# Accessibility Checklist for Developers

Adopting accessibility at the outset has been proven to be the most effective way to implement accessible digital products. The purpose of this user story checklist is to help you write accessible code as fluidly and efficiently as possible.

## Leveraging the 10 Essentials for Developers

Using the table below, place a check mark in each row for elements that are included in the user story on which you are currently working. When you are ready to begin developing, use the links in the third column to see coding examples for making the element(s) accessible for assistive technology (AT) users. Note: If you are interested in understanding why and how each element is critical for AT users, use the links in the 10 Essentials Elements column for detailed explanations.

| √ | 10 Essential Elements | Coding Techniques |
| --- | --- | --- |
|  | [Headings and landmarks have correct semantic markup](https://accessibility.huit.harvard.edu/use-semantic-elements-regions-and-content). | * [Use Heading Structure](https://accessibility.huit.harvard.edu/practice-heading-structure) * [Create Main Landmark](https://accessibility.huit.harvard.edu/practice-main-landmark) |
|  | [Page layouts adapt to different window widths and text sizes](https://accessibility.huit.harvard.edu/support-flexibility-and-adaptation). | * [Enable text resizing](https://accessibility.huit.harvard.edu/practice-enable-text-resizing) * [Text Spacing](https://accessibility.huit.harvard.edu/technique-text-spacing) * [Orientation](https://accessibility.huit.harvard.edu/technique-orientation) * [Reflow](https://accessibility.huit.harvard.edu/technique-reflow) |
|  | [Color contrast is sufficient and there is no reliance on color only.](https://accessibility.huit.harvard.edu/avoid-reliance-color) | * [Indicate required fields without reliance on colors](https://accessibility.huit.harvard.edu/avoid-reliance-color) * [Indicate accessible current page without reliance on colors](https://accessibility.huit.harvard.edu/practice-accessible-current-page-indication) |
|  | [Images and icons have the appropriate text alternative](https://accessibility.huit.harvard.edu/provide-accessible-images). | * [Provide icon information to users of assistive technology](https://accessibility.huit.harvard.edu/practice-icon-fonts) * [Provide useful alternative text](https://accessibility.huit.harvard.edu/example-alt-tag) |
|  | [Interactive controls have an accessible descriptive name](https://accessibility.huit.harvard.edu/custom-widgets-and-controls). | * [Provide accessible names for buttons](https://accessibility.huit.harvard.edu/practice-accessible-names-close-buttons) |
|  | [Input element and input groups have descriptive labels](https://accessibility.huit.harvard.edu/provide-accessible-labels-and-instructions). | * [Create descriptive input labels](https://accessibility.huit.harvard.edu/technique-input-labels) * [Identify Input Purpose](https://accessibility.huit.harvard.edu/technique-identify-input-purpose) * [Indicate required fields](https://accessibility.huit.harvard.edu/technique-required-fields) |
|  | [Interactive elements are highlighted upon keyboard focus. Their order follows a logical sequence](https://accessibility.huit.harvard.edu/provide-logical-and-visible-focus-indication). | * [Provide focus order for interactive elements](https://accessibility.huit.harvard.edu/%E2%9C%8E-technique-focus-order-and-tabindex) * [Indicate paragraph link focus](https://accessibility.huit.harvard.edu/technique-paragraph-link-focus-indication) * [Managing focus and inactive elements](https://accessibility.huit.harvard.edu/technique-managing-focus-and-inactive-elements) |
|  | Interactive elements are [selected and activated using the keyboard](https://accessibility.huit.harvard.edu/support-keyboard-interaction). | * [Interaction with added content on hover or focus](https://accessibility.huit.harvard.edu/technique-content-hover-or-focus) * [Ability to turn off or remap character key shortcuts](https://accessibility.huit.harvard.edu/technique-character-key-shortcuts) * [Create accessible modal dialogs](https://accessibility.huit.harvard.edu/practice-basic-modal-dialog) * [Create accessible expandable sections](https://accessibility.huit.harvard.edu/%E2%9C%8E-practice-expandable-sections) * [Create custom controls that are keyboard operable](https://accessibility.huit.harvard.edu/technique-focus-order-and-tabindex) |
|  | [Custom controls and widgets have the correct role (e.g., link, button, tab panel)](https://accessibility.huit.harvard.edu/provide-name-role-and-value-information) | * [Provide accessible label in name](https://accessibility.huit.harvard.edu/technique-label-name) * [Autocomplete input controls](https://accessibility.huit.harvard.edu/technique-aria-autocomplete) * [Indicate required fields](https://accessibility.huit.harvard.edu/technique-required-fields) |
|  | [Dynamic updates, including error messages, are conveyed to assistive technology](https://accessibility.huit.harvard.edu/provide-notification-dynamic-changes-content). | * [Form feedback with live regions](https://accessibility.huit.harvard.edu/technique-form-feedback-live-regions) * [Status message](https://accessibility.huit.harvard.edu/technique-status-messages) |

## Testing

Test the accessibility of your code, whether a portion or an entire user interface.

| √ | Testing Objectives | Testing Techniques & Tools |
| --- | --- | --- |
|  | Navigate through the interface and complete interface tasks using keyboard only | Navigate, access content, and complete tasks just with the Tab, Enter, and arrow keys |
|  | Navigate through the interface and complete interface tasks using a screen reader | [Free screen readers](https://usabilitygeek.com/10-free-screen-reader-blind-visually-impaired-users/) |
|  | Verify appropriate color contrast | [WebAIM Color Contrast Checker](https://webaim.org/resources/contrastchecker/) |