

Jak se specifikace ARIA rozrůstala, jak zapadá do HTML a co je nového

Od prvního návrhu prošla ARIA čtyřmi hlavními verzemi a dnes je nedílnou součástí toho, jak prohlížeče popisují obsah stránky čtečkám obrazovky a dalším asistivním technologiím. Ve druhém díle seriálu o WAI-ARIA se podíváme, jak se specifikace postupně rozrůstala, jak zapadá do běžného HTML a co se v ní změnilo za poslední rok.

Jak se specifikace rozrůstala

- roadmap a první návrh během roku 2006
- verze 1.0 v roce 2014
- verze 1.1 v roce 2017
- verze 1.2 v roce 2023
- verze 1.3 stále rozpracovaná, zatím jen pracovní návrh (v roce 2026)

Každá verze přidávala hlavně to, co se ukázalo jako chybějící při reálném používání. 1.1 třeba přinesla `aria-current` (označení aktuální položky v navigaci nebo aktuálního kroku v rámci postupu) a rozšířila `aria-haspopup` z prostého ano/ne na konkrétní typ vyskakovacího obsahu (menu, dialog, strom...). 1.2 doplnila role, které chyběly pro plné pokrytí HTML (`blockquote`, `caption`, `code`, `deletion`, `emphasis`, `generic`, `insertion`, `meter`, `paragraph`, `strong`, `subscript`, `superscript`, `time`).

Také výrazně přepracovala vzor comboboxu a naopak u čtveřice dřív univerzálně použitelných atributů (`aria-disabled`, `aria-errormessage`, `aria-haspopup`, `aria-invalid`) zúžila použití jen na konkrétní widgety, místo aby šly nasadit kdekoliv.

Jak to zapadá do HTML

ARIA sama o sobě nemění, jak stránka vypadá, ani jak se chová. Nemění tedy ovládání klávesnicí, fokus, nic vizuálního. Jen doplňuje nebo přepisuje informace ve stromu přístupnosti, který prohlížeč staví z DOM a předává dál operačnímu systému a čtečce obrazovky.

Nativní HTML prvky mají svoji roli zabudovanou už samy, například `<button>` je `role="button"` a `<nav>` je `role="navigation"`, aniž by to bylo potřeba psát ručně. Tohle mapování formálně definuje specifikace „ARIA in HTML“ a přímo doporučuje: pokud existuje nativní prvek s rolí, kterou chceš, nepiš ji znovu jako atribut.

To je zároveň první a nejdůležitější z pravidel používání ARIA, jak je definuje W3C:

1. Pokud jde použít nativní HTML prvek nebo atribut se sémantikou a chováním, co potřebuješ, použij ho, místo aby ses snažil totéž dodělat pomocí role na obecném prvku.
2. Neměň nativní sémantiku prvku, pokud fakt nemusíš (nadpis, který má zároveň fungovat jako záložka: obal ho do `<div role="tab">` a roli nepiš přímo na nadpis).
3. Každý interaktivní prvek s ARIA rolí musí jít ovládat klávesnicí.
4. `role="presentation"` ani `aria-hidden="true"`

nikdy nepatří na prvek, který může dostat fokus.

JanBalak.name — <https://www.janbalak.name/clanky/jak-se-specifikace-aria-rozrustala-jak-zapada-do-html-a-co-je-zvlast-dulezita> je otázka, jak se skládá „přístupné jméno“ prvku, tedy text, který se čtečka rozhodne přečíst jako jeho název. Existuje pro to přesně daný algoritmus (specifikace Accessible Name and Description Computation) s pevným pořadím priority: `aria-labelledby`, pak `aria-label`, pak nativní labelování (`<label for>`, `alt`, `title`, `<caption>...`), a teprve nakonec viditelný obsah prvku. Tohle pořadí je důvod, proč `aria-label` „přebije“ viditelný text a proč se dá snadno omylem udělat tlačítko, které vidí jinak vidící uživatel a jinak čtečka obrazovky.

Co je nového za poslední rok

Za poslední rok se v ARIA ekosystému stihlo odehrát hned několik věcí:

- Specifikace **ARIA in HTML**, která popisuje, jak se role a atributy ARIA doplňují s významem, který už mají nativní HTML prvky samy o sobě, dostala 11. srpna 2026 novou verzi doporučení, která nahrazuje tu z prosince 2021. Nová verze mimo jiné upřesňuje, jak se má chovat prvek `summary`. Řeší taky situaci, kdy `<label>` není spárovaný s žádným formulářovým prvkem, takže podpora ARIA je na něm proto pouze podmíněná. A doplňuje sémantiku pro nově přichází, uživatelsky přizpůsobitelný `<select>`.
- **Core-AAM 1.2** je specifikace, která přesně určuje, jak se jednotlivé role a atributy ARIA namapují na konkrétní platformní rozhraní pro přístupnost, třeba na Windows nebo macOS. Na jaře 2026 postoupila do fáze Candidate Recommendation Draft, tedy o krok blíž k finálnímu schválení.
- Výroční audit milionu domovských stránek **WebAIM Million 2026** ukázal, že ARIA teď používá 82,7 % zkoumaných stránek, o něco víc než loňských 79,4 %. Zajímavější než samotné rozšíření je ale to, že stránky, které ARIA používají, mají v průměru **59 chyb přístupnosti**, zatímco stránky bez ní jen **42**. Průměrný počet ARIA atributů na stránku navíc od roku 2019, kdy jich bylo 22, vzrostl na víc než 130. Tahle čísla potvrzují často opakované pravidlo „žádná ARIA je lepší než špatná ARIA“: víc ARIA na stránce samo o sobě neznamená přístupnější web, a pokud se používá bez správného pochopení, může naopak škodit.
- U moderních komponent, takzvaných Web Components, přibývá nativní podpora rozhraní `ElementInternals` a `ARIAMixin`. Díky nim může vlastní (custom) element deklarovat svoji ARIA sémantiku přímo v JavaScriptu, místo aby ji vývojář musel ručně dopisovat jako atributy do značky. Podpora v prohlížečích Chrome, Firefox i Safari je už solidní. Pořád se ale řeší, jak spolehlivě propojit ARIA vztahy mezi prvky, například `aria-labelledby`, který odkazuje na popisek jinde v dokumentu, v situaci, kdy leží každý kus za hranicí takzvaného shadow DOM, tedy izolované části stránky, kterou si komponenta spravuje sama.
- Samotná **ARIA 1.3** je zatím jen pracovní návrh, ne finální doporučení. Přináší mimo jiné dvě nové role, `sectionheader` a `sectionfooter`. `<header>/<footer>` uvnitř `<article>` nebo `<section>` doteď žádnou konkrétní roli nemělo, nově je čtečka obrazovky rozpozná jako záhlaví/zápatí té konkrétní sekce. Druhá změna je zákaz `aria-hidden="true"` přímo na kořenovém elementu `<html>`, aby nešlo omylem schovat celou stránku před asistivní technologií najednou. K oběma změnám (a k dalším atributům a rolím obecně) se podrobně věnuje pozdější díl o novinkách z 1.3. Nic z toho navíc zatím není definitivní. Co všechno se z pracovního návrhu nakonec dostane do finální verze, se ještě může změnit.

Zdroje

- [W3C WAI-ARIA 1.2 \(aktuální doporučení\)](https://www.w3.org/TR/wai-aria-1.2/) (<https://www.w3.org/TR/wai-aria-1.2/>)
- [W3C WAI-ARIA 1.3 \(aktuální pracovní návrh\)](https://www.w3.org/TR/wai-aria-1.3/) (<https://www.w3.org/TR/wai-aria-1.3/>)
- [W3C: ARIA in HTML - mapování rolí](https://www.w3.org/TR/html-aria/) (<https://www.w3.org/TR/html-aria/>)
- [W3C: ARIA in HTML - aktualizace ze srpna 2026 \(novinka W3C\)](https://www.w3.org/news/2026/updated-w3c-recommendation-aria-in-html/) (<https://www.w3.org/news/2026/updated-w3c-recommendation-aria-in-html/>)
- [W3C: Using ARIA - pravidla použití](https://www.w3.org/TR/using-aria/) (<https://www.w3.org/TR/using-aria/>)

- [W3C: Accessible Name and Description Computation — výpočet přístupného jména](https://www.w3.org/TR/accname-1.2/)
(<https://www.w3.org/TR/accname-1.2/>)
- [JanBalak.name — https://www.janbalak.name/clanky/jak-se-specifikace-aria-rozrusala-jak-zapada-do-html-a-co-je-noveho.html](https://www.janbalak.name/clanky/jak-se-specifikace-aria-rozrusala-jak-zapada-do-html-a-co-je-noveho.html)
- [W3C: Core-AAM 1.2 - status Candidate Recommendation Draft](https://www.w3.org/TR/core-aam-1.2/)
(<https://www.w3.org/TR/core-aam-1.2/>)
- [WebAIM Million 2026 — statistiky použití a chyb](https://webaim.org/projects/million/) (<https://webaim.org/projects/million/>)