Empowering every user

/thoughtworks

Patterns + meaning = language

Ability + language barrier = linguistic disability

(Pattern + meaning)

Accessible tech and product language

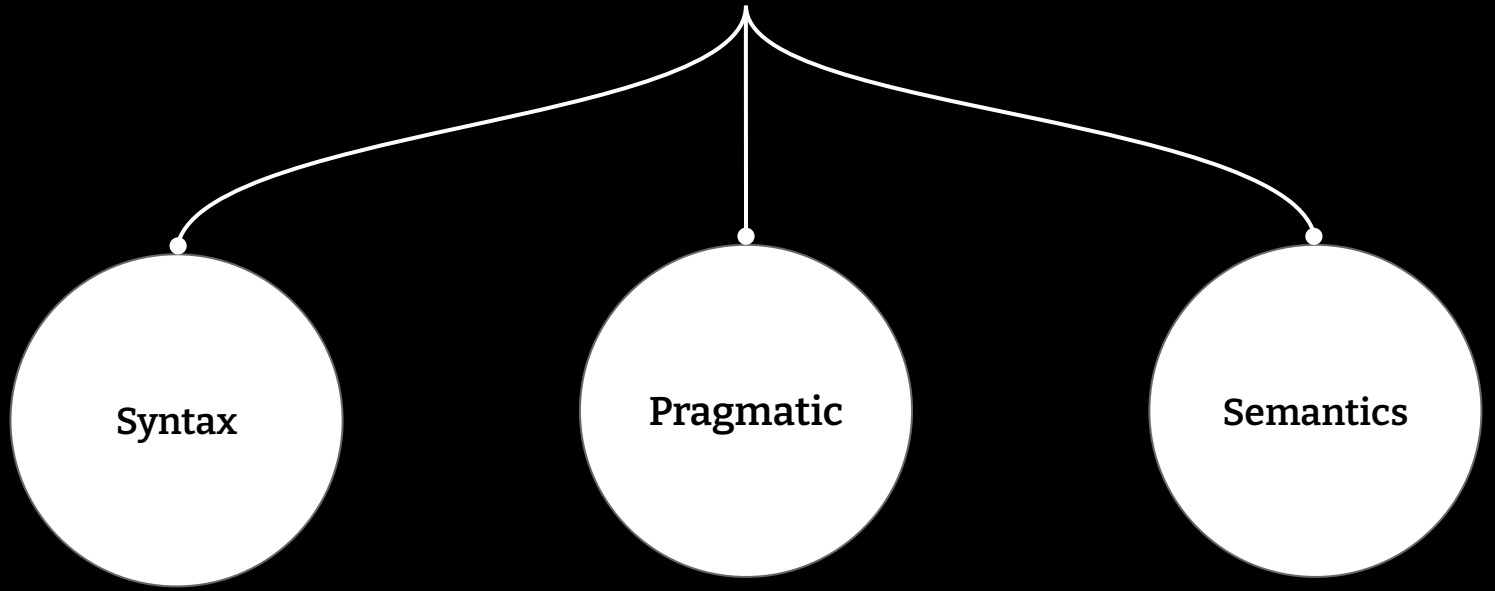
https://www.youtube.com/watch?v=XB4cjbYywqg&t=13s

Access information + recognise pattern + generate meaning + interpret from previous knowledge **= Accessible language**

Semiotics & language

- Human recognise patterns of information and organise them to generate meaning. Collection of these organised patterns form the languages that humans use when they communicate
- When the perception of these patterns leads to interpretation of information in the context of previous knowledge, we might say meaning occurs
- One widely used approach to the study of the relationship among patterns of perception and meaning is called semiotics. Central to semiotics is the notion of the sign

Semiotics



Semiotics

the study of signs and symbols and their use or interpretation.

According to the theory of saussure, sign is the smallest unit of meaning used to communicate. It is the whole that results from the association of the signifier with the signified.

Signifier: any material form, e.g. letter, sound and image
Signified: the concept that signifier refers to

Sign

Signified

Signifier

Concept

Sound.word. Image

Syntax + Pragmatic = Semantics

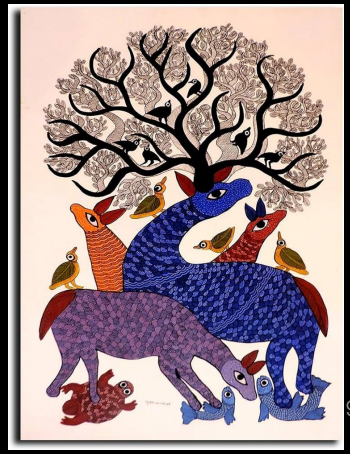
Structure

Interpretation (direct and underlying)

Syntax example: Koob, ookb, Book

Pragmatic example:

Direct meaning is deer but underlying layer is gond art.



Access of information

Differently abled users uses assistive technology to access the informations first before interpreting and get the meaning out from it.

Access for assistive technology + Semantics = Semantically accessible

The foundation of digital product interfaces includes elements, components, patterns, behaviors, and more.

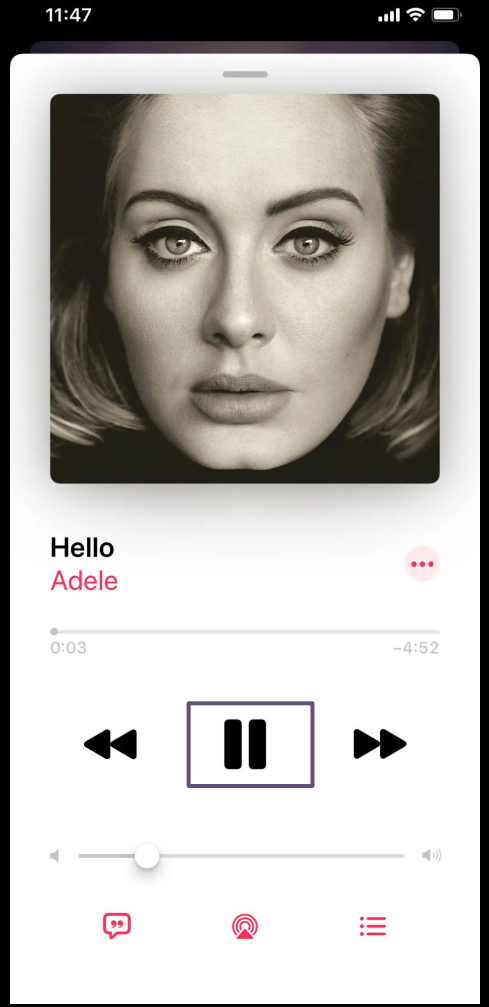
Primary digital consumption: Mobile and Web

Mobile: Numerous iOS apps are inaccessible due to the absence of essential metadata that interacts with assistive technologies like VoiceOver

Vision
Auditory
Motor
Cognitive

Voice over

UiAccessibility Protocol



“Pause, button”

IOS accessibility ecosystem

Assistive technologies

- Voice over - Primary AT
- Dynamic text, zoom etc
- Alternative input - keyboard and braille board

API

- UIAccessibility protocol
- UIAccessibility container protocol
- UIAccessibility notification

Tools

- Accessibility inspector

How to add semantically accessible metadata?

- Apple accessibility guidelines for designers and developers while in the phase of design and development

- Semantic checks in design, development and testing

Design phase

- Understand the use of semiotics during research phase for permanent, temporary and situational disabled
- Accessibility documentation while creating design system
- Annotating visual designs using semiotics using UiAccessibilityProtocol

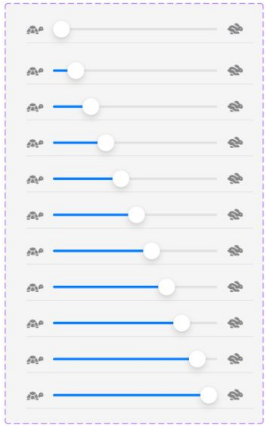
Design system

Component

Sliders

A slider is a horizontal track with a control, called a thumb, that people can adjust between a minimum and maximum value.

https://developer.apple.com/design/human-interface-guidelines/sliders

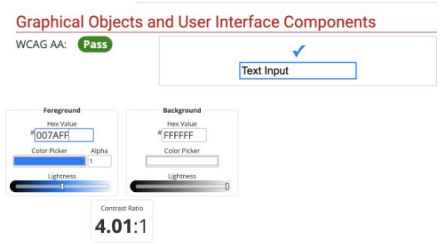


Accessibility

Accessibility documentation

Color, touch target, UI protocol and keyboard access

Color



Touch target

44px X 44px

Properties for voiceover

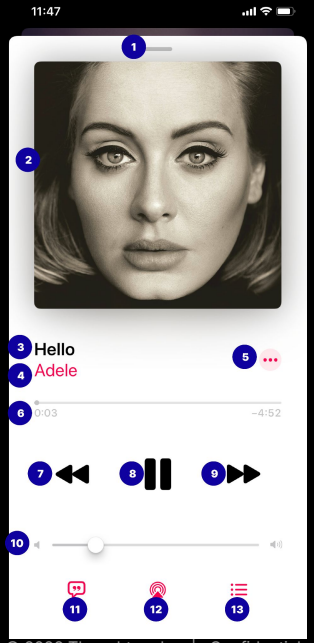
isAccessibilityElement: true
accessibilityLabel: “volume”
accessibilityTraits: “UIAccessibilityTraitsAdjustable”
accessibilityValue: “35%”
accessibilityHint: “Swipe up and down to adjust the volume”

Keyboard access

Move forward - tab
To activate - space
To go Home - Fn H

Design annotation

Syntax (focus order)



Pragmatic

accessibilityHint: “Swipe up and down to adjust the volume”

accessibilityLabel: “Volume”

Semantic of a component

isAccessibilityElement: true

accessibilityLabel: “volume”

accessibilityTraits: “UIAccessibilityTraitsAdjustable”

accessibilityValue: “35%”

accessibilityHint: “Swipe up and down to adjust the volume”

Accessibility traits

- Button, Link, Image
- Selected
- Tab bar
- Toggle
- Search field
- Header
- Summary element
- Adjustable
- Not enabled

Development phase

- Use of annotated protocols in swift

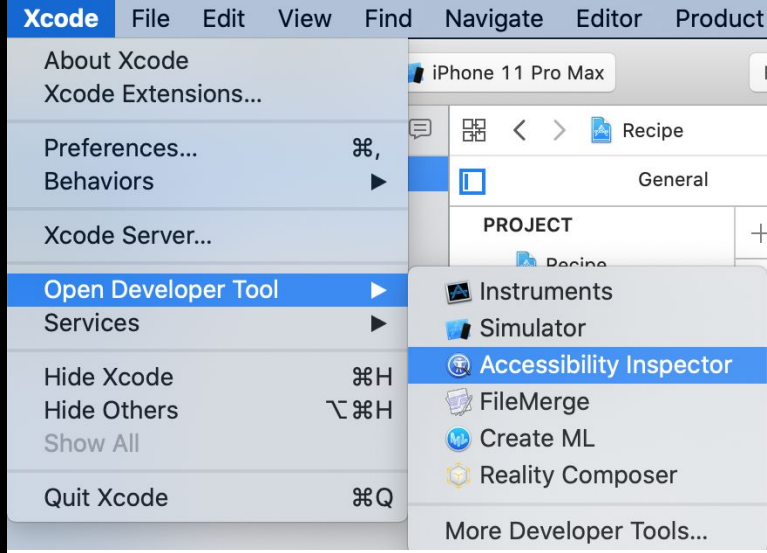
Testing phase

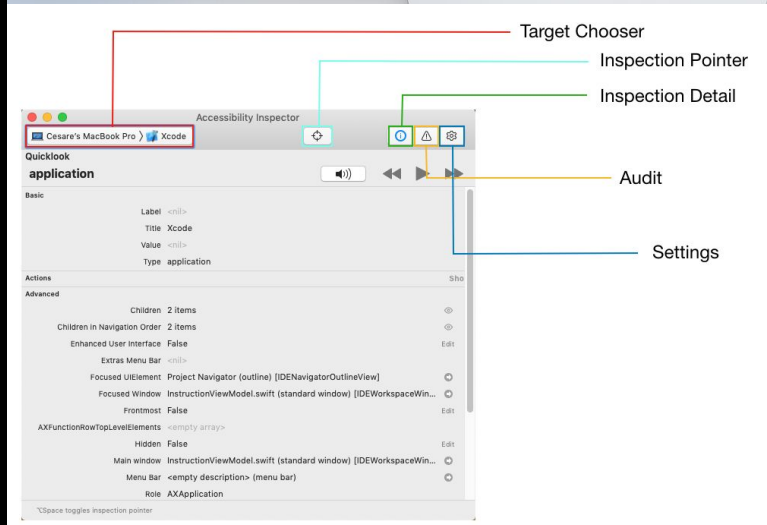
Users

- Involving users to check the semantics

Tools

- Xcode accessibility inspector





Web

Access information + recognise pattern + generate meaning + interpret from previous knowledge **=**
Accessible language

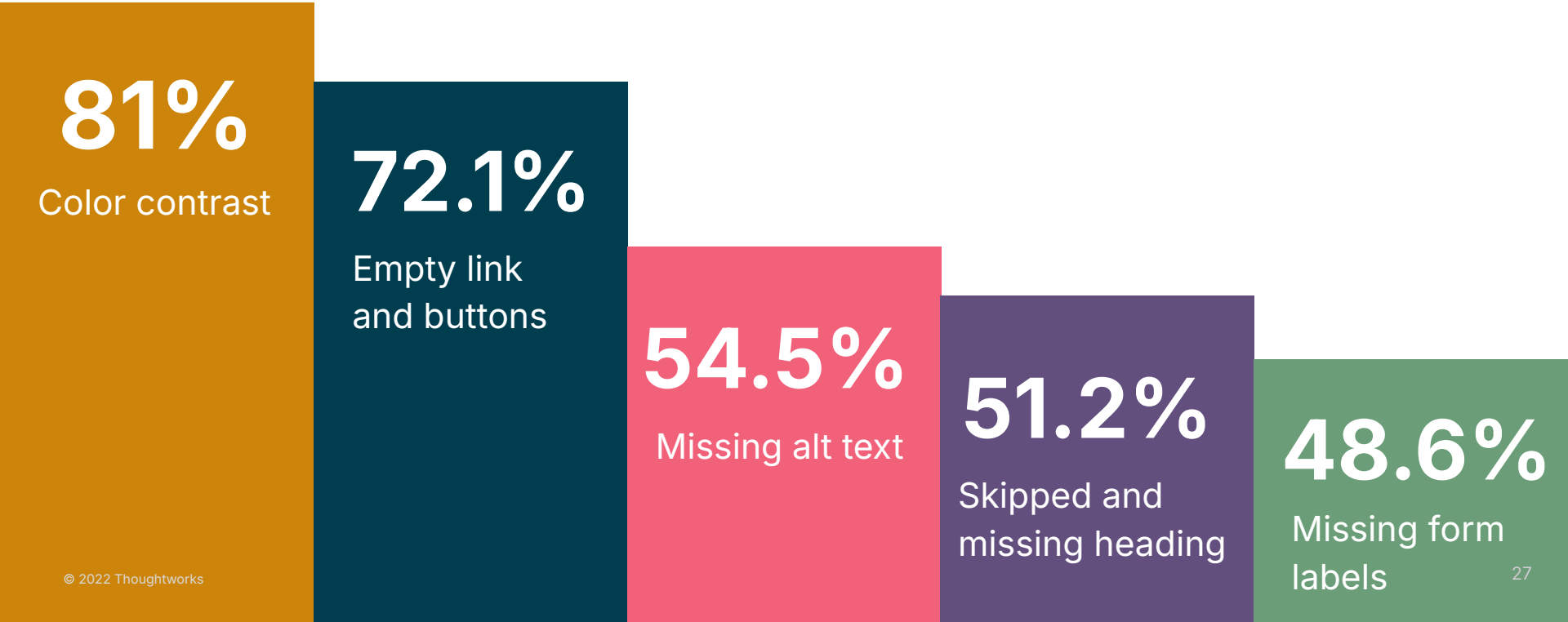
Demo

Bad news

95.9% of home pages had detected WCAG 2.0 failures in 2024 by WebAIM survey with millions of top grossing websites

Design focused A11y issue

WebAIM 2024 survey



WebAIM Survey

For the sixth consecutive year, WebAIM conducted an accessibility evaluation of the home pages for the top 1,000,000 web sites with the help of WAVE Stand-alone API and Testing Engine

Tool	Website	Tranco ranking
The WAVE accessibility engine was used to analyze the rendered DOM of all pages after scripting and styles were applied. WAVE detects end-user accessibility barriers and WCAG conformance failures.	The million home page list was derived from the Tranco ranking	https://tranco-list.eu/ Google, facebook, amazonaws, yahoo, youtube, microsoft, apple etc

Baymard research data

94% of the Largest E-Commerce Sites Are Not Accessibility Compliant. 33 top-grossing e-commerce sites against 4 core accessibility guidelines reveals that 94% of sites are incompliant (WCAG 2.1 AA):

	Images	Links	Form Fields	Keyboard Nav
Home Depot	✗	✗	✗	✗
Argos	✗	✗	✓	✗
Crate & Barrel	✗	✗	✗	✗
Ann Taylor	✗	✗	✗	✗
John Lewis	✗	✗	✗	✗
Wayfair	✗	✗	✗	✗
Macys	✗	✗	✗	✗
Walgreens	✗	✗	✓	✗
Nordstrom	✗	✗	✗	✗
H&M	✓	✓	✓	✓
Victoria's Secret	✗	✓	✓	✗
Disney	✗	✗	✓	✗

	Images	Links	Form Fields	Keyboard Nav
Office Depot	✗	✗	✓	✗
Best Buy	✗	✓	✓	✓
Sears	✗	✗	✗	✗
Northern Tools	✗	✗	✗	✗
Kohls	✗	✗	✗	✗
Williams-Sonoma	✗	✗	✗	✗
Lowes	✗	✗	✗	✗
REI	✓	✓	✓	✓
Apple	✗	✓	✓	✓
ASOS	✓	✗	✓	✓
B&H Photo	✗	✗	✓	✓
Ikea	✗	✓	✓	✗

	Images	Links	Form Fields	Keyboard Nav
Crutchfield	✗	✓	✗	✗
Nike	✗	✗	*	✗
Sephora	✓	✗	✗	✗
Zalando	✗	✗	✗	✓
Adidas	✗	✗	✗	✓
Walmart	✓	✗	✗	✓
Target	✓	✓	✗	✓
Amazon	✗	✗	✓	✓
Staples	✗	✓	✓	✓

*: Unable to rate

82% of sites have accessibility-compliance issues with images

73% of sites have accessibility-compliance issues with links

58% of sites have accessibility-compliance issues with form field markup

64% of sites have accessibility-compliance issues with keyboard navigation

67%

Of WCAG criteria can be covered during the design phase

Step 1: Annotate design before dev hand off

Demo

Is annotating sufficient?

Color contrast, missing link and button, alt text, skipped heading and form label

Step 2: Documenting keyboard and screen reader accessibility

keyboard accessibility (syntax)

Keyboard accessibility

Focus indicators

A sighted keyboard user must be provided with a visual indicator of the element that currently has keyboard focus

Navigation order

The default keyboard navigation order must be logical and intuitive

Lengthy navigation

Provide a "skip to main content" link on the page.
Use a proper heading structure.
Provide regions or ARIA landmarks

Inaccessible custom widgets

All custom controls must still be accessible to keyboard users

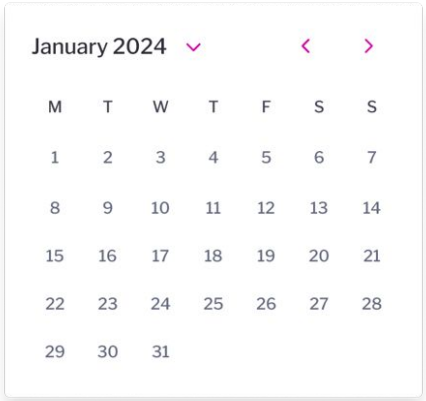
"Tab" is not the only key for keyboard accessibility

		return to the element that opened the dialog.
Slider	• ↑/↓ or ←/→ - increase or decrease slider value • Home/End - beginning or end	• For double-headed sliders (to set a range), Tab/Shift + Tab should toggle between each end. • In some sliders PageUp/PageDown can move by a larger increment (e.g., by 10%)
Menu bar	• ↑/↓ - previous/next menu option • Enter - expand the menu (optional) and select an option. • ←/→ - expand/collapse submenu	A menu bar dynamically changes content within an application. Links that utilize Tab/Enter are NOT menu bars.
Tab panel	• Tab - once to navigate into the group of tabs and once to navigate out of the group of tabs • ↑/↓ or ←/→ - choose and activate previous/next tab.	This is for 'application' tabs that dynamically change content within the tab panel. If a menu looks like a group of tabs, but is actually a group of links to different pages, Tab and Enter are more appropriate.
'Tree' menu	• ↑/↓ - navigate previous/next menu option • ←/→ - expand/collapse submenu, move up/down one level.	
Scroll	• ↑/↓ - scroll vertically • ←/→ - scroll horizontally • Spacebar/Shift + Spacebar - scroll by page	The space bar will, by default, scroll the page, but only if an interactive control that allows space bar input is not focused. Horizontal scrolling within the page should be minimized.

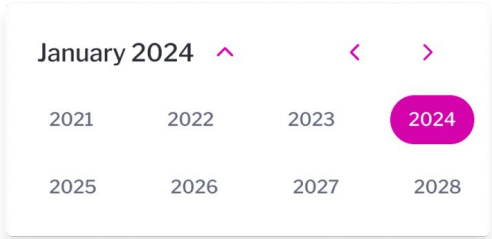
Step 2.1: How to document the keyboard navigation for components?

Date picker

Calendar icon



Date picker dialog



Month, year and date

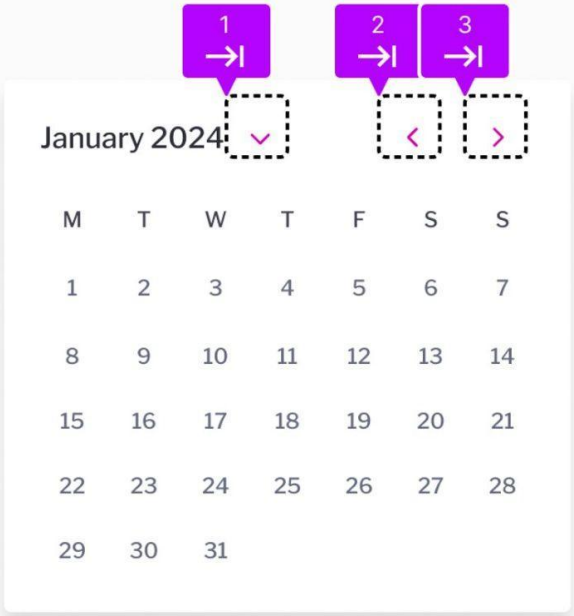
Date Picker icon

`Space, Enter`

- Open the date picker dialog.
- Move focus to selected date, i.e., the date displayed. If no date has been selected, places focus on the current date.

`Esc`

- Closes the dialog and returns focus to the "Choose Date" button.



Date Picker Dialog

Tab

- Moves focus to next element in the dialog Tab sequence.

Shift + Tab

- Moves focus to previous element in the dialog Tab sequence.

Date Picker Dialog: Date Grid

Tab

- Moves focus to next element in the dialog Tab sequence.

Shift + Tab

- Moves focus to previous element in the dialog Tab sequence.

Up arrow

- Moves focus to the same day of the previous week

Down Arrow

- Moves focus to the same day of the next week

Home

- Moves focus to the first day (e.g Sunday) of the current week.

End

- Moves focus to the last day (e.g. Saturday) of the current week

Page Up…

- Changes the grid of dates to the previous month.
- Moves focus to the day of the month that has the same number. If that day does not exist, moves focus to the last day of the month.

Page down

- Changes the grid of dates to the next month.
- Moves focus to the day of the month that has the same number. If that day does not exist, moves focus to the last day of the month.

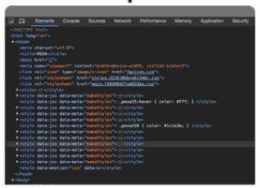
Shift + Page Down

- Changes the grid of dates to the same month in the next year.
- Moves focus to the day of the month that has the same number. If that day does not exist, moves focus to the last day of the month.

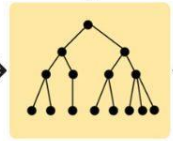
Shift + Page up

- Changes the grid of dates to the same month in the previous year.
- Moves focus to the day of the month that has the same number. If that day does not exist, moves focus to the last day of the month.

Step 2.2: Semantics (screen reader accessibility)

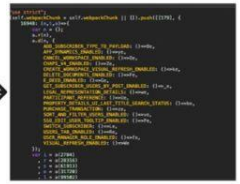


HTML



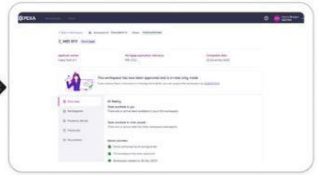
DOM tree
(Document object model tree)

read/modify via DOM API

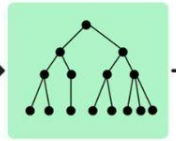


Js

Painting into visual UI



Transformed into
accessibility tree



Accessibility tree

Assistive technology

User

Accessibility tree will only respond when components have semantics

- Native HTML or ARIA

Accessible Rich Internet Application

ARIA is a way of taking dynamic interaction from the browser such as what you do with java script and HTML, making them semantically accessible for AT. "It's a semantic meta language works for all AT"

Nature of ARIA

It's a group of properties that you can attach to HTML to give it more meaning

Role

What the element is and what function it supposed to serve in the page? Sometimes it's obvious and sometimes it's difficult

State

How you are going interact with the elements?

For example: Is the element checked?

Is the element disabled?

Properties

Define purpose or relationship of HTML elements in the accessibility tree
Example: does the element have a description?

The role: What am I?

Implicit:
Attach to know element. Part of the HTML specification. For example: <h1> <button>

<h1>

Heading

</h1>

***You don't need aria-level1**

Explicit

Something doesn't exist in the HTML. Using aria you can give a specific role to the element.

What am I?

- It doesn't act like a check box for assistive technology
- It doesn't respond to the keyboard
- It might look like check box as you have designed it that way but it doesn't work that way

Although, you can do this way but you might have to write loads of java script to handle native keyboard interaction

State: What am I doing now?

Without aria state information

<button id="menu-toggle">

Toggle Menu
</button>

It doesn't tell you whether or not menu is open

With aria state information

<button id="menu-toggle

aria- expanded="true "
>
Toggle Menu
</button>

AT users able to understand whether it's available for them or not

Properties:

Who am I related to and what do you need to know about me?

```
<button id="menu-toggle
aria- expanded="true"
aria-controls="menu-main-menu"
>
Menu
</button>
```

Now this button knows not only it's open but exactly what it is opening. You can go farther and add more information.

```
<button id="menu-toggle
aria- expanded="true"
aria-controls="menu-main-menu"
aria-label="close Main Navigation Menu"
>
Menu
</button>
```

*aria-label substitute the text of the button and call out for AT users.

"No ARIA is better than Bad ARIA"

Role	Attribute	Element	Usage
dialog			Identifies the element that serves as the dialog container.
	aria-labelledby=""		Gives the dialog an accessible name by referring to the element that provides the dialog title.
	aria-describedby=""		• Gives the dialog an accessible description by referring to the dialog content that describes the primary message or purpose of the dialog
	aria-modal="true"		Tells assistive technologies that the windows underneath the current dialog are not available for interaction.

ARIA APG

Follow authoring practice guide for all the components

Clear semantics for assistive technology

- Role
- State
- Properties

Use landmarks

- Header
- Footer
- Main
- Nav

Don't use ARIA unnecessary

If the semantics of component is clear enough with HTML 5, ARIA is not required

Access of info + keyboard (syntax) + pragmatic (screen reader) = Web for everyone

Thank you

Nilmani Kumar
UX Designer
neil.mani@thoughtworks.com

/thoughtworks