Accessibility Guide

FOR DIGITAL ASSETS

Last revision - April 30, 2019

Primerica Accessibility Guide for Digital Assets

SUPPORTING THE AMERICANS WITH DISABILITIES ACT (ADA)

Striving for digital accessibility.

In accordance with Title III of the Americans with Disabilities Act, Primerica continuously strives to provide equal access to goods and services for individuals with disabilities. Providing accessible electronic information technology to persons with disabilities is just as much a necessity as providing someone in a wheel-chair access to our facilities. Many people with disabilities use assistive technologies that enables them to use computers & other digital assets. Some assistive technologies involve separate computer programs or devices, such as screen readers, text enlargement software, and programs that enable people to control the computer with their voice. Properly designed websites and other digital assets help to remove unnecessary barriers for people with impairments or disabilities.

Primerica's goals are to ensure design and development work meets the accessibility guidelines. This document attempts to cover major aspects and components related to digital web assets.

Disabilities and impairments

Visual
Blindness, Low vision, Color-blindness.

Auditory
Mild, Moderate, Severe, Profound (absence of hearing).

Motor
Traumatic injuries, Cerebral palsy, Muscular dystrophy, Multiple sclerosis, Spina bifida, Lou Gehrig's disease, Arthritis, Parkinson's disease, Essential tremor.

Cognitive (Functional)
Memory, Problem-solving, Attention, Reading, Linguistic, Verbal comprehension, Math comprehension, Visual comprehension.

Seizure
Photo-epileptic seizures caused by strobing, flickering, or flashing effects.

Contents

Accessibility guide for development and design of digital assets.

Who is responsible for ADA compliance?

All teammates should be aware of and advocate for accessibility best practices across all digital assets.

What digital assets are affected?

Web Applications
Websites
Mobile Applications
Mobile Websites
Digital Communications
Digital Advertisements
Social Media Content

"One billion people, or 15% of the world's population, experience some form of disability, and disability prevalence is higher for developing countries. One-fifth of the estimated global total, or between 110 million and 190 million people, experience significant disabilities."

source: worldbank.org

WHAT'S IN THIS DOCUMENT?

Introduction

Understanding ADA, Section 508 Standards, and Section 504.

Digital Accessibility Initiatives
The Americans with Disabilities Act
Section 508
WCAG 2.1

Who Does this Apply to?
Anyone developing digital web assets for Primerica and its subsidiaries.

PRIMERICA DIGITAL ACCESSIBILITY INITIATIVES

Primerica has adopted the Web Content Accessibility Guidelines (WCAG) 2.1, level AA conformance which is an internationally recognized standard developed by the Worldwide Web Consortium to assist web developers in creating accessible web content for disabled users. These guidelines in combination with the Americans with Disabilities Act and Section 508 standards under the Rehabilitation Act, helps Primerica create accessible digital assets for a diverse range of user abilities.

THE AMERICANS WITH DISABILITIES ACT

The Americans with Disabilities Act, modeled after the Civil Rights Act of 1964, The Americans with Disabilities Act of 1990 (ADA) prohibits discrimination and ensures equal opportunity for persons with disabilities in employment, state and local government services, public accommodations, commercial facilities, and transportation.

For more information visit https://www.ada.gov/2010_regs.htm

ABOUT SECTION 508 STANDARDS

Section 508 is part of the Rehabilitation Act (a civil rights law), it requires Federal agencies to make their electronic and information technology (EIT) accessible to people with physical, sensory, or cognitive disabilities.

For more information visit https://www.section508.gov/content/learn/laws-and-policies

ABOUT WCAG 2.0 and 2.1 GUIDELINES

The W3C Web Accessibility Initiative (WAI) provides a set of web accessibility standards recognized around the world. The goal of WAI is to provide proactive guidance to make web content more accessible to people with disabilities and participate equally on the web. WAI's coverage of web accessibility includes websites, web applications, authoring tools, browsers and other 'user agents', and W3C technical specifications, including WAI-ARIA for accessible rich Internet applications.

On June 5,2018, the new WCAG 2-1 guidelines were formally published into law. The 2.0 success criteria are exactly the same (verbatim, word-for-word) in 2.1, but WCAG 2.1 provides 17 additional success criteria, 12 of which are applicable to level AA.

For more information visit https://www.w3.org/WAI/

Introduction

Understanding ADA, Section 508 Standards, and Section 504.

Digital Accessibility Initiatives
The Americans with Disabilities Act
Section 508
WCAG 2.1

Who Does this Apply to?
Anyone developing digital web assets for Primerica and its subsidiaries.

FOUR PRINCIPLES FOR WEB CONTENT ACCESSIBILITY | WCAG 2.1

WCAG 2.1 is organized under 4 principles, Perceivable, Operable, Understandable and Robust.

Principle 1 – Perceivable
Information and user interface components must be presentable to users in ways they can perceive.

- Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.
- Captions and other alternatives for multimedia
- Create content that can be presented in different ways (for example simpler layout) without losing information or structure.
- Make it easier for users to see and hear content including separating foreground from background.

Principle 2 – Operable user interface and navigation
Provide ways to help users navigate, find content, and determine where they are.

- Functionality is available from a keyboard
- Users have enough time to read and use the content
- Content does not cause seizures
- Users can easily navigate, find content, and determine where they are
- Make it easier for users to operate functionality through various inputs beyond keyboard.

Principle 3 – Understandable information and user interface
Make text content readable and understandable.

- Text is readable and understandable
- Content appears and operates in predictable ways
- Users are helped to avoid and correct mistakes

Principle 4 – Robust content and reliable interpretation
Maximize compatibility with current and future user agents, including assistive technologies.

- Content is compatible with current and future user tools

For more information visit https://www.w3.org/TR/WCAG20/#guidelines

DEVELOPMENT

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

NAVIGATION

All Navigation should be accessible by using the keyboard only. Assistive technology users should also be able to navigate and skip the main menus and start navigating or reading from the main body content of the web page or site. Below illustrates the skip the main navigational menu functionality. When implementing the Skip Navigation options, ensure the proper color contrast is used for font and background colors. Color changes should show keyboard focus.

Techniques & methods on how to create "Skip Navigation" links: http://webaim.org/techniques/skipnav

Navigation Focus Indicators

Focus and hover indicators should be used to indicate where the user is in the navigation. Active link indicators in the navigation should highlight the active link the user is on. In the example below, a blue line indicates the current page being visited.

EMBEDDED TEXT IN IMAGES

Text should be HTML and not embedded in images so that search engines can crawl and screen readers can read the page. Do not embed verbiage or content into images such as hero images and CTA buttons.

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

BUTTONS AND LINKS

Ensure buttons are HTML text links that are styled. The link itself should be text with proper description of what the link is about or routes to. Vague links such as "Click Here" or "Learn More" should be avoided, but if necessary they must provide ALT text describing the link's purpose.

ALTERNATIVE TEXT

Use Alternate text for all images (Primerica logo, hero, promotion tiles, social media, etc.) to provide a description of the image for screen readers. Alternate text should be a text equivalent of the image that a user would see by looking at the image. Example: *"Photo: Family playing. Text: We help families prepare for the future"* If the image has no value, use "null" as an alternative tag. Alternative tags should be 100 characters or less.

TAB ORDER

Ensure the proper tab order is in place for web pages, forms, and login fields. Users should be able to navigate through each website using only their keyboard from top to bottom. The tab order of all forms must be from left to right, top to bottom. As an example: A user logging into Primerica websites should not be able to tab through the fields to the "Sign On" button before tabbing through required fields before the button. Do not disable the default tab order for browsers using code.
NOTE: The tab order moves through interactive content, while arrow keys move through all clickable and non-clickable content. The arrow keys will work when using assistive technologies.

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

FORM ELEMENTS

Make form elements accessible by labeling each control (including buttons, check boxes, drop-down menus, radio buttons, and text fields) with a descriptive HTML tag, along with the appropriate ARIA attributes (ARIA details and examples are available later in this document).

Labels

```
<form action="http://www.Primerica.com/subscribe.php">
      <label>Age: <input type="text" name="age" /></label>
      <br/ >
      Gender:
      <input id="female" type="radio" name="gender" value="f">
      <label for="female">Female</label>
      <input id="male" type="radio" name="gender" value="m">
      <label for="male">Male</label>

</form>
```

Buttons

Parameterize accordingly based on how they are coded indicating the action that will take place.

```
<button type="submit">Submit</button>
<button type="button">Cancel</button>
<input type="submit" value="Submit">
<input type="button" value="Cancel">
```

Checkboxes

Parameterize checkboxes accordingly.

```
<input type="checkbox"
       name="service"
       value="agree"
       checked="checked" /> Agree
```

Radio Buttons

```
<input id="male" type="radio" name="gender" value="m">
```

Text Fields

```
<input type="text" name="username" size="15" maxlength="30" />
```

Age:

Gender: Female Male

Agree

User Name:

More on creating Accessible Forms: http://webaim.org/techniques/forms/controls
http://webaim.org/techniques/forms/advanced

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

TABLES

CAPTIONS
Use captions after the <table> tag to describe the data displayed.

IDENTIFYING COLUMNS AND ROWS
Complex data tables use the "id" attribute to denote names of columns and rows to make identification and navigation easier.

ROW AND COLUMN SCOPE:
Row and Column Headings with <TH scope="row"> and <TH scope="col"> should be present for simple data tables. Columns & rows should have unique headers. The scope attribute tells the browser and screen reader that everything within a column that is associated to the header with scope="col" in that column, and that a cell with scope="row" is a header for all cells in that row.

EXAMPLE A

```
<table>
  <caption>EXAMPLE A</caption>
  <tr>
    <th></th>
    <th scope="col"id="Month">Month</th>
    <th scope="col" id="Savings">Savings</th>
  </tr>
  <tr id="row1">
    <td scope="row">1</td>
    <td>January</td>
    <td>$100</td>
  </tr>
  <tr id="row2">
    <td scope="row">2</td>
    <td>February</td>
    <td>$80</td>
  </tr>
</table>
```

EXAMPLE A

	Month	Savings
1	January	$100
2	February	$80

For more information on tables visit http://webaim.org/techniques/tables/data
http://a11yproject.com/posts/accessible-data-tables/

FRAMES

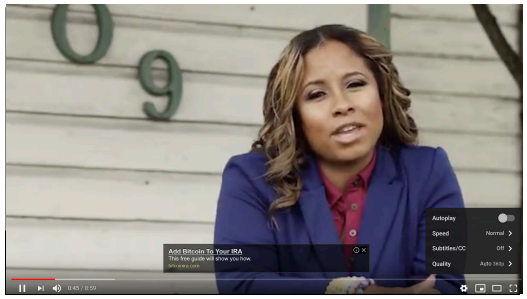
If using frames or iframes, ensure they are titled. Use of frames is not encouraged. However, if used, ensure the frames are readable by screen readers. The content should be readable by navigating to the section only utilizing the tab key as navigation through the page.

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

CAPTIONING VIDEOS

The Web Content Accessibility Guidelines (WCAG) requires captioning across all video media. Videos must have a PDF or text transcript link that can be downloaded and read by a screen reader. Ensure that the transcript is in a text format that can be read by screen readers.

An example of a Primerica video with closed captioning with an English language option.



Closed captioning settings for a video may allow the user to set specific fonts and sizes.

Things to know about captioning.

- Sign language can be added, but is not required for WC3 multimedia.
- Open captions play automatically, and closed captions appear only once you turn them on.
- Use 9 point type or larger on a background that contrasts enough for the text to be readable.
- Captioning are for users who are unable to hear the audio in a video or have cognitive impairments.

Tools for captioning.

- MAGpie; also creates audio descriptions
- Subtitle Workshop
- ccPlayer
- CapScribe
- CaptionTube
- World Caption
- CC for Flash
- Adobe captioning component
- DFXP

Learn about captioning videos: https://www.digitalgov.gov/2014/06/30/508-accessible-videos-how-to-caption-videos/https://www.w3.org/2008/06/video-notes#q5

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

INFORMATION and PAGE STRUCTURE

Common issues related to the information structure of a web page:

- All pages should contain headings (H1, H2, H3...). A H1 and H2 heading minimum is required.
- Do not use artificial bulleted lists using dashes or other symbols. Use a bullet list tag.
- PDF's should be tagged for accessibility.
- Use linked style sheets to control layout and presentation. Avoid using inline styling for font, backgrounds, and text treatments.
- Do not use drag and drop elements as these are not accessible if only the keyboard is used.
- Do not use white space characters to control spacing within a word, such as " " 3 or more times consecutively.
- Avoid using instructions that indicates "click on a shape or icon to continue" or to perform an action
- Avoid using sound only as an indication to continue or as an instructional audio element.
- When using tables, define the headers, header rows, column headers and table summaries (See Tables & Forms section).
- If browser text size is doubled (200%) ensure the page is functional and content is readable.
- Indicate when a link will open a new window through accurate labeling for screen readers.
- Do not change the navigation links when navigating through the site. The location of these elements should be consitent.

HTML lang ATTRIBUTE (LANGUAGE)

Identify the language of the page by defining it in the HTML lang attribute. The following example indicates the proper language.

IN ENGLISH:

```
<!DOCTYPE html>
<html lang="en">
```

IN SPANISH / ESPANOL:

```
<!DOCTYPE html>
<html lang="es">
```

For listing of language tags visit https://www.w3schools.com/tags/ref_language_codes.asp

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

ARIA ATTRIBUTES

What does ARIA (WAI-ARIA) stand for?
WAI stands for Web Accessibility Initiative. ARIA stands for Accessible Rich Internet Applications.

ARIA for Web Content and Applications:
ARIA helps to identify dynamic content, widgets, behaviors, structures and user controls developed with, but not limited to Ajax, HTML, JavaScript, and DHTML to users with disabilities and provides a way to add the missing semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets. ARIA attributes also help keyboard users navigate throughout a page without having to press the tab many times.

ARIA provides web authors with the following:

- Roles to describe the type of widget presented, such as "menu", "treeitem", "slider", and "progressmeter"
- Roles to describe the structure of the Web page, such as headings, regions, and tables (grids)
- Properties to describe the state widgets are in, such as "checked" for a check box, or "haspopup" for a menu.
- Properties to define live regions of a page that are likely to get updates (such as stock quotes), as well as an interruption policy for those updates—for example, critical updates may be presented in an alert dialog box, and incidental updates occur within the page
- Properties for drag-and-drop that describe drag sources and drop targets
- A way to provide keyboard navigation for web objects and events, such as those mentioned above

ARIA IS SUPPORTED BY

ARIA is supported in the latest versions of all major browsers, including Firefox, Safari, Opera, Chrome, and Internet Explorer. Many assistive technologies, such as the open source NVDA and Orca screen readers, also support ARIA.

For additional information about WAI-ARIA, visit https://www.w3.org/WAI/intro/aria.php
Check and validate ARIA markup in HTML5 DOCTYPE with W3C Nu Markup Checker: http://validator.w3.org/nu/

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

ARIA ATTRIBUTES

The ARIA specification is split up into three different types of attributes: roles, states, and properties.

- **Roles** describe widgets that aren't otherwise available in HTML, such as sliders, menu bars, tabs, and dialogs.
- **Properties** describe characteristics of these widgets, such as if they are draggable, have a required element, or have a pop-up associated with them.
- **States** describe the current interaction state of an element, informing the assistive technology if it is busy, disabled, selected, or hidden.

Quick Reference to ARIA Roles, States and Property Quick Reference to use in Markup:
https://www.w3.org/TR/wai-aria/appendices#quickref_header

ARIA ROLE ATTRIBUTE

Below is an example of a tabs widget with ARIA Role attributes added. For a list of Roles and definitions, visit https://www.w3.org/TR/wai-aria/roles#role_definitions


```
<!-- Role attributes added to describe the tab list and each tab. -->
 <ol role="tablist">
  <li id="ch1Tab" role="tab"><a href="#ch1Panel">Chapter 1</a></li>
  <li id="ch2Tab" role="tab"><a href="#ch2Panel">Chapter 2</a></li>
  <li id="quizTab" role="tab"><a href="#quizPanel">Quiz</a></li>
</ol>
```

ARIA PROPERTY ATTRIBUTE

Below is an example of a selectable menu with ARIA Property attributes added.

```
<!-- The role and aria-labelledby attributes were added to describe these panels. -->
<div>
  <div id="ch1Panel" role="tabpanel" aria-labelledby="ch1Tab">Chapter 1 content</div>
  <div id="ch2Panel" role="tabpanel" aria-labelledby="ch2Tab">Chapter 2 content</div>
  <div id="quizPanel" role="tabpanel" aria-labelledby="quizTab">Quiz content</div>
</div>
```

For more information on Property attributes visit https://www.w3.org/TR/wai-aria/states_and_properties

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

ARIA STATE ATTRIBUTE

Dynamic presentational changes include using CSS to change the appearance of content (such as a red border around invalid data, or changing the background color of a checked checkbox), as well as showing or hiding content. ARIA provides attributes for declaring the current state of a UI widget.

Examples include (but are not limited to):

- **aria-checked:** indicates the state of a checkbox or radio button
- **aria-disabled:** indicates that an element is visible, but not editable or otherwise operable
- **aria-grabbed:** indicates the 'grabbed' state of an object in a drag-and-drop operation

Below is an example of a selectable menu with ARIA State attributes added.


```
<ul id="fontMenu"
  class="menu"
  role="menu"
  aria-hidden="true">
<li id="serif"
  class="menu-item"
    role="menuitemradio"
    tabindex="-1"
    aria-controls="st1"
    aria-checked="false">Serif</li>
<li id="sans-serif"
  class="menu-item"
    role="menuitemradio"
    tabindex="-1"
    aria-controls="st1"
    aria-checked="true">Sans-serif</li>
```

NOTE: Use ARIA states to indicate the state of UI widget elements and use CSS attribute selectors to alter the visual appearance based on the state changes (rather than using script to change a class name on the element).

The CSS that is used to alter the visual appearance of the selected item is shown in the example above. Note that there is no custom classname used, only the status of the aria-checked attribute on line 1 below.

```
li[aria-checked="true"] {
    font-weight: bold;
    background-color: #0075c9;
    }
```

For a complete list and clarification of 'state' versus 'property' can be found at https:/www.w3.org/TR/wai-aria/states_and_properties

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

WAI-ARIA ATTRIBUTE DON'TS

What NOT to do with ARIA attributes:
Web developers *must not* use the ARIA role and aria-* attributes in a manner that conflicts with the semantics described in the Document conformance requirements for use of ARIA attributes in HTML table. Web developers *should not* set the ARIA role and aria-* attributes to values that match the implicit ARIA semantics defined.

The following example uses a *role=heading* on a button element. This is not allowed, because the button element has default characteristics that conflict with the heading role.

```
<button role="heading">search</button>
```

For Additional examples in JavaScript visit https://developer.mozilla.org/en-US/docs/Web/Accessibility/

WAI-ARIA REFERENCE VIDEOS

Dynamic Form Lables with ARIA: https://youtu.be/Xr32OASZO_Q
Creating Accessible Components & Screen Readers: https://youtu.be/Lktz1KXbTOU ARIA Labeling for Checkboxes and Radio buttons: https://youtu.be/1D25YXLBBX8

WAI-ARIA BEST PRACTICES REFERENCE GUIDE

WAI-ARIA Authoring Practices: https://www.w3.org/TR/wai-aria-practices/

- Design Patterns and Widgets
- Landmark Regions - identify the structure of a web page by classifying and labeling page section
- Developing a Keyboard Interface - Navigation of components using keyboard
- Hiding elements that are only for presentation such as decorative images

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

MODAL WINDOWS

Modal windows are windows that shift your focus from an existing part of the website to another area or focus. In order to ensure assistive technology users are able to utilize the keyboard when modal windows appear, proper semantics must be used for interaction and to read the content within the window.

Modal window close "X" example:
The below example ensures assistive technology users that use keyboards will know what the "X" means in order to close the window. Adding the correct ARIA label allows the screens readers to know what the "X" means.

```
<div id="modal_overlay">
<button type="button" class="btn-close" id="modal_close" aria-
label="close"> X </button> </div>
```

Modal window checklist:

- **Savings Last Active Element** - when a modal window opens the element the user last interacted with should be saved. Once the modal window is closed the user should return to where they were.
- **Shifting Focus** - when shifting focus from the last element to a new element, the first interactive element should be the automatic focus. This helps assistive technologies users who use keyboards.
- **Full Screen Modal - For** modal windows that take over the whole screen confine the contents of the main page so that screen readers and sighted users cannot easily tab out of the modal without realizing it and thus not complete what was asked of them in the modal window.
- **Dismissing Modal Window** - Modal windows should be easy to dismiss. For "Alert" windows, allowing the user to hit either ESC (escape) or the "close" X should allow users to dismiss the window. However, in some cases you may not want the user to close the modal window until the user performs an action.
- **ARIA Roles, States, and Properties:**
  Use *role="dialog"* to the element that contains the modal's content.
  Use *role="alertdialog"* if the element is an Alert or error message.
  Use *aria-label* or *aria-labelledby* if the modal window has a heading. If no heading, use aria-label.

Example of an accessible modal: https://www.smashingmagazine.com/wp-content/uploads/2014/accessible.html

Who Does this Apply to?
Anyone developing digital web assets for Primerica or its subsidiaries.

ALERTS AND TIME OUTS

Alert dialogs interrupts a user's workflow to communicate an important message that requires a response or attention. Alert dialogs are distinguishable from other dialogs when using assistive technologies and browsers. The use of the role="alertdialog" allows assistive technologies the option of giving alert dialogs special treatment, such as playing a system alert sound.

Alert Dialog Best Practices:

- Warn users when they are being redirected to a new site or page.
- Avoid designing alerts that automatically disappear.
- Avoid frequent alert or notification dialog interruptions inhibits usability for people with visual and cognitive disabilities.
- The element that contains all elements of the dialog, including the alert message and any dialog buttons, has role *role="alertdialog"* .
- The element with role alertdialog has either:
  - A value for *aria-labelledby* that refers to the element containing the title of the dialog if the dialog has a visible title
  - A value for aria-label if the dialog does not have a visible title.
- The element with role alertdialog has a value set for aria-describedby that refers to the element containing the alert message.

Sound and Color

Do not rely on sound and or the use of color to provide instructions to a user. Users with learning, hearing and/or vision impairments or any user will not be able to interpret the instructions properly.

ARIA alert Role Example: https://www.w3.org/TR/wai-aria-practices/examples/alert/index.html

DESIGN

Who Does this Apply to?
Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

ACCESSIBLE IMAGES

Alternative text for images are used for screen readers. However, there are additional techniques and principles regarding creating proper images to not only enhance accessibility, but also increase comprehension. Those with other disabilities, such as learning disabilities, reading disorders or no disabilities can also benefit from illustrations, maps, charts, and animations.

- **Illustrations** with preferably HTML text descriptions can enhance a users ability to absorb and understand the content.
- **Maps & Charts** in color should use not lose its meaning if the maps are converted to black & white
- **Animations** that are well designed can enhance and provide visual focus on content. Animations should almost always be user controlled or very short in duration.
- Icons should be consistently used and often times should be used in addition to text. Be mindful that icons can be interpreted differently across cultures and languages.
- Flashing, Strobing, Bright and Dynamic images or media can cause photo-epileptic seizures. Seizures can be dangerous and even life-threatening.
- Images with embedded text are not preferred. However, if used, it will be increasingly difficult for vision impaired users to read. When images with embedded text are enlarged the text in the image becomes pixelated. True text should be used when practical.

To avoid causing seizures for users with photosensitive epilepsy for flashing image or multimedia:

- Do not flash more than 2 times per second.
- Do not use large flashing images. Small flashing images such as a cursor will not cause a seizure.
- Do not use high contrast between flashes.
- Additionally, the color red is more likely to cause a seizure.

More information on accessible graphics can be found at http://webaim.org/techniques/images/#seizures

Who Does this Apply to?
Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

COLOR CONTRAST

What is color contrast?
Color contrast involves the color combinations for sufficient contrast according to WCAG 2.1 (Luminance Ratio). Color contrast is important when there is a combined foreground and background. Primerica strives to have the proper color contrast ratio to comply with WCAG 2.1, Level AA.

- Menus and Navigational elements
- Titles and Headers
- Font colors in combination with any background colors
- Font colors in combination with font sizes and weights
- Background colors in combination with images or text

Color contrast examples for Level AA rating:
The table illustrated on the next page is an example of color contrast to ensure conformance with WCAG 2.1 guidelines. It displays colors used by Primerica as our branded standard. Normal text must always have a contrast ratio of at least 4.5:1 or higher. Large or bold text must have a color contrast ratio of 3:1 or higher.

- All color ratios which tested lower than 3:1 are marked **FAIL**, and should *not* be used under any circumstances for text content.
- Color ratios between 3:1 and 4.5:1 are marked as LARGE/ **BOLD** and should *only* be used with large text and bold text, **19 pixels and bold or larger**, **OR 24 pixels (24px/1.5em) or larger with a normal font weight.**
- Color ratios above 4.5:1 are marked as PASS and can be used both with normal text, **below 19 pixels (19px/1.1875em) or Larger**. PASS means the foreground and background pairing is suitable for
  all text purposes, normal and large text.
- Purely decorative images or text in images have no color contrast requirement.
- Logos: Text that is part of a logo or brand name has no minimum contrast requirement.

DEVELOPER NOTE:
Relative units such as **Percents** or **EMs** for the font size is preferred. Points are only used when coding for print.

For pixels to EM conversions, visit https://www.w3schools.com/cssref/css_pxtoemconversion.asp

Design

Choosing the Right Design Elements

Who Does this Apply to?

Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

DIGITAL BRAND COLORS

Please see our latest brand colors on our Design System.
Color contrast ratios can be checked at http://webaim.org/resources/contrastchecker/

DO'S AND DON'TS

Below are some Do's and Don'ts examples regarding color contrast and designing for digital assets.

VISUAL TREATMENTS:

For digital components and patterns with white text and a colored background, use the color contrast ratio checker to ensure the color and font size combination conforms to WCAG 2.1 guidelines.

Below are examples of graphics treatments for text and color combinations. F ollow the brand guidelines for each line of business.

CORRECT

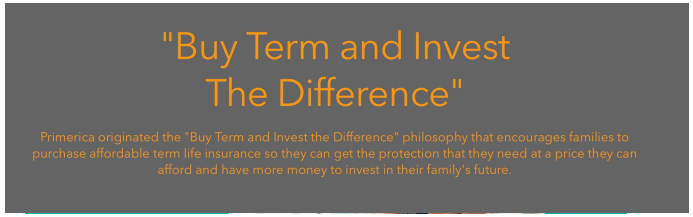
Correct use of a solid (non-transparent) background color with the proper color contrast ratio. *(true size not represented below)*

Correct use of background color block and button.

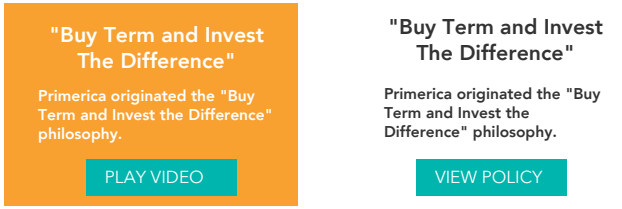
Correct color contrast ratio for buttons.

INCORRECT

Incorrect use of a solid (non-transparent) background color. *(true size not represented below)*



Incorrect use of background color block and buttons.



Incorrect color contrast ratio for buttons.

Who Does this Apply to?

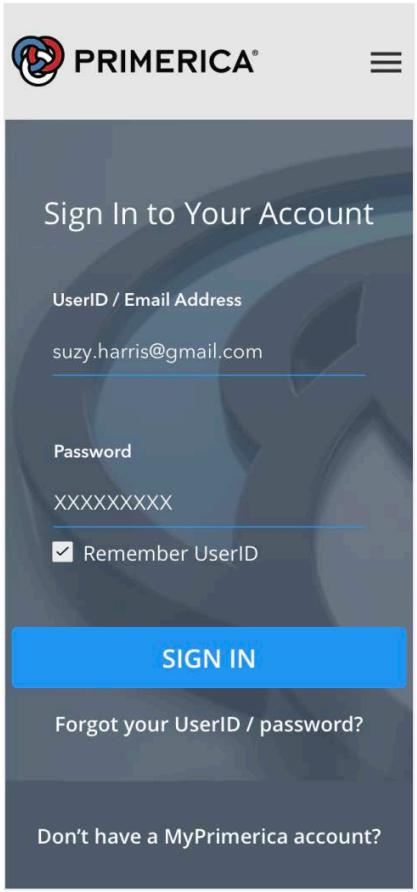
Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

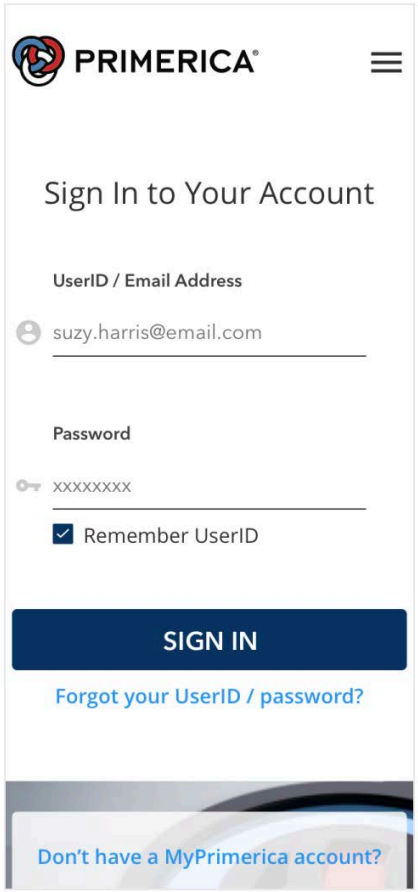
DIGITAL ASSETS EXAMPLES

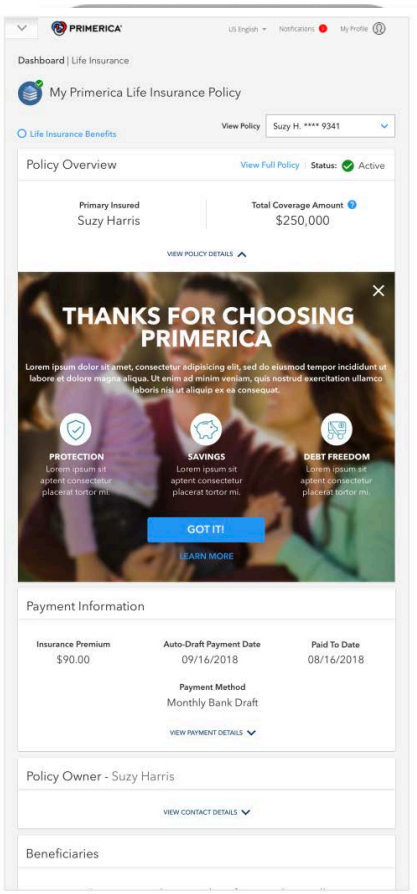
Below are some examples of digital assets that have been designed to align with WCAG 2.1 guidelines, level AA.

Mobile App

Digital components and patterns with white text and a colored background uses proper color contrast ratios. Colors that may fall outside of the proper color contrast ratio may be used as graphic accents such as icons. However, using graphic elements that have the pr oper color contrast ratio are recommended.







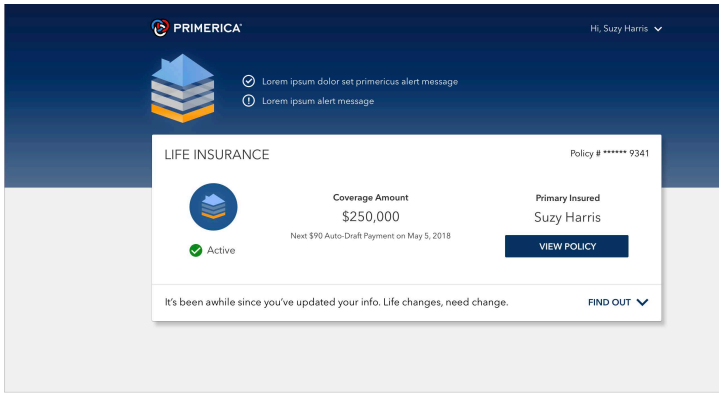
Who Does this Apply to?

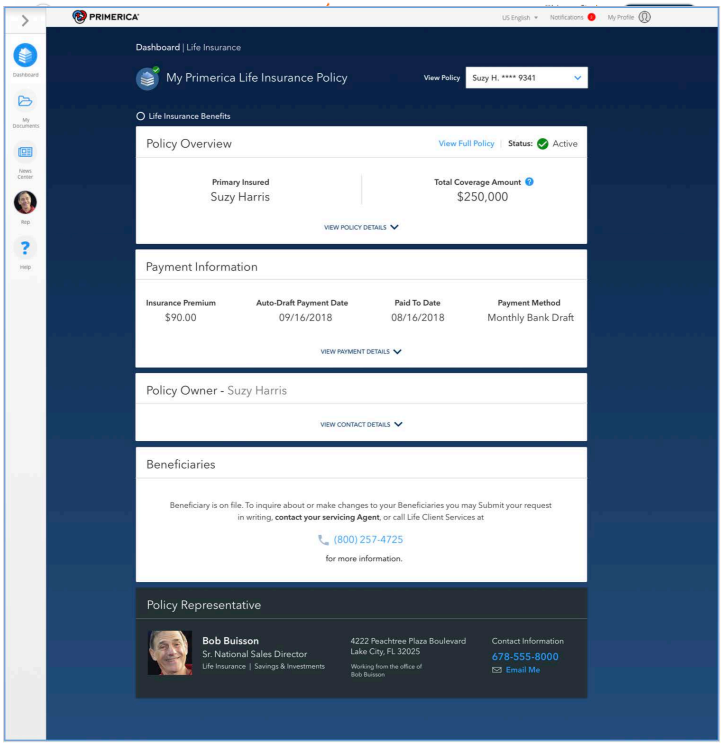
Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

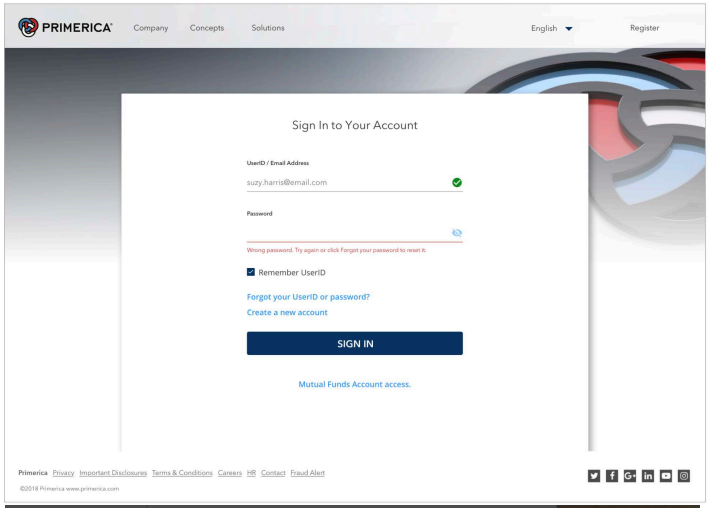
DIGITAL ASSETS EXAMPLES

Life Policy

Navigational elements, modules and components reflects proper color contrast ratios. Colors that fall outside of the proper color contrast ratio are used as graphic accents such as icons.







Modal windows:
The proper ARIA attributes should be used to help assistive technologies users focus on the first section of content within the window.

Who Does this Apply to?

Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

DOCUMENT CREATION

Where necessary, provide documents in an alternative text-based format, such as HTML or RTF (Rich Text Format), in addition to PDF. Text-based formats are the most compatible with assistive technologies.

USE OF ACRONYMS

Refrain from using acronyms. Instead, spell out the entire acronym.

SCREEN FLICKERING OR FLASHES

Design all digital assets to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz. The frequency of flashing imagery or pages may possible cause physical injury due to visual over stimulation.

USE OF FONT

Fonts should be controlled using cascading style sheets (CSS). The recommended fallback or secondary font is Arial, which is a commonly used standard font available on the end users device.

- Do not use font icons. Use real font and not images.
- Use readable fonts and font weights. Do not use cursive fonts.
- Use a limited number of fonts. Using too many font faces may make it difficult for users with reading disorders, learning disabilities, or attention deficit disorders.
- Ensure font weights provides proper color contrast based on background colors.
- Avoid small font sizes. The minimum FS Albert font size for body text is 15px; For legalese 14px.
- Do not rely *only* on font colors or variations to convey meaning to a user.
- Do not use blinking or moving text. Users with slower reaction times, tremors, or other motor difficulties may not be able to click on the links accurately.
- Using all capitals is discouraged. However, there are exceptions. Screen readers will generally read capitalized text and normal case the same. However, it may change the voice inflection with exclamation points. Screen readers also may read the uppercase as an acronym. For example, “CONTACT US” as “Contact U.S.”, and interpret “US” as “United States”.
- Use of relative units, percents or ems, is preferred for font sizes. Pixels are preferred over points, unless used for print.

RESOURCES

Who Does this Apply to?

Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

SCREEN READERS

The following desktop and mobile device tools are recommended for use at Primerica.

Desktop

- NVDA Screen Reader: http://webaim.org/blog/testing-with-the-nvda-screen-reader/
- Apple VoiceOver: http://www.apple.com/accessibility/mac/vision/
- JAWS (Windows): http://www.freedomscientific.com/Products/Blindness/JAWS

Mobile Devices

- Apple VoiceOver: http://www.apple.com/accessibility/mac/vision/
- Talkback (Andriod Accessibility): https://support.google.com/accessibility/android/answer/6283677?hl=en

NOTE: Apple VoiceOver also works as a face reader.

DEVELOPER TOOLS

Mozilla Firefox Accessibility Evaluation Toolbar (Must have FireFox)
Access Safari Developer Tools (Inspector: Command-Option-I
WAVE Tool: Strips out visuals on page to content only to determine the order of the content on a web page
WAI-ARIA: For Dynamic components such as progress Bars, Use in HTML, for Form Accessibility, ARIA tag for ALT on images that do not require a description: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA
Progress Bar for ARIA: http://www.w3schools.com/bootstrap/bootstrap_progressbars.asp
Closed Captioning on YouTube: https://www.google.com/accessibility/for-developers.html#youtube
Accessibility for Android for Developers: https://developer.android.com/guide/topics/ui/accessibility/index.html
Google Chrome Vox for Accessibility: http://www.chromevox.com/
Broken Links Checker: https://validator.w3.org/checklink
Check and validate WAI-ARIA markup in HTML5 DOCTYPE: http://validator.w3.org/nu/

Who Does this Apply to?

Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

WCAG 2.1 CHECKLIST

The following checklist can be used as a method of veifying common accessibility issues for digital assets. http://webaim.org/standards/wcag/checklist

CSS Validation: Check your CSS (Cascading Style Sheets to find errors, typos, and incorrect use of CSS as well as usability risks. https://jigsaw.w3.org/css-validator/

BASIC TERMS

Website – a collection of web pages that is hierarchically organized around a homepage.

Web Browser – a computer program that downloads web pages. It is the program installed on your computer that you use to access web pages on the Internet.

HTML – short for "hypertext mark-up language," a common mark-up language used to present web pages. It tells the web browser how information should be structured and accessed.

Screen Reader – a computer program that speaks written text. It allows a person to listen to the written text on a web page or in a computer program. Screen readers read only text; they cannot describe pictures or other images, even if the images are pictures of text.

HTML Tags – specific instructions understood by a web browser or screen reader. One type of HTML tag, called an "alt" tag (short for "alternative text", is used to provide brief text descriptions of images that screen readers can understand and speak. Another type of HTML tag, called a "longdesc" tag (short for "long description", is used to provide long text descriptions that can be spoken by screen readers.

Refreshable Braille display – an electronic device that translates standard text into Braille characters and uses devices such as rounded pins on a refreshable display to create Braille text that can be read by touch.

Color Contrast – the color combinations used for sufficient contrast according to WCAG 2.1 (Luminance Ratio.

WCAG – stands for Web Content Accessibility Guidelines, which was developed by the Web Accessibility Initiative (WAI as a resource for designers and developers to make web pages more accessible to a diverse range of users which include disabilities or impairments.

Who Does this Apply to?

Anyone designing or writing requirements for digital web assets for Primerica or its subsidiaries.

EDUCATIONAL VIDEOS

Technology and cerebral palsy: https://www.youtube.com/watch?v=glR-Q6xFe-w

How Tech transforms and enables persons with disabilities: https://www.youtube.com/watch?v=vSDvu_RvKJs

Life with Essential Tremor: https://www.youtube.com/watch?t=1&v=pQNKCa-2oSY

REFERENCES AND SOURCES

GUIDELINES

ADA.gov: https://www.ada.gov/pcatoolkit/ch5_toolkit.pdf

Section 508 Standards Web Content Accessibility Guideline: https://tinyurl.com/ycy7t7x3

Angular JS: https://docs.angularjs.org/api/ngAria

BROWSER TOOLS

aXe (Accessibility testing in Chrome Developer Tools): https://tinyurl.com/n6g5ozj

aXe-CLI: - https://github.com/dequelabs/axe-cli

Chrome Accessibility Toolbar: https://tinyurl.com/mqq2yk9

Mozilla Firefox Accessibility Evaluation Toolbar: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA

Safari Developer Resource: https://developer.apple.com/safari/resources/

EVALUATION TOOLS

WAVE (Web Accessibility Evaluation) Tool: http://wave.webaim.org/

W3schools: http://www.w3schools.com/bootstrap/bootstrap_progressbars.asp

Webaim.org - Web Accessibility in Mind: https://jigsaw.w3.org/css-validator/

AUDIO

Apple VoiceOver: http://www.apple.com/accessibility/mac/vision/

Talkback (Andriod Accessibilty): https://support.google.com/accessibility/android/answer/6283677?hl=en

VIDEO

Closed Caption Maker: http://www.ccmaker.com/

Transcript Best Practices: http://www.uiaccess.com/transcripts/transcripts_on_the_web.html#bests

Digitalgov.gov: https://www.digitalgov.gov/2014/06/30/508-accessible-videos-how-to-caption-videos/

YouTube - Google for ADA: https://youtu.be/glR-Q6xFe-w,
https://youtu.be/vSDvu_RvKJs,
https://youtu.be/pQNKCa-2oSY

CONTACT US

Contact Us

Primerica Internal Contacts

CXi Team
Primerica Design System
Design System Forum

Who should I contact?

The Accessibilities Guideline is an enterprise-wide initiative for all digital assets. Please contact the team most appropriate for your question or concern.

CXi Team, PRIMERICA DESIGN SYSTEM, DESIGN SYSTEM FORUM

The contact list below is for internal use only.

CXi Team

_DL.UXTeam@Primerica.com

Primerica Design System

http://primerica-design-system.webflow.io

Design System Forum

The Design System Forum is made up of SMEs across Primerica. This forum has been created and recognized as part of the ever-evolving culture of Primerica. This forum is to ensure that our Design System continues to evolve and preserve excellence in every digital experience Primerica provides to its valued Representatives and Clients. Below is a list of representatives of the Design System Forum representing each discipline within Primerica.

If you should have a question or need assistance in using this Design System please reach out to: Debbie Pelton or Josue Lux

Forum SMEs

CXi
Josue Lux, Debra Pelton, Thuycam Nguyen, Andres Prieto, Judy Thomas, Maulik Soni, Christopher Manning
Marketing
Blake Stewart
Communications
Natalie Daly
Technology
Hetal Shah
Digital
Product: Nick Stowe
Legal
Bunny Sandefur