Engineer

ARIA

balisage ajouté qui nomme une commande aux aides techniques

Ce que c’est

ARIA est une spécification W3C d’attributs HTML qui indique aux technologies d’assistance la nature, le nom et l’état d’un élément lorsque HTML seul en dit trop peu. Les lettres signifient Accessible Rich Internet Applications. Le titre complet WAI-ARIA ajoute la Web Accessibility Initiative du W3C. La recommandation actuelle est la version 1.2.

Un attribut `role` indique le type d’élément : composant comme `button` ou `tab`, structure comme `list`, l’un des huit rôles de région tels que `navigation` ou `main`, ou région dynamique comme `status`, dont les changements sont annoncés spontanément par un lecteur d’écran. Les attributs `aria-*` se répartissent ensuite en propriétés, comme `aria-labelledby`, et états, comme `aria-expanded`, qui changent avec les actions. Le navigateur traduit l’ensemble vers l’API d’accessibilité du système, que lit le lecteur d’écran ou la commande vocale.

ARIA ne change que ce qui est annoncé. Un `div` avec `role="button"` est lu comme bouton, mais ne reçoit aucun focus et ignore Entrée et Espace tant qu’un `tabindex` et un script n’apportent pas ces comportements. D’où la première règle du guide W3C Using ARIA : si un élément ou attribut HTML natif possède déjà la sémantique et le comportement nécessaires, utilisez-le plutôt que de détourner un autre élément et d’ajouter ARIA.

```
<!-- announced as a button, behaves like a div -->
<div role="button">Send</div>

<!-- announced, focusable and operable as a button -->
<button type="button">Send</button>
```

L’autre moitié est le nom accessible, les mots prononcés pour une commande. Accessible Name and Description Computation 1.2 fixe l’ordre : d’abord `aria-labelledby`, puis `aria-label`, puis l’étiquette native — élément `label`, `alt`, `title` SVG —, puis le texte de l’élément pour les rôles qui en tirent leur nom, et l’attribut `title` en dernier. Un `aria-label` remplace donc les mots visibles d’un bouton au lieu de s’y ajouter.

Utilisez le concept pour prendre une décision

Utilisez « Que signifie ARIA ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des interfaces rapides, utilisables sur téléphone et cohérentes dans tous les navigateurs de vos clients.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Développement cartographique GIS full-stack — cartographie full-stack, de la base au navigateur

Développement cartographique GIS full-stack Contactez-nous →

Pourquoi c’est important

ARIA pose à une équipe le choix entre construire et réutiliser. Le critère WCAG 2.2 4.1.2, Nom, rôle et valeur, exige que chaque composant expose nom et rôle aux logiciels, permette de définir par programme ses états et valeurs modifiables par l’utilisateur et signale leurs changements. Un `button`, `select` ou `details` natif le satisfait d’emblée. Des onglets, listes combinées ou arbres personnalisés ne le font qu’avec ARIA et le script qui maintient ses informations exactes.

Un composant personnalisé devient donc un coût récurrent : traitement du clavier, gestion du focus, états suivant chaque interaction et tests avec plusieurs lecteurs d’écran à chaque refonte. Jugez une bibliothèque selon le modèle de l’ARIA Authoring Practices Guide du W3C suivi par chaque composant, et selon la présence effective du comportement clavier lorsque la souris est rangée.

La pression juridique atteint désormais les entreprises privées. La directive (UE) 2016/2102 sur l’accessibilité Web couvre les sites et applications publics. L’European Accessibility Act, directive (UE) 2019/882, s’applique depuis le 28 juin 2025 à différents produits et services de consommation, dont commerce électronique et banque aux particuliers, avec exemption des microentreprises fournissant des services. Les deux s’appuient sur EN 301 549, norme harmonisée dont les exigences Web sont les WCAG niveau AA. Nom, rôle et valeur est de niveau A : une commande cachant son nom ou son rôle à un lecteur d’écran échoue donc au niveau minimal examiné par l’auditeur.

À quoi cela ressemble en pratique

Les modèles de ce site utilisent ARIA avec parcimonie. La barre supérieure de chaque page intérieure est un `div` avec `role="banner"`, pas le `header` habituel. Le commentaire explique pourquoi : la feuille de style habille un `header` nu comme le grand visuel d'accueil pleine hauteur. L'élément imposerait donc cette géométrie aux pages intérieures, tandis que le rôle apporte seulement la région. C'est lui qui place l'emblème, lien vers l'accueil, dans une région : le cas limité autorisé par la première règle. Le menu prend l'autre voie : une paire `details` et `summary` laisse au navigateur l'état ouvert, la sémantique du bouton et `aria-expanded`. Le `nav` du menu et le `nav` du pied portent des `aria-label` distincts, car deux régions de navigation homonymes enfreignent la règle d'unicité du moteur de contrôle, et le contrôle d'accessibilité obligatoire du site exige zéro constat. La mise en œuvre conservée du filtre de portfolio utilise une région `role="status"` conçue pour rester silencieuse avant la saisie. Ce filtre est actuellement désactivé par `searchEnabled = false` : il illustre le code conservé, pas une interaction disponible sur les pages publiées.

Le récit du passage du site aux WCAG 2.2 niveau AA complète l'histoire du bandeau : l'activation des règles de bonnes pratiques du moteur a immédiatement provoqué un échec, car sur chaque page intérieure l'emblème n'appartenait à aucune région. La même page cite l'absence de tests avec de véritables technologies d'assistance parmi les limites publiées.

Le récit des onze défauts déclarés sains par le navigateur montre l'écart entre balisage et expérience. Tous concernaient le rendu, pas les déclarations du balisage, seule chose touchée par ARIA. Huit liens du pied restaient atteignables par Tab après qu'un style avait supprimé uniquement l'interaction au pointeur. C'est la même structure qu' `aria-hidden` sur un élément focalisable : le lecteur d'écran le perd, le clavier le conserve. Et seules soixante-dix des cent cinq règles du moteur étaient actives, celle signalant une région manquante figurant parmi les désactivées.

La dernière page montre le même problème là où ARIA ne joue aucun rôle. Nommer les dix-sept icônes d'interface pour les lecteurs d'écran concerne SwiftUI, sans HTML ni attribut `aria-`. Son étiquette est propre à SwiftUI, équivalent natif d' `aria-label`, pas instance de cet attribut. Les symboles système étaient lus sous leur nom interne : le bouton d'envoi annonçait « arrow up circle, button », et dix-sept commandes à icône seule décrivaient leur dessin plutôt que leur fonction. Chacune a reçu une étiquette nommant l'action et suivant l'état si nécessaire. Un analyseur exige désormais une étiquette dans les

huit lignes entourant la construction qui crée une commande à icône seule : une dix-huitième commande ne peut entrer sans elle. La limite publiée vaut aussi pour `aria-label` : le contrôle confirme une étiquette proche, mais seule une personne qui écoute peut juger sa justesse.

Où les problèmes surviennent

Un rôle annonce un composant jamais atteint au clavier. Un `div` reçoit `role="button"` et un gestionnaire de clic. Le lecteur dit « bouton », mais Tab le passe et Entrée ne fait rien. La troisième règle d'ARIA exige que toute commande interactive fonctionne au clavier. Utilisez l'élément natif ou construisez tout le modèle clavier APG, puis testez avec la souris débranchée.

Le focus atteint un élément considéré absent par le lecteur d'écran. `aria-hidden="true"` est placé sur un élément focalisable ou son ancêtre, comme un panneau hors écran. Tab y va toujours et l'utilisateur entend le silence. La quatrième règle l'interdit. Retirez aussi le contenu caché de l'ordre de tabulation avec `hidden`, `display: none` ou `inert`, et réservez `aria-hidden` aux décorations qui ne reçoivent jamais le focus.

Un nom est donné à un élément qui n'en accepte pas. `aria-label` sur `div`, `span` ou `p` veut l'expliquer, mais WAI-ARIA 1.2 interdit de nommer `generic` et `paragraph` : les lecteurs l'ignorent ou le lisent irrégulièrement. Sur une commande, l'étiquette remplace les mots visibles. S'ils diffèrent, une personne utilisant la commande vocale et prononçant les mots visibles à l'écran manque sa cible. Nommez les éléments autorisés et commencez chaque `aria-label` par les mots visibles, comme le demande WCAG 2.5.3, Étiquette dans le nom.

Deux rôles se contredisent. Un rôle explicite remplace le rôle natif. Un lien avec `role="button"` est annoncé comme bouton mais fait défiler avec Espace ; `role="tab"` sur `h2` le retire des titres ; `role="navigation"` sur `ul` perd la liste et son nombre d'éléments. La deuxième règle demande de ne changer la sémantique native que si nécessaire, et ARIA in HTML énumère les rôles acceptés par chaque élément.

Une région dynamique bavarde. `aria-live="assertive"` ou `role="alert"` sur un compteur de résultats ou compte à rebours modifié à chaque frappe ou seconde interrompt l'utilisateur pour chaque valeur. Utilisez les mises à jour polies que fournit par définition `role="status"`, annoncez une fois la saisie stabilisée et gardez la région silencieuse au chargement.

Pour aller plus loin

- WAI-ARIA 1.2 — rôles, états, propriétés, attributs pris en charge et rôles qui ne peuvent être nommés.
- ARIA in HTML — rôles et attributs autorisés par élément HTML et ceux qui sont redondants.
- Using ARIA — guide pratique W3C avec les règles d'utilisation et un exemple de chacune.
- Accessible Name and Description Computation 1.2 — algorithme déterminant le nom d'une commande annoncé par un lecteur d'écran.
- Comprendre Nom, rôle et valeur — exigences de WCAG 2.2 4.1.2 pour chaque composant, avec techniques et échecs.
- ARIA Authoring Practices Guide — balisage et comportement clavier de chaque modèle courant, avec exemples fonctionnels.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Que signifie ARIA ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

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

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/aria/