>
<main>
    <h1>Prices</h1>
    <section aria-labelledby="hosting">
        <h2 id="hosting">Hosting</h2>
    </section>
</main>
<footer><!-- contact, legal --></footer>
```

## Further reading

- WAI-ARIA 1.2: Landmark Roles — the eight roles, what each one is for, and which ones must carry a label.
- Landmark Regions — the W3C authoring guide on choosing roles, labelling repeated landmarks and keeping all content inside one.
- HTML Accessibility API Mappings 1.0 — which HTML elements become which roles, including when `header`, `footer`, `aside` and `section` count as landmarks.

- ARIA11: Using ARIA landmarks to identify regions of a page — the WCAG technique, with its test procedure and the success criteria it meets.

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 landmark?” 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/landmark/