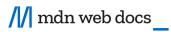
MDN Plus now available in <u>your</u> country! Support MDN <u>and</u> make it your own. Learn more <sup>⁺</sup>→



X

# Mobile accessibility checklist

This document provides a concise checklist of accessibility requirements for mobile app developers. It is intended to continuously evolve as more patterns arise.

### Color

- Color contrast must comply with <u>WCAG 2.1 AA level requirements</u> 

   ☐:
  - Contrast ratio of 4.5:1 for normal text (less than 18 point or 14 point bold.)
  - Contrast ratio of 3:1 for large text (at least 18 point or 14 point bold.)
- Information conveyed via color must be also available by other means too (underlined text for links, etc.)

# Visibility

- Content hiding techniques such as zero opacity, z-index order and off-screen placement must not be used exclusively to handle visibility.
- Everything other than the currently visible screen must be *truly* invisible (especially relevant for single page apps with multiple *cards*):
  - Use the hidden attribute or visibility or display style properties.
  - Unless absolutely unavoidable, aria-hidden attribute should not be used.

### **Focus**

- All activatable elements must be focusable:
  - Standard controls such as links, buttons, and form fields are focusable by default.

- Non-standard controls must have an appropriate <u>ARIA Role</u> assigned to them, such as button, link, or checkbox.
- Focus should be handled in a logical order and consistent manner.

## Text equivalents

- Text equivalent must be provided for every non-strictly presentational non-text element within the app.
  - Use alt and title where appropriate (read Steve Faulkner's post about <u>Using the</u>
     <u>HTML title attribute</u> ☑ for a good guide.)
  - If the above attributes are not applicable, use appropriate <u>ARIA States and</u>
     <u>Properties</u> ☑ such as aria-label, aria-labelledby, or aria-describedby.
- Images of text must be avoided.
- All user interface components with visible text (or image of text) as labels must have
  the same text available in the programmatic <u>name</u> ☑ of the component. <u>WCAG 2.1:</u>
  Label in name. ☑
- All form controls must have labels (<<u>label></u> elements) for the benefit of screen reader users.

## Handling state

• Standard controls such as radio buttons and checkboxes are handled by the operating system. However, for other custom controls state changes must be provided via <a href="#">ARIA</a>
<a href="#">States</a> <a href="#">States</a> <a href="#">Such as aria-checked</a>, aria-disabled, aria-selected</a>, aria-expanded, and aria-pressed.

## Orientation

- Content should not be restricted to a single orientation, such as portrait or landscape, unless essential. WCAG 2.1: Orientation ☑
  - Examples of when an orientation is essential is a piano application or a bank check.

# General guidelines

• An app title must be provided.

· Headings must not break hierarchical structure

```
<h1>Top level heading</h1>
<h2>Secondary heading</h2>
<h2>Another secondary heading</h2>
<h3>Low level heading</h3>
```

- <u>ARIA Landmark Roles</u> ☑ should be used to describe an app or document structure, such as banner, complementary, contentinfo, main, navigation, search.
- For touch events, at least one of the following must be true (<u>WCAG 2.1: Pointer</u>
   <u>Cancellation</u> ☑):
  - The down-event should not be used to trigger any action
  - The action is triggered on the up event and an option to abort the action before its completion is available or an option to undo the action after its completion
  - The up-event will undo any action that was triggered on a down event
  - It is essential to trigger the action on the down event. For example, playing a game or a piano application.
- Touch targets must be large enough for the user to interact with (see the <u>BBC Mobile</u>
   <u>Accessibility Guidelines</u> ☑ for useful touch target size guidelines).
- **Note:** The <u>original version of this document</u> ☑ was written by <u>Yura Zenevich</u> ☑.

Last modified: Sep 28, 2022, by MDN contributors