ACCESSIBILITY DAYS 2025

Oltre i confini del default: progettare componenti non standard accessibili

DIANA BERNABEI
Esperta di accessibilità @ Inclusio

16 MAGGIO > 17:30

TRACK 2

Chi sono

Diana Bernabei

Accessibility Engineer & Frontend Developer

- Founder e CEO @ Inclusio
- Certificata web accessibility expert
- +4 anni di esperienza nel settore
- Collaboratrice con Accessibility Days APS

Inclusio

Scaletta dei temi

1. Sfide componenti non standard
2. Utilizzare le WAI-ARIA
3. Casi studio pratici
4. Strumenti di testing e best practice

Le sfide dei componenti non standard

Mancanza di semantica nativa

I componenti personalizzati spesso non hanno una semantica standard che permette le tecnologie assistive una semplice interpretazione.

Interazione complessa

Pattern di interazione sofisticati richiedono soluzioni custom per navigazione da tastiera.

Stati dinamici

I cambiamenti di stato devono essere comunicati correttamente agli screen reader.

Perché WAI-ARIA è fondamentale

Colma il divario semantico

Gli attributi ARIA arricchisce gli elementi HTML con ruoli e proprietà semantiche mancanti.

Consente agli elementi di comunicare il loro scopo reale agli screen reader.

Comunica comportamenti dinamici

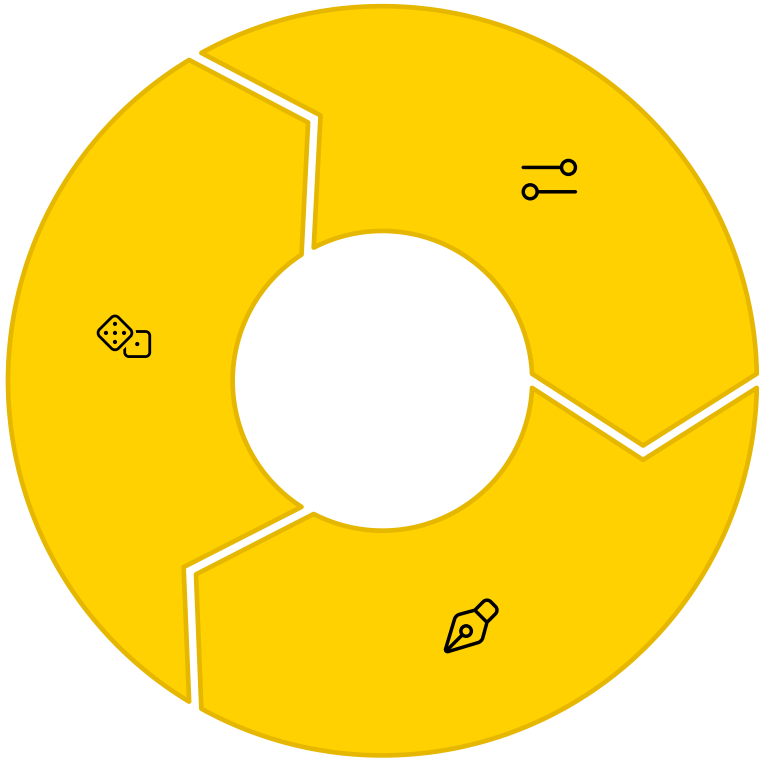
Permette di gestire cambiamenti di stato senza richiedere aggiornamenti di pagina completi.

Offre feedback in tempo reale agli utenti con disabilità visive.

I tre pilastri di WAI-ARIA

Ruoli

Definiscono cosa è un elemento. Comunicano il tipo di widget o struttura.



Stati

Riflettono le condizioni correnti dell'elemento. Cambiano spesso durante l'interazione.

Proprietà

Forniscono informazioni aggiuntive sul comportamento o relazioni dell'elemento.

La prima regola e più importante regola è non usare attributi ARIA se non necessario

Semantica nativa prima

Preferire elementi HTML semantici quando disponibili.

ARIA non corregge tutto

Aggiungere ARIA a elementi inaccessibili non risolve problemi strutturali.

Minimo indispensabile

Utilizzare solo gli attributi ARIA necessari per colmare lacune semantiche.

Tipi di navigazione

1

Navigazione visiva (mouse)

Basata su riferimenti visivi e interazione diretta con gli elementi grafici

2

Navigazione uditiva (tastiera)

Dipende dal feedback sonoro e dall'ordine logico degli elementi

Un'interfaccia veramente accessibile deve supportare entrambe le modalità, garantendo un'esperienza equivalente a tutti gli utenti.

Casi Studio: Componenti Interattivi Accessibili

Componente switch

Un controllo binario che necessita di stati e feedback chiari per tutti gli utenti.

Menù drodpown

Un pattern di navigazione che deve mantenere il focus e comunicare l'espansione agli screen reader.

Navigazione tab

Interfaccia a schede che richiede gestione del focus e relazioni tra contenuti correlati.

Carosello

Slideshow di contenuti che deve offrire controlli accessibili e alternative alla navigazione visiva.

Switch Accessibile

Caso studio: Switch accessibile

Attributo ARIA	Elemento	Scopo
role="group"	Contenitore degli switch	Racchiudere semanticamente i controlli
role="switch"	Pulsante che si comporta da switch	Trasforma il ruolo di pulsante in interruttore/switch
aria-labelledby="id"	Contenitore	Collega il titolo al contenitore
aria-checked="true/false"	Pulsante	Permette di salvare lo stato on/off

Quali eventi dobbiamo ascoltare

Evento	Scopo
click	Aggiunto ad ogni elemento con ruolo switch per alternare gli stati on/off di aria-checked true/false

Solo questo? Si

Eventi ascoltati in modo implicito

Evento	Scopo
focus	Elemento button ha il focus nativamente
keydown / keyup / keypress	Permette l'ascolto della barra spaziatrice e del pulsante invio che fa scattare lo stesso evento del click, quindi il cambio di stato

E ora andiamo al codice: **https://github.com/dianaberna/ad2025-oltreiconfinideldefault/blob/main/switch.html**

Dropdown Accessibile

Caso studio: Dropdown accessibile

Attributo ARIA	Elemento	Scopo
role="menu"	Contenitore del menu	Comunica agli screen reader che l'elemento è un menu
role="menuitem"	Voci del menu	Identifica le opzioni selezionabili all'interno del menu
aria-expanded="true/false"	Pulsante trigger	Comunica lo stato di apertura o chiusura del dropdown
aria-haspopup="true/false"	Annuncia l'apertura di un popup	Comunica che attivando questo elemento si aprirà qualcos'altro
aria-controls="id"	Contenitore del menu	Associa il menu al suo pulsante di controllo

Quali eventi dobbiamo ascoltare per il mouse?

Evento	Scopo
click	Sul pulsante principale
click	Sui pulsanti dei sottomenu
click	Sull'intero documento per poter chiudere tutto il dropdown

Quali eventi dobbiamo ascoltare per la tastiera?

Tasto	Azione
ArrowDown	Passa al menu item successivo
ArrowUp	Torna al menu item precedente
ArrowRight	Apre un sotto-menu, se presente
ArrowLeft	Chiude il sotto-menu e torna al genitore
Escape	Chiude il menu corrente o tutto
Enter / Space	Attiva il pulsante o apre/chiude un sotto-menu

E il focus?

Il focus non va gestito in ogni elemento ma semplicemente deve essere spostato all'interno dell'elenco dei link o dei pulsanti del dropdown.

E ora andiamo al codice: **https://github.com/dianaberna/ad2025-oltreiconfinideldefault/blob/main/dropdown.html**

Tab Panel Accessibile

Caso studio: Tab Panel accessibile

Attributo ARIA	Elemento	Scopo
role="tablist"	Contenitore tab	Identifica il gruppo di tab
role="tab"	Singola tab	Definisce l'elemento come selettore
role="tabpanel"	Pannello contenuto	Collega il contenuto alla tab
aria-selected	Tab attiva o disattiva	Comunica quale tab è selezionata
aria-controls	Button tab	Collega il tab al rispettivo tabpanel tramite ID.
aria-labelledby	tabpanel, tablist	Associa il contenuto o il gruppo al suo titolo/tab.

Quali eventi ascoltiamo in modo esplicito?

Evento	Scopo
Click	Per attivare il tab selezionato.
Keydown	Per supportare la navigazione da tastiera (ArrowLeft, ArrowRight, Home, End).
Focus	Dopo la selezione il focus viene spostato al tab attivo
tab-index=-1	Gestisco il focus solo tramite javascript e non tramite Tab
tab-index=0	Rende focusabile un elemento che normalmente non lo è

E ora andiamo al codice: **https://github.com/dianaberna/ad2025-oltreiconfinideldefault/blob/main/tab.html**

Carosello Accessibile

Caso studio: Carosello accessibile

Attributo ARIA	Scopo
role="region"	Definisce la sezione principale del carosello.
role="group"	Ogni slide è un gruppo logico.
role="tablist"	Wrapper dei bottoni indicatore delle slide.
role="tab"	Ogni indicatore è una tab.
role="status"	Area per annunci tramite aria-live.

Altri attributi ARIA utilizzati

Evento	Scopo
aria-roledescription="carousel"	Descrizione personalizzata del ruolo region.
aria-roledescription="slide"	Descrizione personalizzata dei gruppi slide.
aria-label="Slide 1 di 3"	Per etichettare ogni slide.
aria-label="Slide precedente" / Slide successiva	Per i pulsanti di navigazione.
aria-label="Vai alla slide X"	Per i pulsanti indicatore.
aria-controls	Usato nei bottoni indicatore per riferirsi all’id="carousel-slides" (la viewport).

Eh si altri ancora

Evento	Scopo
aria-selected	Indica quale tab/slide è attualmente selezionata (true o false).
aria-current	segnala la slide attuale nel set degli indicatori.
aria-hidden	Usato dinamicamente via JavaScript per nascondere visivamente e semantica le slide non attive.
aria-live="polite"	Notifica silenziosa dello stato del carosello (annunci in background).

Quali eventi ascoltiamo?

Evento	Scopo
click	Per cambiare slide, precedente, successiva o una slide specifica.
keydown	Per andare alla slide successiva o precedente quando siamo sul menu di navigazione (ArrowLeft e ArrowRight)
change	Abilita/disabilita il loop delle slide

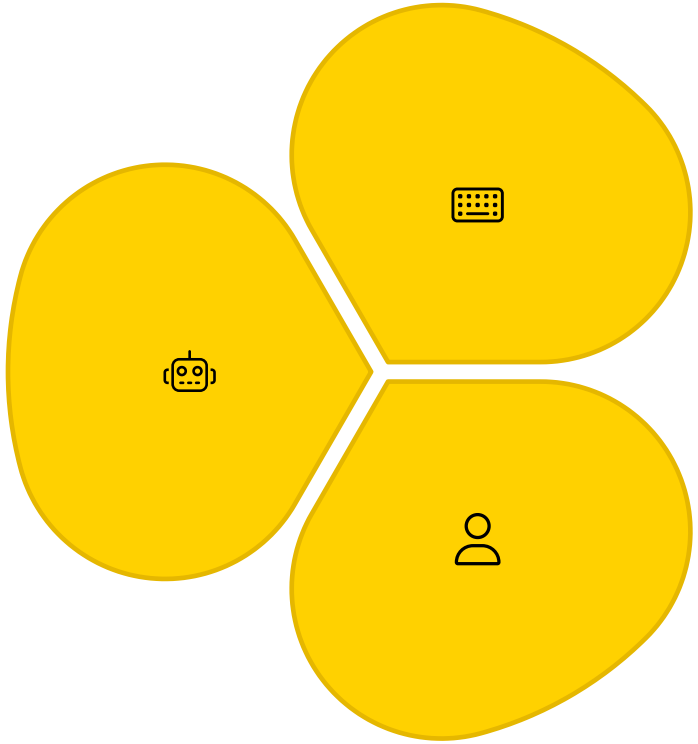
E ora andiamo al codice: **https://github.com/dianaberna/ad2025-oltreiconfinideldefault/blob/main/carousel.html**

Testing dell'accessibilità: un processo continuo

Verifica automatizzata

Utilizzare strumenti come Axe, WAVE o Lighthouse per identificare problemi base.

Integrare i test nel processo di sviluppo continuo.



Test manuali

Verificare la **navigazione da tastiera** e il comportamento con **screen reader**.

Provare componenti con VoiceOver, NVDA o JAWS.

Test con utenti reali

Coinvolgere persone con disabilità per feedback reale sull'usabilità.

Identificare problemi non evidenti con test automatici.

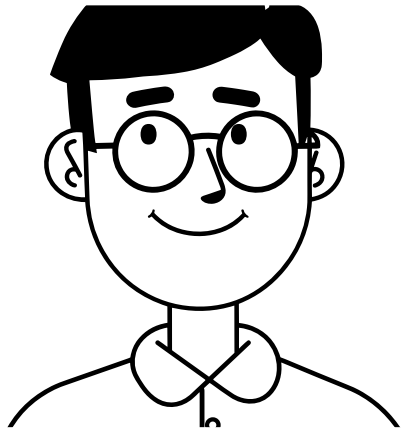
Documentazione: comunicare l'accessibilità

Documentazione tecnica

Documentare attributi ARIA utilizzati e pattern di interazione da tastiera nei commenti del codice.

Libreria di componenti

Creare esempi funzionanti con note sulla compatibilità con tecnologie assistive.



Guide utente

Fornire istruzioni specifiche per chi si occupa dei contenuti .

Grazie per l'attenzione!

Email

diana@inclusio.it

LinkedIn

linkedin.com/in/dianabernabei

Sito Web

www.dianabernabei.it

www.inclusio.it