Engineer

Bouton d'interface

une commande que l'on active pour déclencher une action

Ce que c’est

Un bouton d’interface est l’élément qui agit lorsqu’on l’active : il envoie un message, enregistre un brouillon, ouvre une boîte de dialogue ou coupe le son d’un appel. Sur le Web, il s’agit généralement de l’élément HTML `<button>`. En matière d’accessibilité, c’est tout ce qui porte le rôle `button`. « Bouton-poussoir » est un nom plus ancien pour la même chose.

La définition de ce rôle dans WAI-ARIA 1.2 peut se traduire ainsi : « une entrée qui permet à l’utilisateur de déclencher des actions par un clic ou une pression ». La spécification ajoute que les boutons servent surtout à des actions ponctuelles. Un simple bouton de commande ne conserve aucun état. Un bouton bascule, marqué par `aria-pressed`, reste activé ou désactivé jusqu’à une nouvelle pression. Un bouton de menu, marqué par `aria-haspopup`, ouvre une liste de choix supplémentaires.

La distinction essentielle sépare le bouton du lien. Un lien emmène le lecteur quelque part ; un bouton change quelque chose à l’endroit où il se trouve déjà. L’ARIA Authoring Practices Guide considère ces deux fonctions comme nettement différentes. Lorsqu’un lien est dessiné comme un bouton, ou inversement, il préfère adapter l’apparence à la fonction plutôt que d’emprunter le rôle de l’autre élément.

Le HTML Living Standard décrit l’élément `<button>` comme un bouton dont le contenu fournit l’étiquette. Son attribut `type` possède trois mots-clés. `submit` envoie le formulaire auquel le bouton appartient, `reset` rend aux champs du formulaire leurs valeurs initiales et `button` ne fait rien par lui-même, laissant l’action au script. Lorsque `type` manque ou est mal orthographié, le bouton est

dans l'état Auto. Un bouton dans cet état, à l'intérieur d'un formulaire et sans commande associée, est un bouton d'envoi du formulaire. Aucun contenu interactif ne peut se trouver à l'intérieur : ni lien, ni champ, ni second bouton.

La norme permet aussi à un bouton d'agir sur un autre élément sans script : `commandfor` désigne la cible par son identifiant et `command` indique l'action, comme afficher un popover ou ouvrir une boîte de dialogue modale. Pour savoir quelle part de ces fonctions un navigateur donné prend en charge, il faut consulter ses propres notes de version.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'un bouton d'interface ? » 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

Le bouton est le point où la lecture devient une action : acheter, envoyer, enregistrer et supprimer passent tous par lui. Un bouton que certaines personnes ne peuvent trouver, atteindre ou comprendre les exclut de la partie du produit qui le finance.

L'élément natif arrive avec son comportement. Un `<button>` reçoit le focus au clavier, exécute son action avec Espace et Entrée, indique aux technologies d'assistance qu'il est un bouton et quel nom il porte, et cesse de répondre lorsqu'il porte `disabled`. Le document explicatif du critère de réussite WCAG 2.2 4.1.2, Nom, rôle et valeur, précise que les commandes HTML standard satisfont déjà ce critère lorsqu'elles sont utilisées conformément à la spécification. Un `<div>` avec un gestionnaire de clic n'a rien de tout cela. Chaque élément omis exclut quelqu'un.

Le nom est la promesse. Un lecteur d'écran annonce un bouton par son nom accessible et son rôle. Le nom doit donc dire ce que le bouton fait, pas à quoi il ressemble. L'APG ajoute pour les boutons bascules une règle facile à enfreindre : l'étiquette reste identique lorsque l'état change. Un bouton nommé « Mute » (couper le son) est annoncé comme « Mute toggle button pressed » (« bouton

bascule Mute enfoncé ») lorsqu'il est activé. Si le design préfère une étiquette alternant entre « Mute » et « Unmute » (rétablir le son), `aria-pressed` n'est pas nécessaire, puisque l'étiquette elle-même indique l'état.

La taille est une exigence, pas une préférence. Le critère de réussite WCAG 2.2 2.5.8, Taille de la cible (minimum), au niveau AA, demande qu'une cible de pointeur mesure « au moins 24 par 24 pixels CSS ». Les exceptions couvrent les petites cibles suffisamment espacées, une commande de taille suffisante ailleurs sur la page accomplissant la même action, les liens à l'intérieur d'une phrase, les commandes dont le navigateur fixe la taille et les cibles dont la taille est essentielle. Un tremblement, un bus en mouvement et un pouce sur un téléphone rendent tous les petites cibles plus difficiles à utiliser.

Le focus doit arriver à un endroit logique. Après une pression, l'APG prévoit que le focus entre dans la boîte de dialogue ouverte par le bouton puis, à sa fermeture, revienne généralement au bouton qui l'avait ouverte. Lorsque la pression laisse le lecteur dans le même contexte, comme avec un bouton Appliquer, le focus reste généralement sur le bouton. Une personne utilisant le clavier et renvoyée en haut de page après chaque enregistrement doit recommencer la page à chaque fois.

À quoi cela ressemble en pratique

Le cas le plus clair commence par une annonce. Dans une application SwiftUI construite à partir de symboles système, le lecteur d'écran lisait le bouton d'envoi comme « arrow up circle, button », parce qu'un symbole sans étiquette est annoncé sous son nom interne. Le travail consistant à nommer les dix-sept commandes à icône seule pour les lecteurs d'écran a donné à chacune une étiquette indiquant son action plutôt que sa forme. Lorsque le sens d'une commande dépend de son état, l'étiquette suit l'état. La partie la plus durable est le contrôle : un outil d'analyse trouve la construction créant une commande à icône seule et exige une étiquette d'accessibilité dans les huit lignes autour d'elle. Cette distance de huit lignes a été définie à partir de mesures et est volontairement rudimentaire : la règle prouve qu'une étiquette existe à proximité, pas qu'elle est juste. Quelqu'un doit encore écouter.

Une deuxième leçon concerne ce qu'un bouton ne peut pas faire. Dans la couche de paiement et de droits d'accès construite sur Stripe et les achats intégrés Apple et Google, un schéma d'accès séparé de 11 tables décide si un compte

peut utiliser l'annuaire des entreprises, la recherche et l'export de données. Le serveur pose à nouveau cette question pour chaque requête traitée, car un bouton caché est une courtoisie, pas un contrôle.

Les boutons comptent aussi pour les lecteurs qui ne sont pas des personnes. La page du site sur l'adaptation aux agents IA indique que de nombreux agents pilotant un navigateur lisent une page par son arbre d'accessibilité, comme un lecteur d'écran. De vrais boutons et des formulaires étiquetés font partie de ce qui permet à un agent de trouver un formulaire et de le remplir.

Les modèles de ce site ne contiennent aucun élément `<button>`. L'appel « Contactez-nous », avec son fond plein, est un `<a>` habillé en bouton parce qu'il conduit à la page de contact. Sa règle de style fixe une `min-height` de 24 pixels avec un commentaire citant le critère de réussite 2.5.8. Le menu principal est un élément `<details>` dont le `<summary>` l'ouvre et le ferme. Le commentaire du modèle précise que le navigateur gère l'état ouvert et la sémantique du bouton, sans script.

Où les problèmes surviennent

- **Une boîte cliquable remplace l'élément.** Un `<div>` ou un `<span>` avec un gestionnaire de clic répond à la souris et guère à autre chose : il ne reçoit pas le focus, ignore Espace et Entrée et est annoncé comme du texte simple. Ajouter `role="button"` ne corrige que l'annonce ; la gestion du focus et du clavier reste à programmer à la main.
- **Un lien fait le travail d'un bouton, ou l'inverse.** Un lien supprimant un enregistrement peut être ouvert dans un nouvel onglet ou mis en favori, ce qui répète la suppression ou en fait perdre la trace. Un bouton qui ne fait que naviguer ne peut pas être ouvert dans un nouvel onglet et n'a pas d'adresse à copier.
- **Un bouton sans type à l'intérieur d'un formulaire l'envoie.** L'absence de `type` plaçant le bouton dans l'état Auto, un bouton « Ajouter une ligne » ou « Afficher le mot de passe » situé dans un formulaire envoie celui-ci dès qu'on appuie dessus. Écrire `type="button"` sur chaque bouton qui ne doit pas envoyer le formulaire supprime la surprise.
- **L'icône constitue toute l'étiquette.** Une poubelle, un crayon ou une flèche a un sens pour la personne qui l'a dessinée. Sans nom accessible, le lecteur d'écran lit un nom de fichier, un nom de symbole ou simplement « button ». Un mot visible est préférable. Là où il n'y a pas de place, un nom accessible indiquant l'action comble le manque.

- **Un bouton caché ou désactivé est considéré comme une sécurité.**

Retirer le bouton de la page n'arrête que les utilisateurs de bonne foi ; la requête sous-jacente peut toujours être envoyée. Le serveur doit décider lui-même si ce compte peut effectuer cette action.

Pour aller plus loin

- WAI-ARIA 1.2 : rôle button — définition du rôle, variantes bascule et menu, et états et propriétés pris en charge.
- HTML Living Standard : élément button — modèle de contenu, types submit, reset et button, état Auto et attributs `command` et `commandfor` .
- ARIA Authoring Practices Guide : modèle de bouton — boutons et liens, comportement clavier, destination du focus après une pression et règle des étiquettes de bascule.
- Comprendre le critère 2.5.8 : taille de la cible (minimum) — minimum de 24 par 24 pixels, test d'espacement et autres exceptions.
- Comprendre le critère 4.1.2 : nom, rôle et valeur — pourquoi chaque commande nécessite un nom et un rôle lisibles par un logiciel, et quand les commandes HTML standard satisfont déjà le critère.



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 « Qu'est-ce qu'un bouton d'interface ? » 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/button/