

Další ARIA parametry prakticky: aria-describedby, aria-hidden a aria-disabled

Ve čtvrtém díle seriálu o WAI-ARIA navazujeme na předchozí ARIA parametry (*aria-label* a *aria-labelledby*) třemi dalšími, se kterými se v praxi setkáte skoro stejně často: *aria-describedby* pro delší popis, *aria-hidden* pro schování obsahu před čtečkou a *aria-disabled* pro needitovatelný, ale pořád dosažitelný prvek.

aria-describedby

Přístupný popis prvku (delší doplňující informaci, ne krátký název) lze navázat na existující text pomocí *aria-describedby*, který je součástí WAI-ARIA od verze 1.0, platí i v 1.1-1.3.

Co dělá

Doplňuje k prvku popis odkazem na *id* jiného prvku (nebo víc prvků). Na rozdíl od *aria-labelledby*, který dává prvku *jméno*, *aria-describedby* dává prvku *popis*: delší, doplňující text, který čtečka přečte až po jméně a roli:

```
<button aria-describedby="smazat-popis">Smazat</button>
...
<p id="smazat-popis">Položka se přesune do koše a natrvalo smaže po 30 dnech.</p>
```

Čtečka ohlásí „Smazat, tlačítko. Položka se přesune do koše a natrvalo smaže po 30 dnech.“

Na co se aplikuje

Lze použít na libovolné roli a libovolném prvku HTML. Na rozdíl od pojmenovacích atributů tu totiž neexistuje seznam rolí, kde by to bylo zakázané. Typicky formulářová pole s nápovědou nebo chybovou hláškou, tlačítka s netriviálním důsledkem akce, prvky s doplňujícím kontextem:

```
<input type="password" aria-describedby="heslo-pozadavky">
<p id="heslo-pozadavky">Nejmén? 8 znak?, jedno ?íslo a jedno velké písmeno.</p>
```

Více odkazovaných ID

Hodnota může obsahovat víc *id* oddělených mezerou. Stejně jako u *aria-labelledby* se jejich texty spojí v pořadí, v jakém jsou v *aria-describedby* vypsané:

```
<input type="text" aria-describedby="format-napoveda datum-chyba">
<p id="format-napoveda">Formát: DD.MM.RRRR</p>
<span id="datum-chyba">Neplatné datum</span>
```

Omezení

- **Odkazovaný prvek nemusí být viditelný** — může existovat jen kvůli čtečce, typicky skrytý běžnou „sr-only“ technikou a odkrytý třeba tlačítkem „více informací“.
- **Neexistuje-li odkazované id na stránce**, *aria-describedby* se prostě neuplatní: žádná chyba, popis se jen nezobrazí.
- **Není to náhrada za *aria-label/aria-labelledby*** — popis se čte až po jméně a roli, nikdy místo nich, a prvek bez jména a jen s popisem zůstává nepojmenovaný.

Hodnota a vztah k jiným atributům

Hodnota je seznam jednoho nebo víc *id* oddělených mezerou, stejný formát jako *aria-labelledby*, jen s jiným účelem: *jméno* (*aria-labelledby*) vs. *popis* (*aria-describedby*). Existuje i *aria-description* pro přímý textový popis bez potřeby odkazovaného prvku, ale s výrazně nižší podporou u čteček než *aria-describedby*.

Proč to takhle je

- Popis se čte samostatně až po jméně a roli prvku. Proto *aria-describedby*

- nenahrazuje pojmenování, jen ho doplňuje.
- Odkazovaný prvek se použije, i když je sám označený `aria-hidden="true"`. Výpočet přístupného popisu totiž tento atribut na cílovém prvku ignoruje. Naopak `hidden` nebo `display:none` prvek z DOMu skutečně odstraní a odkaz na něj se neuplatní.
 - Kombinace víc `id` umožňuje poskládat popis z kousků textu, které jsou jinak v DOM samostatné (typicky nápověda + chybová hláška vedle sebe).

Příklad správného použití

```
<label for="email">E-mail</label>
<input type="email" id="email" aria-describedby="email-napoveda">
<p id="email-napoveda">Použijeme ho jen pro odpověď na váš dotaz.</p>
```

Čtečka ohlásí „E-mail, textové pole, Použijeme ho jen pro odpověď na váš dotaz.“ Popis existuje jako normální viditelný text vedle pole. `aria-describedby` ho jen navíc naváže na konkrétní pole, aby ho čtečka přečetla spolu s ním.

aria-hidden

Prvek zůstává vidět, ale zmizí z toho, co čtečka vůbec „vidí“. Přesně to dělá `aria-hidden`, který je součástí WAI-ARIA od verze 1.0, platí i v 1.1–1.3.

Co dělá

Odstraní prvek a všechny jeho potomky ze stromu přístupnosti. Čtečka o něm pak neví, i když je na stránce normálně vidět:

```
<span aria-hidden="true">...</span>
```

Ozdobná značka se vizuálně zobrazí, ale čtečka ji přeskočí, jako by tam nebyla.

Na co se aplikuje

Nejčastěji na dekorativní inline SVG ikony vedle textu, který už stejnou informaci nese. `<img>` má na dekorativnost vlastní mechanismus (`alt=""`), ale SVG žádný `alt` nemá, takže se to tam řeší přes `aria-hidden`. Dál se hodí na zdvojený nebo redundantní text, případně na obsah dočasně nedostupný jen vizuálně:

```
<button type="button">
  <svg aria-hidden="true" focusable="false">...</svg>
  Uložit
</button>
```

Ikona je čistě dekorativní, celou informaci nese viditelný text „Uložit“.

Omezení

- **Nikdy se neaplikuje na fokusovatelný prvek ani na jeho rodiče**, protože by ho čtečka přeskočila, ale klávesnicí by se na něj pořád dalo dostat (Tab), což je matoucí past: uživatel se dostane na „neviditelný“ prvek, o kterém čtečka mlčí.
- **Nepoužívat tam, kde je prvek skrytý už jinak**, protože `hidden`, `display:none` nebo `visibility:hidden` prvek ze stromu přístupnosti odstraní samy, `aria-hidden` navíc nic neřeší.
- **`aria-hidden="false"` nepřebije rodiče**, takže je-li nadřazený prvek `aria-hidden="true"`, potomek s `aria-hidden="false"` se stejně neodkryje, protože dědičnost jde odshora dolů.

Hodnota a vztah k jiným atributům

Hodnota je `true` nebo `false`. Bez tohoto atributu se přítomnost prvku ve stromu přístupnosti řídí jen tím, jestli je vůbec vykreslený. Je tu ale jedna výjimka, protože `aria-describedby` a `aria-labelledby` použijí text cílového prvku pro výpočet jména nebo popisu, i když má ten prvek `aria-hidden="true"` (viz výše). Tenhle atribut totiž zablokuje jen to, aby byl prvek vidět jako samostatná položka stromu přístupnosti, ne jeho použití jako zdroje textu pro jiný prvek.

Proč to takhle je

- Prvky odstraňuje jen z rozhraní pro asistivní technologie, ne vizuálně. Proto se nehodí jako náhrada za skutečné skrytí (`hidden, display:none`), jen jako doplněk k redundantnímu nebo dekorativnímu obsahu, který má zůstat vidět.
- Fokusební prvek s `aria-hidden="true"` je past, protože uživatel čtečky se na něj dostane klávesnicí, ale nic mu o něm neřekne.

Příklad správného použití

```
<a href="/kosik">
  <svg aria-hidden="true" focusable="false">...</svg>
  Košík (3 položky)
</a>
```

Čtečka ohlásí jen „Košík (3 položky), odkaz“. Ikonu díky `aria-hidden` úplně přeskočí. Text „Košík (3 položky)“ nese celou informaci sám, ikona vedle něj jen opakuje, co je vidět.

aria-disabled

`aria-disabled` nechává prvek nefunkční, ale pořád dosažitelný a čitelný, na rozdíl od nativního HTML atributu `disabled`. Je součástí WAI-ARIA od verze 1.0, platí i v 1.1–1.3.

Co dělá

Sdělí čtečce, že prvek je vnímatelný, ale momentálně needitovatelný nebo neovladatelný. Na rozdíl od `disabled` ho ale nevyřadí z pořadí procházení Tabem a nic sám o sobě nezmění vizuálně ani funkčně:

```
<div role="button" tabindex="0" aria-disabled="true">Odeslat</div>
```

Čtečka ohlásí „Odeslat, tlačítko, zakázáno“, ale tlačítko zůstává ve fokusovém pořadí, na rozdíl od `<button disabled>`, který by z něj úplně zmizel.

Na co se aplikuje

Kdekoliv, kde má needitovatelný nebo neovladatelný prvek zůstat dosažitelný a čitelný, ne úplně zmizet z pořadí procházení pomocí klávesnice: typicky vlastní ovládací prvky postavené na `<div>`/`<span>` s `role`, nebo tlačítka, u kterých má uživatel pořád vidět, že existují, a proč jsou momentálně nedostupná, ne jen že chybí:

```
<button type="button" aria-disabled="true">Odeslat (vypl?te formulá?)</button>
```

Omezení

- **Sám o sobě nic nevypíná** — na rozdíl od `disabled` musí kliknutí nebo aktivaci zablokovat vlastní kód, `aria-disabled` jen informuje čtečku.
- **Sám o sobě nic nemění vizuálně** — potřebuje vlastní CSS (`[aria-disabled="true"] { opacity: .5; }` apod.), jinak vypadá stejně jako funkční prvek.
- **Na kontejneru zakáže i potomky**, takže `aria-disabled="true"` se na obalujícím prvku dědí na všechny fokusovatelné potomky, i ty, co zakázat nechtěl.
- **`pointer-events: none` nestačí** — zablokuje myš, ne klávesnici. Funkčnost musí zablokovat kód, ne jen CSS.

Hodnota a vztah k jiným atributům

Hodnota je `true` nebo `false`. U nativních formulářových prvků existuje i HTML atribut `disabled`, ten prvek navíc vyřadí z fokusu, ze sekvence odeslání formuláře a automaticky mu dá výchozí vzhled (`:disabled` v CSS). `aria-disabled` nic z toho samo nedělá, jen sdělí stav. Zbytek je na autorovi.

Proč to takhle je

- Zůstat fokusovatelný a čitelný má smysl tam, kde uživatel potřebuje vědět, že prvek existuje, a proč je nedostupný. Nejde o to ho úplně ztratit z procházení klávesnicí, jako by tam nikdy nebyl.
- Typický případ: tlačítko „Odeslat“ dostupné, jen dokud formulář není validní. `disabled` by ho úplně schovalo z pořadí Tabu, `aria-disabled` ho ale nechá na místě a čtečka pak může ohlásit i důvod, je-li napsaný (např. přes `aria-describedby`).

Příklad správného použití

JanBalak.name — <https://www.janbalak.name/clanky/dalsi-aria-parametry-prakticky-aria-describedby-aria-hidden-a-aria-disabled.html>

```
<button type="button" aria-disabled="true" aria-describedby="odeslat-duvod">Odeslat</button>  
<p id="odeslat-duvod">Formulář obsahuje chyby, opravte je před odesláním.</p>
```

Čtečka ohlásí „Odeslat, tlačítko, zakázáno, Formulář obsahuje chyby, opravte je před odesláním.“ Tlačítko zůstává dosažitelné klávesnicí, popis přes `aria-describedby` z první části tohoto dílu ohlásí i důvod, a samotné zablokování kliknutí nebo aktivace zajistí kód (zde kvůli přehlednosti vynechaný).

Zdroje

- [ARIA: aria-describedby attribute](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-describedby) (<https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-describedby>)
- [ARIA: aria-hidden attribute](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-hidden) (<https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-hidden>)
- [ARIA: aria-disabled attribute](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-disabled) (<https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-disabled>)