

Summary and Checklist: Form Labels, Instructions, and Validation

Summary

In order for users to know how to fill out a form, the form has to be accessible. For the most part this is easy to do. Key concepts include:

- Labels for form inputs
- Labels for groups of inputs
- Instructions and hints, where necessary
- Error prevention
- Form validation

With all of these, information must be visible on the screen, accurate and meaningful, programmatically determinable, and programmatically associated with the appropriate form element or group.

The more interactive a form element or process is, the more attention needs to be given to accessibility with respect to:

- Focus management
- Setting and updating ARIA names, roles, and values, where necessary
- ARIA live announcements, where necessary

Checklist

Labels

Semantic Labels

- Labels MUST be programmatically associated with their corresponding elements.
- Labels MUST be programmatically determinable.

Meaningful Label Text

- Labels MUST be meaningful.
- Labels MUST NOT rely solely on references to sensory characteristics.

Icons as Labels

- Icons MAY be used as visual labels (without visual text) if the meaning of the icon is visually self-evident AND if there is a programmatically associated semantic label available to assistive technologies.

Placeholder Text as Labels

- Placeholder text MUST NOT be used as the only method of providing a label for a text input.

Visibility of Labels

- Labels MUST be visible.
- For user interface components with labels that include text or images of text, the name MUST contain the text that is presented visually.

Proximity of Labels to Controls

- A label SHOULD be visually adjacent to its corresponding element.
- A label SHOULD be adjacent in the DOM to its corresponding element.

Multiple Labels for One Field

- When multiple labels are used for one element, each label MUST be programmatically associated with the corresponding element.

One Label for Multiple Fields

- When one label is used for multiple elements, the label MUST be programmatically associated with each of the corresponding elements.

Group Labels

Semantic Group Labels

- Group labels MUST be programmatically associated with the group if the individual labels for each element in the group are insufficient on their own.
- Group labels MUST be programmatically determinable.

Meaningful Group Labels

- Group labels MUST be meaningful.
- Group labels MUST NOT rely solely on references to sensory characteristics.

Proximity of Group Labels

- Group labels SHOULD be visually adjacent to the grouped elements.
- Group labels SHOULD be adjacent in the DOM to the grouped elements.

Visibility of Group Labels

- Group labels MUST be visible.

Summary and Checklist: Form Labels, Instructions, and Validation

Instructions & Other Helpful Info

Instructions for Forms, Groups, and Sections

- Instructions for groups or sections SHOULD be programmatically associated with the group.
- Instructions for groups or sections MUST be programmatically determinable.
- Instructions for groups or sections MUST be meaningful.
- Instructions for groups or sections MUST be visible.
- Instructions for groups or sections SHOULD be visually adjacent to the grouped elements.
- Instructions for groups or sections SHOULD be adjacent in the DOM to the grouped elements.
- If the instructions for groups or sections are not critical, the instructions MAY be hidden until the user requests them.
- Instructions for groups or sections MUST NOT rely solely on references to sensory characteristics.

Instructions for Inputs

- Instructions for an element MUST be programmatically associated with the element.
- Instructions for an element MUST be available as programmatically determinable text.
- Instructions for an element MUST be meaningful.
- Instructions for an element MUST be visible.
- Instructions for an element SHOULD be visually adjacent to the element.
- Instructions for an element SHOULD be adjacent in the DOM to the element.
- If the instructions for an element are not critical, the instructions MAY be hidden until the user requests them.
- Instructions for an element MUST NOT rely solely on references to sensory characteristics.

Required Fields

- Required fields SHOULD be programmatically designated as such.
- Required fields SHOULD have a visual indicator that the field is required.
- The form validation process MUST include an error message explaining that a field is required if the field isn't identified as required both visually and programmatically in the form's initial state.

Input Purpose

- The purpose for each common input field that collects an individual's personal data MUST be programmatically defined based on the list of 53 Input Purposes for User Interface Components.

Dynamic Forms & Custom Widgets

Changes in Context

- Focusing on an element MUST NOT automatically trigger a change of context, unless the user has been adequately advised ahead of time.
- Changing an element's value MUST NOT automatically trigger a change of context, unless the user is adequately advised ahead of time.
- Hovering over an element with the mouse MUST NOT automatically trigger a change of context, unless the user has been adequately advised ahead of time.

Custom Form Inputs

- Native HTML form elements SHOULD be used whenever possible.
- Custom form elements SHOULD act like native HTML form elements, to the extent possible.
- Custom form elements SHOULD have appropriate names, roles, and values.
- Updates and state changes that cannot be communicated through HTML or ARIA methods SHOULD be communicated via ARIA live messages.

Form Validation

Error Identification Considerations

- Error feedback SHOULD be made available immediately after form submission (or after an equivalent event if there is no form submission event).
- Error feedback MUST be programmatically associated with the appropriate element.
- Error feedback MUST be programmatically determinable.
- Error feedback MUST be meaningful.
- Error feedback MUST be visible.

Form Validation (Continued)

Success Confirmation Considerations

- Success confirmation feedback SHOULD be programmatically determinable.
- Success confirmation feedback SHOULD be meaningful.
- Success confirmation feedback MUST be visible.

Accessible Authentication

- **[Added in WCAG 2.2]** A cognitive function test MUST NOT be required for any step in an authentication process, unless specified criteria are met.
- **[Added in WCAG 2.2]** A cognitive function test SHOULD NOT be required for any step in an authentication process, unless there is an alternative method without a required cognitive test or there is a mechanism to assist the user.

Redundant Entry

- **[Added in WCAG 2.2]** Information previously entered by or provided to the user that is required to be entered again in the same process MUST be either auto-populated or available for the user to select, with several allowable exceptions.