Lignes directrices ARIA

Introduction

HTML (Hypertext Markup Language, qu'on peut traduire par « langage de balises pour l'hypertexte ») a été initialement conçu pour créer des pages de texte statiques. JavaScript fournit des **composants** dynamiques qui n'existent pas sous la forme native du HTML. La norme ARIA (**A**ccessible **R**ich **I**nternet **A**pplications, qu'on peut traduire par « applications internet riches et accessibles ») offre aux développeurs des fonctionnalités d'accessibilité qui ne sont pas disponibles uniquement en HTML.

ARIA est un ensemble d'attributs (propriétés des étiquettes HTML) qui rendent le contenu Web et les applications plus accessibles. Lorsqu'il n'est pas pratique d'utiliser le HTML sémantique, ARIA peut communiquer des informations sémantiques et des interactions aux **technologies d'assistance**. ARIA communique des informations sémantiques sur les **widgets** (éléments de l'interface utilisateur (IU)), les structures (relations) et les comportements à la technologie d'assistance. Une expérience accessible appropriée dépend de l'utilisation appropriée de cet outil : une mauvaise utilisation d'ARIA peut conduire à des expériences médiocres ou inutilisables.

Les attributs ARIA peuvent rendre le HTML non sémantique comme les `<div>`s plus accessibles lorsque le code natif ou la technologie sous-jacente ne le permettent pas. Par exemple, supposez que vous utilisez une bibliothèque ou un cadre de travail externe qui utilise un `<div>` comme un bouton. Il peut être plus facile d'ajouter une étiquettes ARIA telle que `role="button"` au composant, plutôt que de réécrire les styles et les fonctions existants. Considérez cette approche avec soin, car vous devrez également vous assurer qu'il y a du JavaScript approprié en place pour rendre l'élément accessible (par exemple, assurez-vous qu'il est focalisable et définissez des gestionnaires d'événements pour les événements de click et de touche enfoncée).

Si vous avez l'intention d'améliorer l'accessibilité du HTML avec les étiquettes ARIA, veuillez consulter les pratiques de création et les exemples détaillés ci-dessous. Pour résumer la première règle d'ARIA : si vous pouvez utiliser des attributs HTML ou sémantiques, alors faites-le. Une autre expression courante est qu'un bon ARIA vaut mieux que pas d'ARIA, mais pas d'ARIA vaut mieux qu'un mauvais ARIA.

Exemples d'utilisation d'ARIA

Points de repère

- Les points de repère définissent les sections d'une page Web, ce qui aide les utilisateurs de lecteurs d'écran à naviguer dans la page. Les repères peuvent être définis soit sous forme d'éléments HTML, soit avec des rôles ARIA : Un repère `<header>` peut être remplacé, si nécessaire, sur n'importe quel élément (comme un `<div>`) en ajoutant `role="banner"`.
- Un point de repère `<aside>` peut être remplacé par `role="complementary"`.

- Un point de repère `<footer>` peut être remplacé par `role="contentinfo"`.

Rôles d'ARIA et le JavaScript

Certains cadres de travails JavaScript génèrent des pages Web avec du code non sémantique, par exemple un élément `<div>` au lieu d'un `<button>`. L'ajout de l'ARIA `role="button"` améliorera la signification sémantique du .

Lorsque JavaScript est utilisé pour créer des composants spéciaux qui n'existent pas nativement en HTML, ils peuvent avoir besoin de se voir attribuer un rôle ARIA pour définir ce qu'ils sont. Par exemple :

- Un accordéon sera développé et réduit avec un bouton qui a cet attribut : `aria-expanded="true"`/`"false"`.
- Un ensemble d'onglets dans une page Web aura besoin de `role="tablist"`, `role="tab"` et `role="tabpanel"`.

D'autres exemples peuvent être trouvés sur le site ARIA Authoring Practices site. (Offert en anglais seulement)

Étiquettes

Il existe plusieurs façons d'étiqueter un bouton – en plus d'utiliser le texte à l'écran – par exemple :

- Un attribut aria-label avec un texte spécifiant l'étiquette, c'est-à-dire `aria-label="continuer"`.
- Un attribut `aria-labelledby` peut pointer vers du texte existant à l'écran comme étiquette.
- Un attribut `aria-describedby` peut pointer depuis un champ de saisie vers une erreur en ligne pour les connecter de manière sémantique (significative). Un lecteur d'écran annoncera automatiquement l'erreur lorsque l'utilisateur accède au champ de saisie.

Régions d'ARIA Live

Une région ARIA live est une section d'une page Web que les lecteurs d'écran annonceront automatiquement si son contenu change dynamiquement après le chargement initial de la page. La région en direct fournit des messages d'état importants. Les exemples de code incluent :

- `role="alert"` ou `aria-live="assertive"` indique au lecteur d'écran d'interrompre un message en cours pour annoncer un changement de contenu dynamique.
- `aria-live="polite"` indique au lecteur d'écran d'annoncer la région en direct mise à jour après avoir annoncé le contenu actuel.

L'attribut aria-hidden

L'attribut `aria-hidden="true"` indique aux lecteurs d'écran d'ignorer un élément et de ne pas l'annoncer. `aria-hidden` peut être utile pour le contenu affiché à l'écran mais déroutant pour un utilisateur de lecteur d'écran.

Par exemple :

- Une icône placée à côté du texte redondant (comme une icône de poubelle avec le mot « Supprimer » à côté) puisqu'un utilisateur n'a pas besoin que les deux soient annoncés.
- Contenu d'arrière-plan grisé lorsqu'une fenêtre contextuelle envahit la page. Lorsqu'une fenêtre contextuelle reprend la page, un utilisateur de lecteur d'écran ne lira pas la page d'arrière-plan.

Ressources

- W3C WAI-ARIA Overview (Offert en anglais seulement)
- WAI-ARIA Authoring Practices 1.2 (Offert en anglais seulement)