Creating Accessible Labels

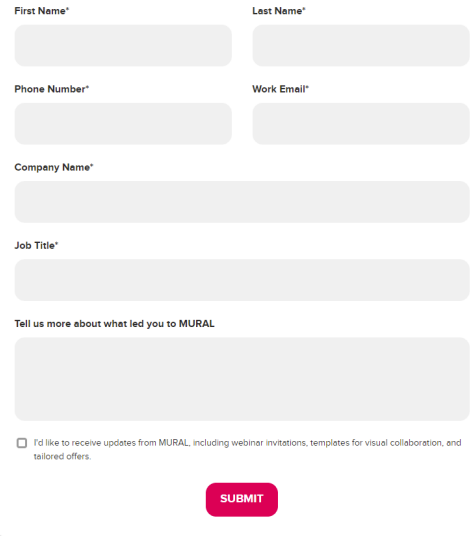
In this module, you'll learn:

- What a label is.
- Characteristics of accessible labels.
- Why accessible labels are important.
- How to ensure the labels in your design are accessible.
- Best practices for creating accessible labels, in a nutshell.

What is a label?

A label is a form of instruction in the interface that helps the end user understand an element and how to interact with it.

For example, the contact form on MURAL's website includes labels that guide the end user to type in their first name, last name, phone number, work email, company name, and job title.



Labels help end users understand:

- Where they are on a webpage.
- What action they need to take.
- How information is organized.

Labels are typically applied to elements such as menus, tags, buttons, icons, text fields, checkboxes, radio buttons, and dropdown menus.

What are the characteristics of accessible labels?

Labels provide descriptions of **inputs** and **form controls** to help end users fill out or submit a form, perform a search, copy a link, enter an email address, and much more.

What's that mean?

Input: A field used to obtain a response from an end-user.

- Examples: Textboxes, search boxes, address bars, amount and date fields, etc.

Form controls: Elements within forms that allow user interactions.

- Examples: Submit buttons, dropdown boxes, radio buttons, checkboxes, etc.

For a label to be considered accessible, it should:

- Always appear in a consistent location so that an end user can easily read and predict the content on the page.
- Be explicit and adjacent to the element it describes.
- Always show up! Don't require the end-user to hover or focus to see it.
- Differentiate table headers and footers from table data.

Why are accessible labels important?

Confusing inputs and forms cause frustration for everyone. Labels create a much better experience for users with and without disabilities by:

- Helping all end users to easily navigate, read, and understand the structure of content and purpose of inputs.
- Providing screen reader users information on what each input is.
- Helping low-vision users focus on relevant content.

How do I ensure my labels are accessible?

As a MURAL designer creating accessible labels, your top priorities are to:

- Include descriptive labels for all inputs (Please note that the content design team is developing guidelines for the tone of voice and the style of language that should be used at MURAL. More information to come).
- Consistently place labels near inputs.
- Visually display labels whenever possible and ensure they're accessible to screen readers.

The tips below will help you design accessible labels for MURAL.

Ensure all inputs have labels

All inputs and form fields should have visible labels whenever possible, including within multiple related fields. For example, there may be two different form fields for a phone number. One field may require the area code and the other the phone number itself.

A visible label is needed for each field as well as a screen reader-accessible label. We'll cover labels for screen readers in another section of the module.

While icons aren't form controls, they also need labels. Without labels, end users have to rely on the icon itself to understand it.

Keep labels brief and straightforward

Your labels should be descriptive yet succinct. A clear and concise label is descriptive enough that a user can quickly understand the purpose and function of an input.

Use persistent, adjacent, and explicit labels

What's that mean?

Persistent labels: Labels that are always visible.

Adjacent labels: Labels that are near or next to the input they describe.

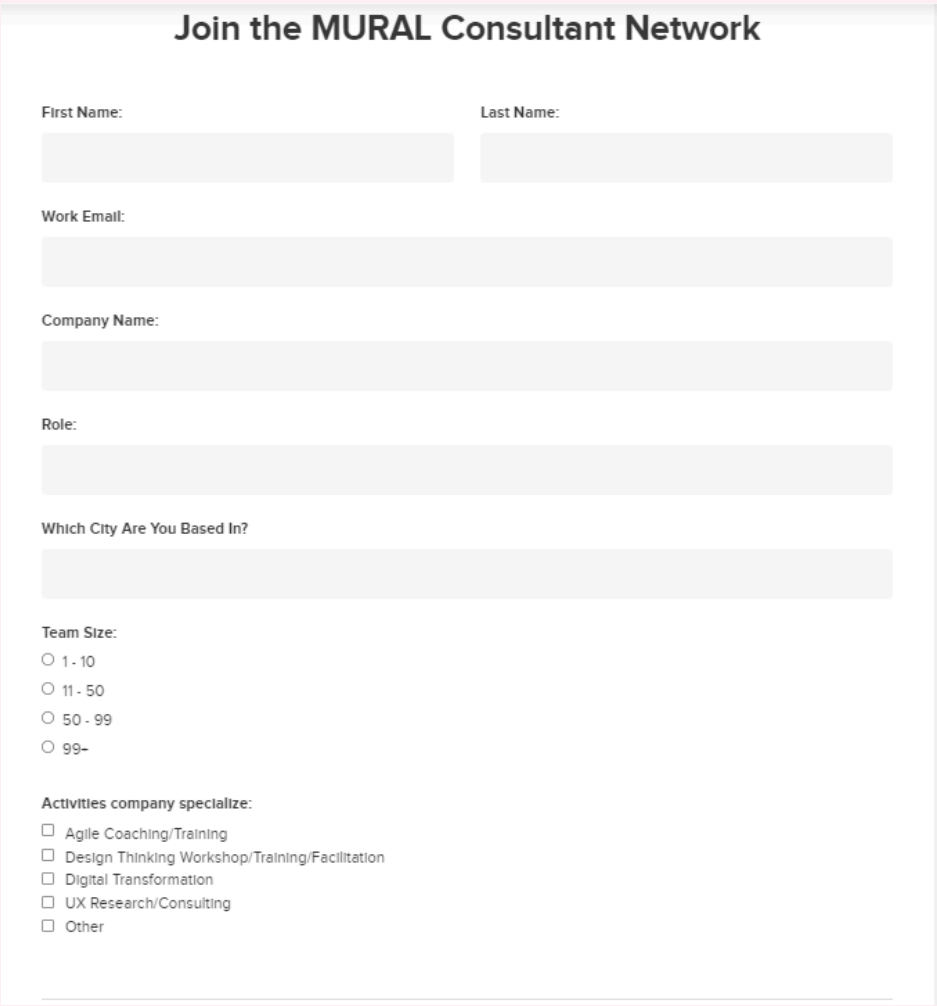
Explicit labels: Labels that are recognizable to assistive technology.

- **Ensure inputs have persistent labels.** Labels that are always displayed don't require an end user to hover or focus on an element to understand its intention.

- **Add accessible labels, such as ARIA (Accessible Rich Internet Applications) labels, to inputs without visible labels.**
    - Icons next to visible labels don't need an accessible label. They're considered decorative. For example, if a form field is labeled "Search" the magnifier icon doesn't need a label. In this case, you should include an aria-label on the form field to indicate its purpose to screen readers.

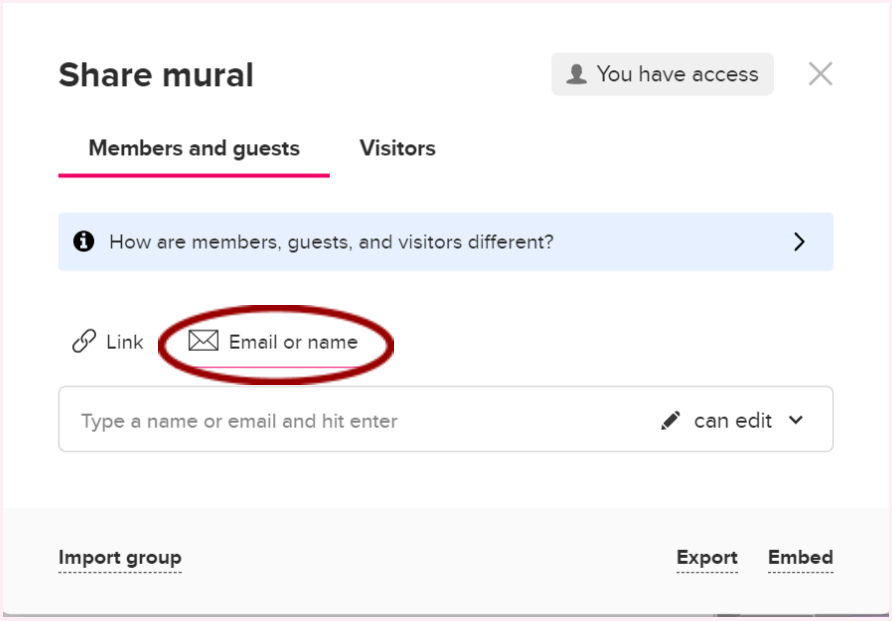
Example: Those who want to join the MURAL Consultant Network are asked to fill out the form below. The form includes labels to type in personal information such as name, email, company, and role.

It also provides radio buttons to select your team's size and checkboxes to describe your company's specialization. The size and specialization categories and their respective choices all include labels.



- **Ensure labels always appear near or adjacent to (next to) the element they describe.** This helps an end user associate the label with the input (e.g., a form control) and understand its purpose.

Example: In the **Share mural** option within MURAL, designers created an adjacent label for the **Email or name** box by labeling it directly above where end user input is expected.



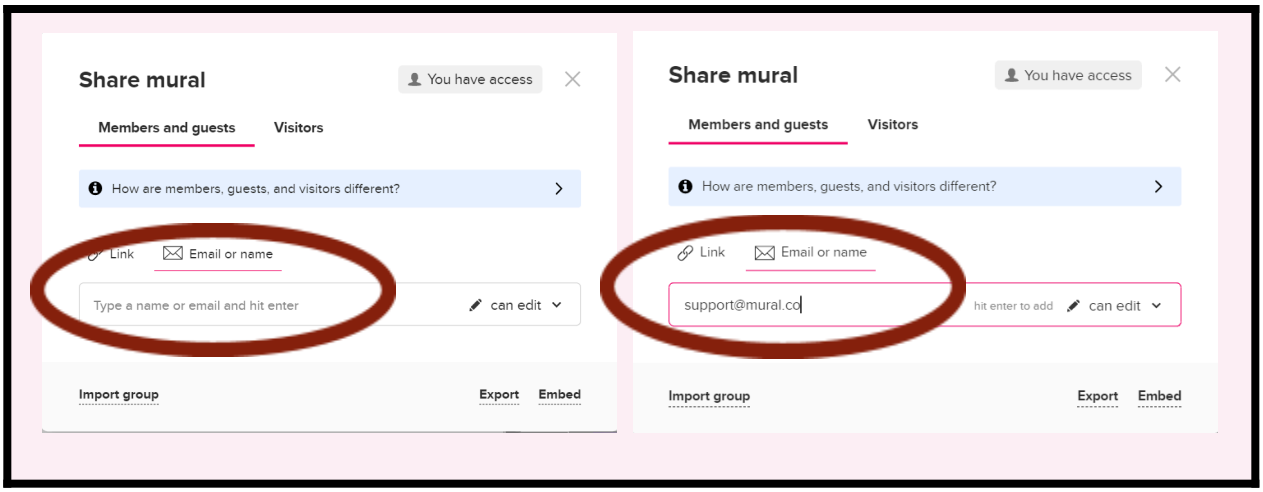
- **Make labels explicit so that screen readers can announce them.** An explicit label means an end user can understand an input by its label alone or by the context of the label.
    - You can do this by providing the form control an ID attribute and adding a FOR attribute to the label with the value of that ID. This helps developers make sure all labels are properly associated with the correct input.

Don’t replace labels or crucial information with placeholder text

Disappearing placeholder text is essentially the same as a missing label or instruction. Placeholder text is problematic because:

- It disappears from the textbox or form field once the end user enters information.
- It forces the end user to remember what action is required, creating additional cognitive load.
- It may not be recognized by screen readers.

Example: The **Share mural** feature discussed in the previous example, included the **Email or name** label as well as placeholder text instructing you to **Type a name or email and hit enter**. As shown in the screenshots below, the placeholder text disappears once you enter your own text. However, the label remains, reminding you of the purpose of the textbox.



Provide instructions in the UI for unique keyboard operations or when labels aren't sufficient

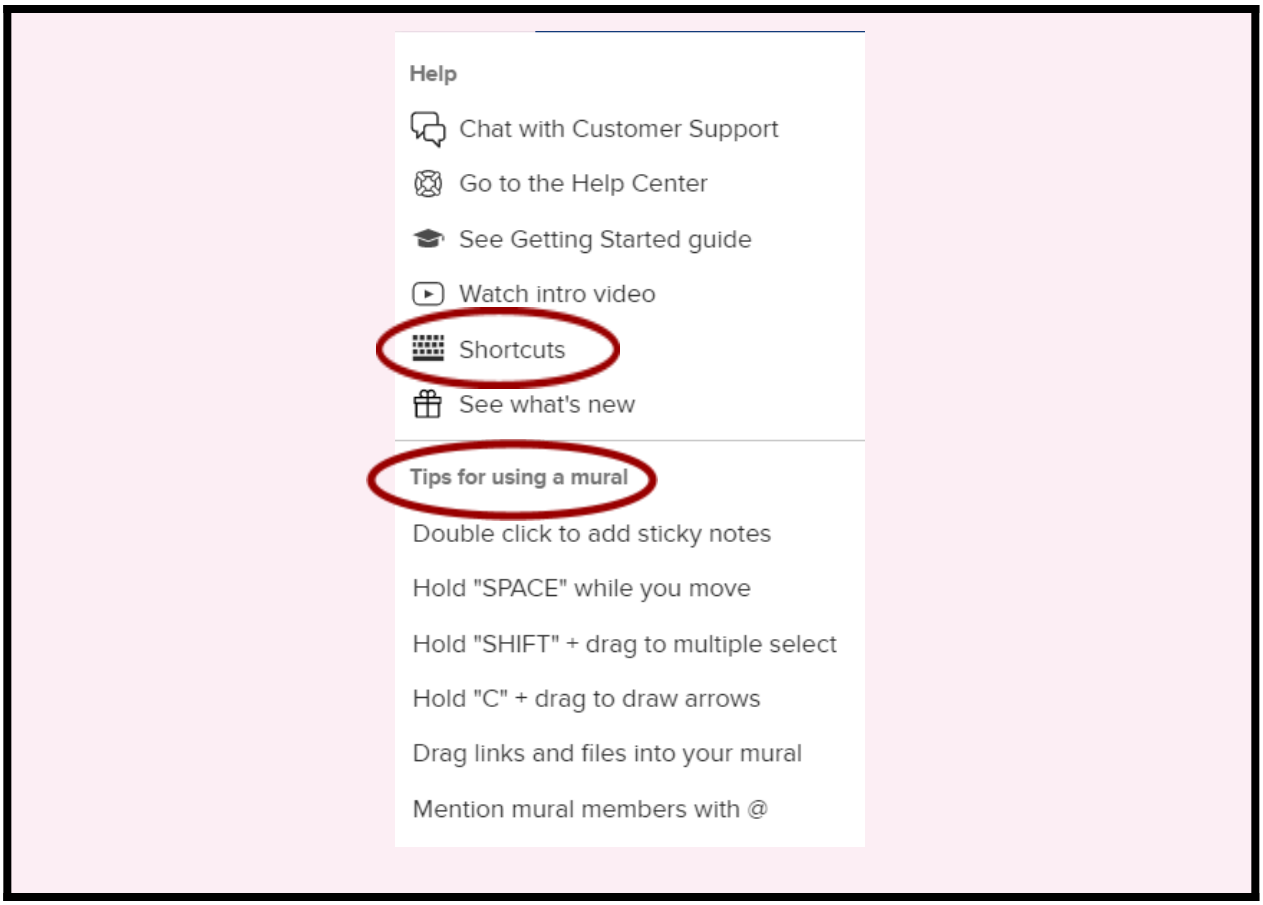
Labels may not include all the information an end user needs. In these cases, designers often include additional context on hover or focus. This can be problematic because:

- Keyboard users may not have access to content through hover or focus.
- The information often disappears when not using hover or focus, creating challenges for people with cognitive impairments who find it difficult to remember.

Consider ways to provide persistent descriptions, instructions, and additional context when labels aren't enough:

- **Provide visible instructions for a keyboard help shortcut.**

Example: In MURAL's **Help menu**, you can find keyboard shortcuts and tips for using MURAL, as shown in the screenshot below.

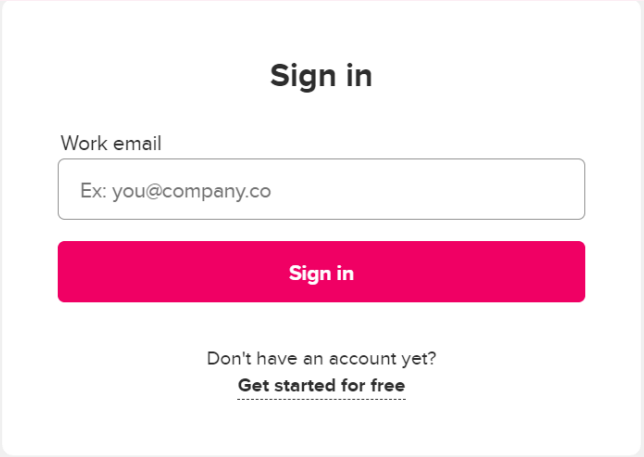


- **Provide instructions inside textboxes in addition to labels** (e.g., find any word or phrase, search, etc.)**.**
- **Display formatting expectations** (e.g., whether an end user needs to use dashes or slashes when entering a birthdate).
- **Identify required fields.** This can often be done by using an asterisk.
- **Design robust error validation.** Whether or not placeholder text is part of a design, error validation helps inform an end user if they typed in the wrong information, skipped a field, or used the wrong format.

Resource: Error validation is an important part of the labeling process. For more information, see the module on **Accessible Error Messages**.

- **Include instructions near the input.** Just as you would place a label near an input, provide adjacent instructions so that it's obvious which input the instructions are for.

Example: The label in the screenshot below indicates that a work email is required to sign in to MURAL. The button below the form is task-oriented, instructing the end user to sign in. Formatting expectations for the email are within the form field.



Label table headers and footers

Table elements need labels to be properly understood by people who use assistive technologies, such as screen readers. Screen readers usually include table navigation commands and will announce column and row headers to help users understand the data.

Ensure your tables are labeled and accessible:

- **Use HTML markup for table elements. Designate row and column headers.**
    - The <td> element is used for table data cells, and the <th> element is used for table header cells.
- **Include ARIA labels for data in columns, rows, and cells.**
- **Provide a brief caption immediately before the table.**

Use an accessible tooltip when persistent labels aren't possible

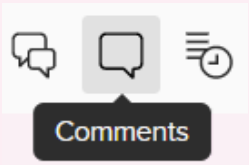
What's that mean?

Tooltip: An informative message or content that appears when a user interacts with an element.

Tooltips should never replace labels. However, they're useful if persistent labels aren't possible like in the case of compact UIs. Keep these tips in mind when deciding to use them:

- **Provide descriptive tooltips when icons don't contain text labels.**

Example: In MURAL, tooltips are provided for many menu items to help end users understand the purpose of an icon.



- **Provide tooltips when persistent instructions aren't possible.**
- **Use appropriate color contrast for tooltips and make sure they don't block related content.**

Keep labels consistent

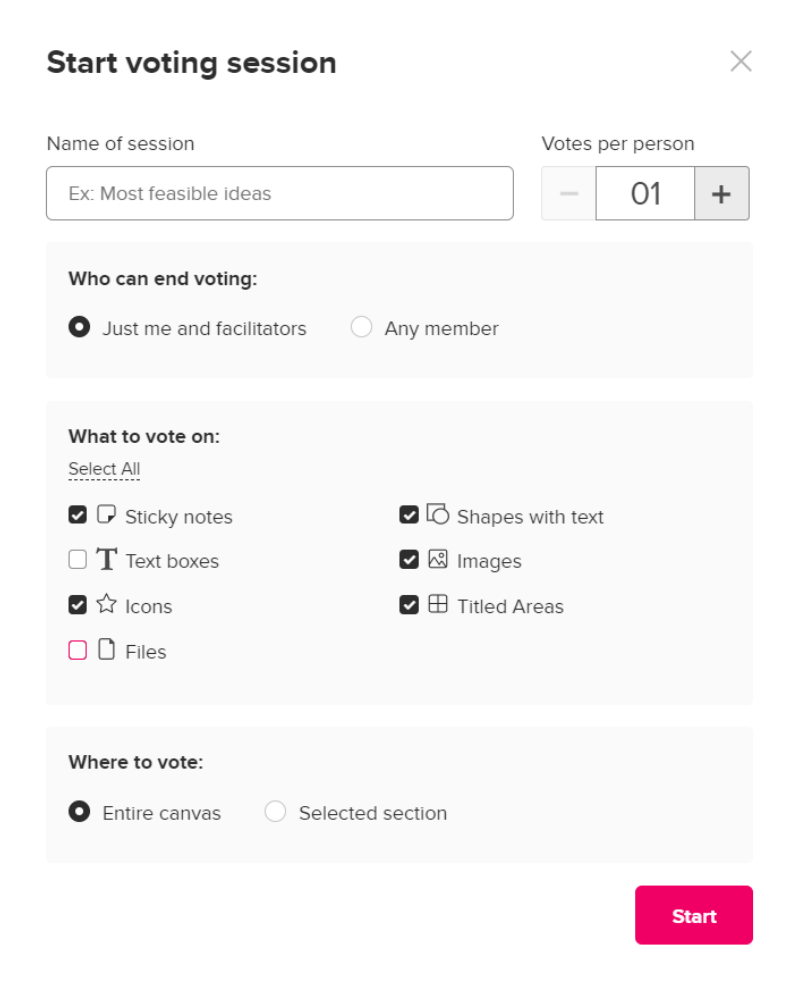
These guidelines can keep your labels consistent:

- **Ensure all inputs have labels and/or instructions.**
- **Avoid using all caps and making labels look like buttons.** This can confuse the end user. If labels can't be clicked they shouldn't look like something that can.
- **Determine whether your design includes format requirements for inputs.**
- **Ensure all form controls have labels.** Examples of form controls that need labels:
    - Text fields.
    - Checkboxes.
    - Radio buttons.
    - Dropdown menus.
- **Determine whether labels and instructions are in expected locations and next to related elements.**
- **Include labels before inputs and form controls.** Radio buttons and checkboxes are the only exceptions to this. In these cases, the label should come after.

Example: When you conduct a voting session in MURAL, a modal pops up to offer various

voting options as seen in the screenshot below.

The radio buttons and checkboxes appear before each of the respective labels to help you decide who can end voting, what to vote on (e.g., sticky notes, images, and any other MURAL objects), and where to vote (e.g., the entire canvas or a selected section).



Best practices, in a nutshell

- **Ensure all inputs have labels.**
- **Keep labels brief and straightforward.**
- **Use persistent, adjacent, and explicit labels.**
- **Don’t replace labels or crucial information with placeholder text.**
- **Provide instructions in the UI for unique keyboard operations or when labels aren’t sufficient.**
- **Label table headers and footers.**
- **Use an accessible tooltip when persistent labels aren’t possible.**

- **Keep labels consistent.**

Test your knowledge

Choose one or more answers to the following questions.

1. Why is placeholder text not an accessible replacement for labels?
    A. Placeholder text disappears when data is entered in the field.
    B. Placeholder text may not be recognized by screen readers.
    C. Placeholder text isn't recognized by most browsers.
    D. All of the above
    E. A and B

2. Which of the following are accessible ways to label form fields or controls?
    A. Place labels inside form fields.
    B. Place labels near the form field it describes.
    C. Place labels before checkboxes.
    D. Avoid using labels for icons.

3. The best way to make a label accessible to a screen reader is to:
    A. Include instructions for the input.
    B. Make sure the label is explicit.
    C. Hide the label.
    D. Place the label on the left side of the input.

4. Which of the following statements about labels are correct:
    A. Labels should be visually close to their corresponding inputs.
    B. Labels should be similar in color to their corresponding form fields.

C. Labels should always include additional instructions.

D. Labels should only appear on hover or focus.

5. True or false: When one visible label is used to label multiple fields, such as the parts of an address, each individual field must be labeled and have a screen reader accessible label.

    A. True.

    B. False.