|  |
| --- |
|  |
| **Web Accessibility Testing Checklist** |
| May 2017 |

Table of Contents

Testing Web Application Accessibility 2

Introduction 2

Testing Tools 2

1. Interactive Interface Elements 3

1.1. Keyboard Access 3

1.2. Labels for Web Forms 4

1.3. Dynamic Content from Client-Side Scripts 6

2. Non-Text Interface Elements 7

2.1. Non-Text Content 7

2.2. Audio and Video 8

3. Color and other Sensory Dependencies. 9

4. Flickering 10

5. Data Tables 10

6. Style Sheet Dependence and Semantic Markup 11

7. Repetitive Navigation Links 12

8. Required Plug-ins 12

9. Timed Response 13

10. Text 14

References 15

# Testing Web Application Accessibility

## Introduction

This is a guide to assessing accessibility for Web applications. This guide is based on Web Accessibility in Mind's (WebAIM) interpretation of Section 508 and the Web Content Accessibility Guidelines (WCAG) 2.0 standards. It does not guarantee compliance, but will help you identify many common accessibility-related issues. For the full text of section 508, please see the official 508 website (http://www.section508.gov/). For the full text of the WCAG 2.0 standards see the WCAG website (http://www.w3.org/TR/WCAG20/).

Use the following guide to test your Web applications. Anything marked as "Fail" should be fixed unless it is impossible to do so.

## Testing Tools

The following tools are helpful for testing accessibility.

|  |  |
| --- | --- |
| **Firebug** | https://addons.mozilla.org/en-US/firefox/addon/firebug/ |
| **WebAim's WAVE Accessibility Toolbar** | http://wave.webaim.org/toolbar/ |
| **WebAim's Color Contrast Checker** | http://webaim.org/resources/contrastchecker/ |
| **NVDA Screen Reader (Windows)** | http://webaim.org/articles/nvda/ |
| **VoiceOver Screen Reader (Mac)** | http://webaim.org/articles/voiceover/ |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Interactive Interface Elements | | | |
| Interactive elements include menus, forms, navigation, buttons, maps, etc.. | | | |
| * 1. Keyboard Access | | | |
| All interface elements and functions that can be accessed by mouse must also be accessible to users who only use the keyboard to navigate. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Check for the visual focus that moves with keyboard navigation. This is typically a dotted rectangle that outlines a button or link or a vertical bar in a text field.  **WCAG 2.4.7/508 Scripts (c)**  If there is no visual indication of current focus, open Firebug (or other tool) and add a style to add it e.g. a:focus {background-color:yellow;}. This will help you determine keyboard accessibility. | Visual indication of the current focus is always available. | Visual indication of the current focus is lost. | |
| 2 | Use keyboard shortcuts (Tab, [Shift+Tab], Space bar, arrow keys, Enter, etc.) and the application’s special keystrokes to move the focus to and activate all menus and functions. **WCAG 2.1.1/508 Scripts (a)** | All interactive elements and functions can be accessed by keyboard. | Any interactive element or function cannot be accessed by keyboard. | |
| 3 | Check to ensure that the application's shortcut keys do not conflict with existing browser and screen reader shortcuts. **WCAG 2.1, 508 Scripts (b)** | All browser and screen reader shortcut keys work. | Any browser or screen reader shortcut key is used by application or is disabled. | |
| 4 | Check to ensure that the keyboard focus is never trapped. **WCAG 2.1** | Keyboard focus moves to and from all navigable page elements. | Keyboard focus is trapped. | |
| 5 | Navigate through all form fields with the keyboard. Enter text, arrow to different options from drop down lists, select and unselect checkboxes and radio buttons.  **WCAG 2.1.1/508 Scripts (a)** | All form fields can be completed using the keyboard. | Any form field cannot be accessed by keyboard. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * 1. Web Forms | | | | |
| Form fields must be explicitly associated with text labels so that users can complete the forms using a screen reader. | | | | |
| No. | | How to Test: | Pass | Fail | | |
| 1 | Tab through the form to see that the navigation order of links, form elements, etc. is logical and intuitive. **WCAG** [**2.4.3**](http://www.w3.org/TR/WCAG20/#navigation-mechanisms-focus-order)**/508 Scripts (l)** | The navigation order of links, form elements, etc. is logical and intuitive. | The navigation order of links, form elements, etc. is not logical and intuitive. | |
| 2 | Click on the text label for each form field to see if the appropriate form input is put in focus. Alternately, use WAVE to check the labels.  **WCAG** [**1.3.1**](http://www.w3.org/TR/WCAG20/#content-structure-separation-programmatic)**/508 (n)**  If there is no text label, view the source to see if a screen readable label is available. Ensure styles used to hide labels are still accessible to a screen reader. | All form fields have associated text labels. | Any form fields are not labeled properly. | |
| 3 | If there are instructions for form elements that require a specific format, value, or length, use Firebug to confirm they are provided within the element's label.  **WCAG 3.3.1/508 Scripts (l)** | Instructions are inside the label. | Instructions to fill in a form field are not inside of the label. | |
| 4 | If there are required fields, use Firebug to check to see if sufficient cues are provided via instructions, examples, properly positioned form labels, and/or fieldsets or legends. **WCAG 3.3.2/508 Scripts (l)**  **If required text is used for a radio or checkbox group, the instructions should be inside the legend as part of a fieldset.** | **There are sufficient labels, cues, and instructions. They are properly positioned.** | **There are not sufficient labels, cues and instructions for interactive elements.** | |
| 5 | Using the keyboard, complete some form fields incorrectly and review the error messages. (For example, use the wrong format or leave some required fields incomplete). If an input error is displayed, ensure suggestions for fixing the issue in a timely and accessible manner are available. **WCAG 3.3.1/508 Scripts (l)** | **Form validation errors are presented in an intuitive, efficient, and accessible manner.** | Form validation errors are not presented in an efficient, intuitive, and accessible manner | |
| 6 | Fill out form fields with incorrect information to see if there are helpful error messages. **WCAG 3.3.3** | Input errors should be detected and provide suggestions for correct input where possible. | No suggestions are provided for incorrect inputs. | |
| 7 | Use Firebug to check that form buttons have a descriptive value. If form buttons are images, make sure they have descriptive alt text. **WCAG 1.1.1/508 (a) and Scripts (l)** | **Form button has a descriptive value or descriptive alt text.** | **Form button does not have a descriptive value or descriptive alt text.** | |
| 8 | Use Firebug or WAVE to check to see that related form elements are grouped with fieldsets and legends. **WCAG 1.3.1/508 Scripts (l)** | Related form elements are grouped with fieldset/legend. | Related form elements are not grouped with fieldset/legend. | |
| 9 | Fill out form fields that cause legal or financial commitments or transactions and try to reverse them. **WCAG 3.3.4** | Web pages that cause legal commitments or financial transactions for the user to occur must either be reversible and confirmed. | Transactions cannot be reversed. | |
| 10 | "ARIA is supported by most up-to-date browsers and screen readers. It is also supported by many scripting libraries. Although ARIA is not yet universally supported, when used with existing HTML and scripting accessibility techniques, it can provide additional accessibility support where it is supported while not causing compatibility issues where it is not yet supported." WebAIM (http://webaim.org/techniques/aria/) | ARIA is not required by 508 or by WCAG but helps accessibility. |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| * 1. Dynamic Content from Client-Side Scripts | | | |
| Content and functionality provided by scripting must be directly accessible to assistive technologies and keyboards. | | | |
| No. | How to Test: | Pass | Fail | | |
| 1 | Use the mouse to identify any mouse events that display new information. Then check to see if those events are keyboard accessible. **508 (l)** | Content and functionality provided by scripting is directly accessible to assistive technologies and the keyboard. | Content and functionality provided by scripts only work with a mouse or cannot be accessed by assistive technologies. | |
| 2 | Use screen reader software to listen to Web pages where the content is generated dynamically (i.e., without a page refresh). Ensure that the focus is placed where the new content is.  **508 (l)** | Dynamically generated content is read by the screen reader where and when it is contextually relevant. | Dynamically generated content is not read by the screen reader or is not read where or when it is contextually relevant. | |
| 3 | Use WAVE to detect popup windows. If used, ensure that the link that triggers the pop-up window is labeled with either text or an icon with appropriate alt text to indicate that a new window will open.  **WCAG 3.2.1, 508 Scripts (d)** | Links to popup windows are labeled appropriately. | Links to popup windows are not labelled appropriately. | |
| 4 | If any other substantial change is made to the page due to a user action, ensure the user is informed before the action happens. | Users are informed that an action will cause substantial changes made to the page. | Users are not informed that an action will cause substantial changes to the page. | |
| 5 | If the user must interact with the page to see content, ensure there are instructions.  **WCAG 1.3.3** | Instructions are provided for understanding and operating content. Do not rely solely on sensor characteristics of components such as shape, size, visual location, orientation or sound. | User does not have instructions. | |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Non-Text Interface Elements | | | |
| Images, animations, graphs, and audio files must have equivalent descriptions for screen reader users. | | | |
| * 1. Non-Text Content | | | |
| Provide text alternatives to any non-text content. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Use WAVE to check every image, applet, embedded media, plug-in, etc. on the page for equivalent text.  **WCAG 1.1.1, 508 (a).** | Every image, applet, embedded media, plug-in, etc. that conveys content has equivalent alternative text (alt, longdesc, or in the element context). | A non-text element has no alt or text description or the description is not equivalent, or is not described in the adjacent text. |
| The alternative text succinctly describes the content conveyed by the element, without being too verbose (for simple objects) or too vague (for complex objects). Phrases such as "Image of" or "Photo of" are not used. | Alternative texts are verbose, vague, misleading, inaccurate, or redundant to the context (e.g. the alt text is the same as adjacent text). Phrases such as "Image of" or "Photo of" are used. |
| Complex graphics (graphs, charts, etc.) are accompanied by equivalent text, either through a description in the body of the page, a link to a description on a separate page, and/or the longdesc attribute. | Complex graphics have no alternative text or the alternative does not fully convey the content of the graphic. |
| Images that have a function (images within links, image buttons, and image map areas) have alternative text, which describes the associated function. | Alternative texts for linked images, image buttons, or hot spots are not descriptive of the function. |
| Decorative graphics are CSS background images or have null/empty alt values (alt=""). Images with text alternatives in element content are given empty alt text to avoid redundancy. | Decorative graphics have alternatives of "spacer", "decorative graphic," or other extraneous text. Graphics have alt text that is redundant with adjacent text. |
| 2 | Check if images with text could be created without the image. **WCAG 1.4.5** | If you can achieve the visual presentation of an image with text using only text then do so. | Image with text could have been created without text. |
| 3 | Pause, stop or hide moving, blinking, scrolling or autoupdating text. **WCAG 1.4.2** | Moving, blinking, scrolling, and auto updating information can be paused, stopped or hidden. | Moving, blinking, scrolling, and auto updating information cannot be paused, stopped or hidden. |

|  |  |  |  |
| --- | --- | --- | --- |
| * 1. Audio and Video | | | |
| Audio and video must be made accessibile. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Manually check any audio or video on the page to ensure there is an accessible alternative. **WCAG 1.2, 508 (b)**  WebAIM recommends providing both captions and a transcript. Transcripts are searchable. They are also accessible to both the deaf and blind because it outputs as refreshable Braille. | Recorded or live video files and live audio broadcasts have synchronizedcaptions. | Recorded or live video files or live audio broadcasts do not have captions or captions are not synchronized.  \*Note: DOE only requires a text transcript. |
| Content presented through video, but not through audio, is provided through a text or audio description track. | Audio or text descriptions are not provided for visual-only content in multimedia. |
| Transcripts are provided for audio content. | Audio does not have transcripts. |
| 2 | When you load a page, ensure that sound on an audio or video files does not automatically play. | Sound on audio or video files does not autoplay. | Sound on audio or video files autoplays. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Color and other Sensory Dependencies | | | |
| Information provided by color, sound, or shape must also be available without color, sound, or shape. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Manually look at page to see if color, sound, or shape is used to convey important information.  **WCAG 1.3 and 1.4, 508 (c)** | Color, sound, or shape is not exclusively used to convey important information. | Color, sound, or shape is the sole means of conveying information. |
| 2 | Check links in content to ensure they stand out from surrounding text using a visual indicator such as an underline. If only color is used, check to see if there is sufficient contrast between the link text and body text with the WebAIM color contrast checking tool. | A visual indicator such as an underline is used to indicate links in content. Color alone is not used to distinguish links from surrounding text unless the contrast is sufficient. | Links in content are only indicated by a text color with insufficient color contrast. |
| 3 | Manually scan the page to see if there are any potential issues with the color contrast between text and background color. Use the WebAIM color contrast checking tool to test. | Text and images of text have a contrast ratio of at least 4.5:1.  Large text (over 18 point or 14 point bold) has a contrast ratio of at least 3:1 | The text and images of text do not have sufficient contrast. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Flickering | | | |
| Flickering interface elements may cause seizures for users with photosensitive epilepsy. Look for any repetitive blinking, flickering, or scrolling interface elements. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Visually check for any flashing, scrolling or blinking interface elements. **WCAG 2.3, 508 (j)** | No page content flashes more than 3 times per second. | One or more elements on the page flashes more than 3 times per second. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Data Tables | | | |
| Data table column and row headers must be identified. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Use Firebug to look at the source code for the data tables. Ensure column headers and row headers use THs. If the table is complex and uses subheaders, check to ensure headers and ids are used appropriately.  **WCAG 1.3, 508 (h)** | THs are used for row and/or column headers on simple tables. Use Scope = "col" or scope="row" for table headers.  Data table cells are associated with the appropriate headers using the [scope](http://webaim.org/techniques/tables/data#scope) or [id/headers](http://webaim.org/techniques/tables/data#id) attributes.  [Tables](http://webaim.org/techniques/tables/data) are used for tabular data. Data table captions and summaries are used where appropriate. | THs are not used for row and/or column headers on simple tables.  Data table cells are not associated with column and/or row headers or they are associated incorrectly. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Style Sheet Dependence and Semantic Markup | | | |
| Create content that can be presented without markup and still be understandable. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Use WAVE to view the page. Choose Structure/Order. Check markup to ensure that proper headings and other semantic markup is used.  **WCAG 1.3 and 4.1** | Semantic markup is used to designate headings (<h1>), lists (<ul>, <ol>, and <dl>), emphasized or special text (<strong>, <code>, <abbr>, <blockquote>, for example), etc. Semantic markup is used appropriately. | Semantic markup is not used. |
| 2 | Using WAVE or another tool, view the page with the style sheet turned off.  **508 D** | The document is still readable and understandable (even if it's less visually appealing) when the style sheet is turned off. | The document is confusing or information is missing when the style sheet is turned off. |
| 3 | Run pages through the W3C validator to ensure there are not significant HTML/XHTML validation/parsing error. (http://validator.w3.org/)  **WCAG 4.1** | Validation and parsing errors are minimal. | Validation and parsing errors are significant. |
| 4 | Use the browser to display the text on the page at double the size. **WCAG 1.4** | The page is readable and functional when the text size is doubled. | The page is not readable and functional when the text size is doubled. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Repetitive Navigation Links and Other Content | | | |
| Help users skip repetitive navigation. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Using Firebug, view the source code to ensure that there is a link that lets the user skip over navigation or other repeated content and move to the content main content or that a good heading structure is used.  **WCAG 2.4, 508 (o)** | A link is provided to skip over navigational menus or other lengthy lists of links.  If a page has a proper heading structure, this may be a sufficient replacement for a "Skip to main content" link. Note that navigating by headings is not yet supported in all browsers. | There is no way to skip over repetitive lists of links. |
| 2 | Ensure that navigation is consistently placed across pages. **WCAG 3.2.3** | Navigation is consistently placed across pages. | Navigation is not consistently placed across pages. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Timed Response | | | |
| Users must have enough time to read and use content. | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Manually check the page to see if any content changes without user input. This might include screen refreshes, rotators, etc. **WCAG 1.4 and 2.2, 508 (p)** | A mechanism is provided to stop, pause, mute, or adjust volume for audio that automatically plays on a page for more than 3 seconds. | The user is required to react within limited time constraints. |
| 2 | Manually check to see if the page or application has a time limit, and if so, that the user is able to turn off, adjust, or extend that time limit. **WCAG 2.2, 508 (p)** | User is able to turn off, adjust, or extend the time limit. | User is not able to turn off, adjust or extend the time limit. |
| 3 | Automatically moving, blinking, or scrolling content can be paused, stopped or hidden by the user. **WCAG 2.2** | User is able to pause, stop or hide moving, blinking, or scrolling content. | User is unable to pause, stop or hide moving, blinking or scrolling content. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Text | | | | |
| Provide ways to help users navigate, find content, and determine where they are on the page and in the site. | | | | |
| No. | How to Test: | Pass | Fail |
| 1 | Manually check the page title and ensure it is descriptive and informative.  **WCAG 2.4** | Page title is descriptive and informative. | Page title is redundant, or not descriptive. |
| 2 | Check all links, including form image buttons and image maps, to determine if the link text alone, or the link text and its context, provides enough information for the user to know where it will take them.  **WCAG 2.4** | Purpose of each link can be determined from the link text alone. | Link text does not provide sufficient information about where it goes. |
| 3 | Check page headings and labels for form and interactive controls to ensure they are informative.  **WCAG 2.4** | Page headings and form labels are meaningful. | Page headings and form labels are not meaningful. |
| 4 | View the source code to determine if the language of the page is identified using the HTML lang attribute (<html lang="en">, for example).  **WCAG 3.1** | Language is identified. | Language is not identified. |
| 5 | Resize text using browser controls to 200 percent.  **WCAG 1.4.4** | The user should be able to increase webpage text to 200 percent. | The user cannot increase webpage text to 200 percent. |

# References

* Section 508 (https://www.section508.gov/)
* WebAIM's WCAG 2.0 Checklist (http://webaim.org/standards/wcag/checklist)