ICANN Test Protocol (keyboard, images & forms)

Manual accessibility assessment

February 3rd, 2016

simply accessible.

Table of contents

KIF Testing protocol

This document provides guidance on accessibility testing for **keyboard** access, **images** and **forms**. It is meant to provide ICANN teams with a working knowledge on how to conduct basic accessibility testing that will enable them to find some of the most prevalent issues that can impact user experience for people with disabilities.

This document is not meant to provide an exhaustive list of every single possible test that a tester can do. Rather, it highlights some of the most important aspects to consider, so ICANN testers can build a solid foundation with accessibility using testing techniques that will be consistent with the web accessibility testing protocols used at Simply Accessible.

Each requirement maps to the appropriate success criterion from WCAG 2.0, and it also links to a particular browser-based testing tool where appropriate.

Keyboard access

The following requirements need to be tested for when looking at potential keyboard access issues.

KEYBOARD FOCUS IS NOT APPLIED TO ITEMS WHICH ARE NOT ACTIONABLE

Using only your keyboard, move from one active element to the next using only the tab key. Make sure that for each tab stop, you can set the focus to an object that can be interacted with on the page.

Log an issue for each situation where the keyboard focus is set to an object that cannot be interacted with on the page.

- **Success Criterion:** 2.4.3 - Focus order (level A)
- **Tool used:** keyboard.

ALL ACTIONABLE ELEMENTS CAN RECEIVE KEYBOARD FOCUS

Using your mouse, go over each object in the page, to identify which ones trigger a call to action. Then, using your keyboard's TAB key only, make sure that each identified element can also receive keyboard focus.

Log an issue for each situation where an actionable object does not receive keyboard focus.

- **Success Criterion:** 2.1.1 - Keyboard (level A)
- **Tool used:** keyboard.

ALL CONTENT CAN BE REACHED USING ONLY A KEYBOARD

Using only the keyboard's Tab key, and the arrow keys in the case of non-native widgets, make sure that all of the focusable content in the page can be reached using only the keyboard.

Log an issue for each situation where focusable content can only be reached by first having to rely on a mouse click to either set focus to an area, or dismiss an object.

- **Success Criterion:** SC 2.1.1 - Keyboard (level A)
- Tool used: Keyboard

USERS CAN NAVIGATE AWAY FROM ANY ELEMENT USING ONLY A KEYBOARD

Using only the keyboard's Tab key, make sure that the focus never gets trapped inside an object in the page. Look for any situation where navigation with the keyboard becomes impossible, due to having set focus on an object that traps the focus. Users will also expect to be able to navigate backwards using Shift + Tab, and to close modal dialogs and similar controls using the Escape key.

Log an issue for each situation where the focus becomes trapped in an object and further navigation becomes impossible as a result.

- **Success Criterion:** SC 2.1.2 - No keyboard trap (level A)
- **Tool used:** keyboard.

ACTIVE ELEMENTS CAN BE TABBED THROUGH IN A PREDICTABLE AND LOGICAL ORDER

Using only the keyboard's TAB key, go from one active element to the next in the page, and validate that the order in which these objects get focus is both predictable and logical. Users will also expect to be able to navigate backwards in the reverse order using SHIFT + TAB.

Log an issue for each situation where the keyboard focus randomly jumps from one section of the page to the next, or when the order in which objects get focus in a specific section might result in confusion for the end users.

- **Success Criterion:** 2.4.3 - Focus order (level A)
- **Tool used:** keyboard.

KEYBOARD FOCUS DISPLAYS A VISIBLE FOCUS INDICATOR FOR EVERY ACTIVE ELEMENT

Using only the keyboard's TAB key, go from one active element to the next in the page, and validate that for each tab stop, a clearly visible focus indicator is provided to help users identify where the keyboard focus is currently set.

Log an issue for each situation where the keyboard focus is lost as a result of the object getting focus not being clearly indicated as having keyboard focus.

- **Success Criterion:** 2.4.7 - Focus visible (level AA)
- **Tool used:** keyboard.

KEYBOARD FOCUS DOES NOT AUTO-ADVANCE FROM ONE FIELD TO THE NEXT

Using only the keyboard's TAB key, go through each for control (text field, select element, checkbox, etc.) and validate that entering data into a control does not cause the focus to automatically shift to the next form control.

Log an issue for each situation where the focus auto-advances from one field to the next as a result of interacting with a control.

- Success Criterion: 3.2.2 - On input (level A)
- Tool used: Keyboard

TABINDEX ATTRIBUTES, WHEN USED, ARE NOT ASSIGNED POSITIVE INTEGER VALUES

Using the Web Developer toolbar, go to the Information tab and select the "Display Tab Index" option.

Log an issue for each situation where an object is assigned a tabindex attribute with a positive value (a value of 0 and above).

- Success Criterion: 2.4.3 - Focus order (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

BEHAVIORS SCRIPTED TO OCCUR ON HOVER STATE (ONMOUSEOVER, :HOVER) ARE ALSO REPLICATED ON FOCUS STATE (ONFOCUS, :FOCUS)

Using your mouse, go over each object in the page, to identify which ones trigger a change in behaviour when hovered (an image swapped, a fly out menu revealed, a tooltip displayed, etc.). Then, using your keyboard's TAB key only, make sure that each identified change in behaviour is also provided with the keyboard as focus is set to the object.

Log an issue for each situation where an object that undergoes a change of behaviour when hovered with a mouse does not receive an equivalent change when keyboard focus is set to it.

- Success Criterion: 2.1.1 - Keyboard (level A)
- Tool used: Keyboard

USERS CAN NAVIGATE TO, THROUGH, AND PAST ALL EMBEDDED CONTENT

Using only the keyboard's TAB key, make sure that any embedded object (such as a Flashbased media player or animation, an image, an iframe, etc.), can be reached and tabbed through without causing the keyboard focus to become stuck in the object. Also make sure the keyboard focus can also be set out of the same object without having to rely on a mouse click.

Log an issue for each situation where the focus becomes trapped in an embedded object and further navigation becomes impossible as a result.

- **Success Criterion:** SC 2.1.2 - No keyboard trap (level A)
- Tool used: Keyboard

ALL ACTIVE ELEMENTS CAN BE TRIGGERED, USING ONLY A KEYBOARD

Using only the keyboard's TAB key, make sure that each active element in the page that can be triggered by a mouse click can also be triggered just by relying on the keyboard.

Log an issue for each situation where a call to action can only be operated through the use of a mouse.

- **Success Criterion:** SC 2.1.1 - Keyboard (level A)
- **Tool used:** keyboard.

SIMULATED CONTROLS, SIMULATED DIALOGS, CALENDAR CONTROLS, EMBEDDED MEDIA CONTENT, MENUS AND OTHER ACTIONABLE DYNAMIC CONTENT CAN BE ACCESSED, OPERATED, AND CLOSED FROM THE KEYBOARD

For every situation where a custom form control is created using HTML, CSS and JavaScript instead of relying on a native HTML form element (like creating a checkbox using a div element to align to specific internal design guidelines), make sure that all expected behaviours from the control are also recreated in script for the custom control.

Log an issue for every situation where the expected functionalities of a control are not provided for users who rely only on the keyboard.

- **Success Criterion:** 2.1.1 - Keyboard (level A)
- Tool used: Keyboard

Images

The following requirements need to be tested for when looking at potential image issues.

INFORMATIVE IMAGES ARE DESCRIBED WITH A TEXT EQUIVALENT

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate every informative image, and validate that it provides an alt text value. Also look for informative images that do not get a tooltip (background images).

Log an issue for each situation where an informative image (with or without a tooltip) is not provided with an alt text value.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

INFORMATIVE IMAGES PROVIDE MEANINGFUL EQUIVALENT TEXT

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate every informative image, and validate that it provides an alt text value that provides useful and meaningful content.

Log an issue for each situation where an informative image is provided with an alt text value that does not provide information that will be meaningful to the user (like filler words such as "spacer", or information such as the image's filename), or no alt text value is present at all.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- Tool used: Web Developer toolbar (available for Firefox or Chrome).

TEXT EQUIVALENTS FOR INFORMATIVE IMAGES ARE BOTH CLEAR AND INFORMATIVE

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate every image that conveys information, and validate that it provides an alt text value that conveys the same information as the image itself. Also look for informative images that do not get a tooltip (background images).

Log an issue for each situation where an informative image (with or without a tooltip) is not provided with an alt text value that is descriptive of the information displayed in the image.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

IMAGES PRIMARILY CONVEYING FUNCTION USE THE ALT TEXT VALUE TO DESCRIBE THEIR PURPOSE, RATHER THAN WHAT THEY LOOK LIKE

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate all images that function as link content, button content, or provide another function, and validate that each one provides an alt text value that conveys information about the image's purpose rather than what the image looks like. Also look for functional images that do not get a tooltip (background images).

Log an issue for each situation where a functional image (with or without a tooltip) is not provided with an alt text value that is descriptive of the image's purpose.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

NO IMPORTANT IMAGES NEEDING TEXT EQUIVALENTS ARE PLACED AS CSS BACKGROUND IMAGES

Using the Web Developer toolbar, go to the Images tab and select the "Hide Background Images" option. All images embedded as background images are removed from the page. Identify whether any important information is lost as a result of running this test.

Log an issue for each image conveying information that is embedded using CSS.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

ICONS THAT CONVEY INFORMATION ARE USED THE SAME WAY SITE-WIDE

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate all icons that are intended to provide information to users. Verify that these icons are used consistently across the page, and consistently with how they are used elsewhere on the site.

Log an issue for each situation where an information icon is used in a way that is inconsistent with use elsewhere on the site.

- **Success Criterion:** 3.2.4 Consistent Identification (level AA)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

THE ALT TEXT USED FOR IMAGES WITH TEXT IN THEM AND/OR REPRESENTING TEXT INCLUDES ALL RELEVANT TEXT FOUND IN THE IMAGE

Using the Web Developer toolbar, go to the Images tab and select the “Display Alt Attributes” option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate every image that has text in it, and validate that it provides an alt text value that provides useful and meaningful content in relation to the text of the image.

Log an issue for each situation where an informative image that contains text is provided with an alt text value that does not provide equivalent information that will be meaningful to the user.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

"NULL" ALT ATTRIBUTES (USING ALT="") ARE USED FOR DECORATIVE IMAGES WHICH DO NOT PROVIDE INFORMATION

Using the Web Developer toolbar, go to the Images tab and select the “Outline Images with Empty Alt Attributes” option. All images that have an alt attribute with an empty value will display a border. Locate every image that has an empty alt value, and validate that the image does *not* provides useful and meaningful content to the user.

Log an issue for each situation where a decorative image has an alt text value that is not empty, or an alt attribute is not present.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

NULL ALT ATTRIBUTES (USING ALT="") ARE USED FOR IMAGES THAT ARE ALREADY DESCRIBED IN TEXT IN ADJACENT PAGE CONTENT

Using the Web Developer toolbar, go to the Images tab and select the "Outline Images with Empty Alt Attributes" option. All images that have an alt attribute with an empty value will display a border. Locate every image that has an empty alt value, and validate that if the image conveys information, that information is provided in adjacent text on the page.

Log an issue for each situation where an informational image has an alt text value that is empty where there is not adjacent text that provides the image's information.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

ADJACENT LINKED IMAGES AND TEXT LINKS POINTING TO THE SAME URL ARE MERGED INTO SINGLE LINKS

Using the Web Developer toolbar, go to the Information tab and select the "View Link Information" option. A list of all the links available in the page is displayed. Locate every situation where two links pointing to the exact same URL follow one another, and validate whether they are adjacent in the content.

Log an issue for every situation where a pair of identical, adjacent links are made up of one image link and one text link.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

IMAGES THAT ARE CHANGED OR REPLACED USING JAVASCRIPT HAVE ALT TEXT VALUES THAT ARE UPDATED AS THE IMAGES CHANGE

Locate every image that is dynamically updated in the page, either as a result of user interaction, or a specific functionality or behaviour. Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. The alt text value of each foreground image is displayed. Make sure the alt text value of dynamically updated images changes as the images change.

Log an issue for each situation where an image changes, but the alt text value remains the same.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

ANY CHANGE OF CONTEXT FOR AN IMAGE ALSO UPDATES ITS TEXT EQUIVALENT TO REPRESENT THE CHANGE

Locate every image that is dynamically updated in the page, either as a result of user interaction, or a specific functionality or behaviour. Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. The alt text value of each foreground image is displayed. Make sure the dynamically updated alt text value of images remains descriptive of the images displayed.

Log an issue for each situation where a change of context change the meaning of an image, but the alt text value remains the same.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

ALTERNATE TEXT FOR ICONS, IMAGE BUTTONS, OR OTHER NAVIGATIONAL ELEMENTS CLEARLY DESCRIBES THE DESTINATION OR ACTION

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate all call-to-action images (those that form the content of icons, buttons, or links), and validate that each one provides an alt text value that conveys information about the call-to-action's destination or purpose. Also look for call-to-action images that do not get a tooltip (background images).

Log an issue for each situation where a call-to-action image (with or without a tooltip) is not provided with an alt text value that is descriptive of the call-to-action's destination or purpose.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

THE PAGE CONTAINS NO DIRECT LINKS TO IMAGE FILES

Using the Web Developer toolbar, go to the Information tab and select the "View Link Information" option. A list of all the links available in the page is displayed.

Log an issue for each situation where a URL points directly to an image file format (png, svg, jpg, , jpeg, gif, tiff, bmp, etc).

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

ALT ATTRIBUTES ARE USED FOR ALL AREAS OF AN IMAGE MAP

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate every part of any image map, and validate that it provides an alt text value that provides useful and meaningful content.

Log an issue for each situation where a part of an image map is provided with an alt text value that does not provide information that will be meaningful to the user (like filler words such as "spacer", or information such as the image's filename), or no alt text value is present at all.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

ALTERNATE MEANS OF ACCESSING CAPTCHA INFORMATION ARE PROVIDED (I.E. AUDIO CAPTCHA, LOGICAL QUESTION, ETC.)

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate CAPTCHA images and validate that they provide an alt text value that matches the content of the CAPTCHA.

Additionally, locate options for solving the CAPTCHA without being able to see the image or access the alt text. Look for options to play an audio file, check a box, fill out a math problem, or other means. If the CAPTCHA has interactive controls, also perform the Keyboard verifications included here. If the CAPTCHA includes form elements, such as text fields or checkboxes, also perform the Form verifications included here.

Log an issue for each situation where a CAPTCHA image is present with an alt text value that does not provide information that will be meaningful to the user (like filler words such as "spacer", or information such as the image's filename), or no alt text value is present at all.

Log an issue if there is no additional accessible way to solve the CAPTCHA.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

IMAGES THAT FLASH OR BLINK DO NOT DO SO FOR MORE THAN 3 TIMES PER SECOND

For flashing or blinking graphics, using a stopwatch or timer, count the number of times a graphic flashes or blinks for a number of seconds. Calculate the number of blinks per second.

Log an issue if a graphic blinks more than 3 times per second.

- **Success Criterion:** 2.3.3 - Three Flashes or Below Threshold (level A)
- **Tool used:** Stopwatch or timer

CHARTS, GRAPHS, AND INFOGRAPHICS DO NOT RELY ON COLOUR ALONE TO CONVEY INFORMATION

Navigate to the chart, graph, or infographic. Examine how information is conveyed with colour. Locate any information that is shown that relies on a colour alone to indicate size, importance, a specific value, etc.

- **Success Criterion:** 1.4.1 - Use of Colour (level A)
- Tool used: None

Log an issue if the graphic provides information that is only conveyed with colour without a label, a pattern, or another way of distinguishing values and meanings.

THE PURPOSE OF COMPLEX IMAGES (CHARTS, GRAPHS, INFOGRAPHICS, DIAGRAMS) IS BRIEFLY DESCRIBED USING A DESCRIPTIVE ALT TEXT VALUE

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate alt values for complex images and verify that a brief description of the purpose of the graphic is included as an alt value.

Log an issue for each situation where a complex image has no alt attribute, a very long alt attribute, or an alt attribute that does not provide a short description of the image's content.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

THE FULL DESCRIPTION OF COMPLEX IMAGES IS PROVIDED THROUGH THE LONGDESC ATTRIBUTE OR ADJACENT TEXT

Navigate to the chart, graph, or infographic. Using the browser Web Inspector, examine the complex image's attributes. Verify if a longdesc attribute is provided, and if it points to a URL. Verify that the URL is correct, and that the information provided at that URL sufficiently describes the content of the graphic. Alternately, verify that there is a text description of the graphic provided adjacent to the graphic on the original page.

Log an issue for each situation where a complex image has no longdesc or an adjacent description provided.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Inspector (any browser)

THE ALT TEXT VALUE OF IMAGES AND LINKED IMAGES DOES NOT REPLICATE ANY INFORMATION THAT IS ALREADY BEING CONVEYED BY SCREEN READER TECHNOLOGY

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. All images embedded as foreground images will display a tooltip with their respective alt text values. Locate every informative image, and validate that its alt text is not the same as information provided in regular text on the page.

Log an issue for each situation where an image duplicates content that is provided elsewhere on the page.

- **Success Criterion:** 1.1.1 - Non-text Content (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

Forms

The following requirements need to be tested for when looking at potential forms issues.

RELATED FORM FIELDS ARE GROUPED TOGETHER VISUALLY

Locate any forms elements on the page. Verify that form fields that are collect related information are visually grouped together.

Log an issue for each situation where related form fields are not presented together visually in the form.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Visual check

LONG OR COMPLEX FORMS ARE BROKEN UP BY HEADINGS

Using the Web Developer toolbar, go to the Outline tab and select the "Outline headings" option. All headings on the page will be outlined. Locate any long or complex forms, and verify that related fields of the form are broken up by headings that describe the purpose of that section of the form, at the appropriate heading level.

Log an issue for each situation where a long or complex from does not have headings used to describes related parts of the form.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

WHENEVER PROVIDED, INSTRUCTIONS ON HOW TO USE FORMS ARE CLEARLY EXPLAINED

Navigate to any form content on the page. Location any information that instructs users about how to fill out the form. Verify that the instructions are correct, clearly explained, and provide sufficient information for entering information correctly in the form.

- **Success Criterion:** 3.3.2 Labels or Instructions (level A)
- **Tool used:** Visual check

INSTRUCTIONS PROVIDED IN FORMS DO NOT RELY ON SENSORY CHARACTERISTICS

THEMSELVES

Using the Web Developer toolbar, go to the CSS tab and select the "Disable All Styles" option. Navigate to form content on the page, and verify that instructions are understandable without sensory characteristics provided by styles.

Log an issue for each situation where users must perceive a shape, colour, size, visual location, orientation, sound, or similar to access form instructions.

- **Success Criterion:** 1.3.3 Sensory Characteristics (level A), 1.4.1 Use of Colour (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome)

THE CONTENT OF INSTRUCTIONS PROVIDED IN FORMS DO NOT RELY ON SENSORY CHARACTERISTICS

Navigate to any form content on the page. Locate any instructions that provide users with information about how to fill out the form. Verify that understanding instructions does not rely on sensory input, such as shape, colour, size, visual location ("see to the right"), orientation ("left-pointing arrow"), or sound.

Log an issue for each situation where understanding instructions requires perceiving a shape, colour, size, visual location, orientation, sound, or similar to understand form instructions.

- **Success Criterion:** 1.3.3 Sensory Characteristics (level A), 1.4.1 Use of Colour (level A)
- **Tool used:** Visual check

TEXT THAT IS RELEVANT TO A FORM DOES NOT APPEAR VISUALLY AFTER THE SUBMIT BUTTON

Locate any form on the page. Verify that no information that impacts how the form is understood appears after the button to submit the form.

Log an issue for each situation where information about the form appears visually after the form.

- **Success Criterion:** 1.3.2 Meaningful Sequence (level A)
- **Tool used:** Visual check

TEXT THAT IS RELEVANT TO A FORM DOES NOT APPEAR PROGRAMMATICALLY AFTER THE SUBMIT BUTTON

Using the Web Developer toolbar, go to the CSS tab and select the “Disable All Styles” option. Navigate to form content on the page, and verify that no information that impacts how the form is understood appears after the button to submit the form.

Log an issue for each situation where information about the form appears visually after the form.

- **Success Criterion:** 1.3.2 Meaningful Sequence (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome)

STANDARD HTML CONTROLS ARE USED FOR FORMS

Using the Web Developer toolbar, go to the Forms tab and select “Check All Checkboxes”, “Expand All Select Elements”, and “Populate Form Fields”. This will check check boxes, expand select elements, select a radio button in a set, and fill in all available text fields and textareas. Verify visually that all fields have been modified in some way.

Log an issue for each form field that is not affected.

- **Success Criterion:** 4.1.2 Name, Role, Value (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome)

EVERY FORM CONTROL HAS A UNIQUE LABEL THAT IS VISUALLY CONVEYED

Navigate to form content on the page. Verify that each form control has a unique, visible label that describes the field. Alternately, if the form has fields and labels that repeat for multiple situations, such as contact information for multiple passengers, verify that each section that repeats has a unique heading that helps distinguish between the repeated labels.

Log an issue for each form control that has no visible, unique label, or that does not otherwise provide a visual cue, such as a related heading, to help differentiate between fields with the same label.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Visual check

FORM LABELS ARE CLEAR AND INFORMATIVE

Navigate to form content on the page. Verify that each visible label provides sufficient information to enter information or make a selection with each field, such as the type of information expected and how it should be formatted.

Log an issue for each form field that has an unclear label.

- **Success Criterion:** Headings and Labels (level AA)
- **Tool used:** Visual check

TEXT LABELS ARE MARKED UP USING THE LABEL ELEMENT

Using the Web Developer toolbar, go to the Forms tab and select the "Outline Form Fields Without Labels" option. Navigate to form content on the page, and verify that no form field is outlined.

Log an issue for each form field that lacks a label.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome)

PLACEHOLDER ATTRIBUTE VALUES ARE NOT USED IN LIEU OF REGULAR FORM LABEL ELEMENTS

Using the Web Developer toolbar, go to the Forms tab and select the "Outline Form Fields Without Labels" option. Navigate to form content on the page. Verify that any fields that are outlined do not rely solely on a placeholder within the field to provide instruction to the user.

Log an issue for each text field that displays lacks a label and relies on a placeholder to instruct users about how to enter information in the field.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome)

TEXT LABELS AND FORM CONTROLS ARE PROGRAMMATICALLY ASSOCIATED USING FOR AND ID ATTRIBUTES

Navigate to form content on the page. Click on each form label. Verify that keyboard focus is moved to the associated field when the label is clicked.

Log an issue for each form field that does not send focus to the field when the label is clicked.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- Tool used: Mouse

WHENEVER PROVIDED, INSTRUCTIONS ON HOW TO USE FORMS ARE PROGRAMMATICALLY CONVEYED TO ASSISTIVE TECHNOLOGIES

Navigate to any form content on the page. Locate any instructions that provide users with information about how to fill out the form. Verify that understanding instructions does not rely on sensory input, such as shape, colour, size, visual location ("see to the right"), orientation ("left-pointing arrow"), or sound.

Log an issue for each situation where instructions follow related form fields,

- **Success Criterion:** 1.3.3 Sensory Characteristics (level A)

RADIO BUTTONS GROUPS ARE CLEARLY GROUPED TOGETHER VISUALLY

Locate radio button groups on the page. Validate that radio buttons do not appear alone, but in groups of two or more whenever all related radio buttons are near to each other.

Log an issue for each situation where radio buttons that make up a group are not presented together visually.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Visual check

RADIO BUTTONS ARE PROPERLY GROUPED VIA HTML

Using the Web Developer toolbar, go to the Forms tab and select the "View Form Information" option. Information about all form fields on the page will be listed out, grouped by form. For each form field that has a type of *radio*, verify that the radio button shares a *name* with other radio buttons in its set.

Log an issue for each radio button that does not have a shared name.

- **Success Criterion:** 4.1.2 Name, Role, Value (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

RADIO BUTTON AND CHECKBOX LABELS ARE POSITIONED TO THE RIGHT OF THE FORM CONTROL

Locate radio button groups and checkboxes on the page. Validate that labels are visually positioned to the right of their associated radio button or checkbox.

Log an issue for each situation where a label is displayed to the left of a radio button or checkbox.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Visual check

GROUPINGS ARE PROVIDED FOR LONG OPTION LISTS IN SELECT ELEMENTS USING THE OPTGROUP ELEMENT

Using the Web Developer toolbar, go to the Forms tab and select the "Expand Select Elements" option. Select elements should display their full list of options. Verify that select elements that contain a large number of options that are difficult to distinguish are broken up by option group labels.

Log an issue for each select element that has a very long list of options that are difficult to distinguish without option groups.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

FORM FIELD CONSTRAINTS ARE CLEARLY DISCLOSED IN TEXT

Using the Web Developer toolbar, go to the Forms tab and select the "View Form Information" option. Information about all form fields on the page will be listed out, grouped by form. For each form field that has a *maximum length*, verify that this information is conveyed in the form label or other relevant instructions.

Log an issue for each field that has a maximum length that is not conveyed to users.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome).

CUSTOM FORM CONTROLS REPLICATE ALL INHERENT BEHAVIORS OF NATIVE HTML FORM CONTROLS

Using the Web Developer toolbar, go to the Forms tab and select "Check All Checkboxes", "Expand All Select Elements", and "Populate Form Fields". This will check check boxes, expand select elements, select a radio button in a set, and fill in all available text fields and textareas. Verify visually that all fields have been modified in some way. For fields that have not been modified, test keyboard accessibility and verify that the field has a visual and programmatic label.

Log an issue for each custom form control that does not work like the native HTML control it is mimicking.

- **Success Criterion:** 4.1.2 Name, Role, Value (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome), keyboard

REQUIRED FIELDS ARE VISUALLY INDICATED

Navigate to form content on the page. If the form has a mix of required and optional fields, verify that the labels of fields that are required display an icon indicating the required status. Verify that a note appears at the top of the form indicating that the form has required fields, and that the icon is how users can identify these fields.

Log an issue for each field that is required that does not indicate that it is required.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Visual check

REQUIRED FIELDS ARE PROGRAMMATICALLY INDICATED AS SUCH TO ASSISTIVE TECHNOLOGY

Using the Web Developer toolbar, go to the Images tab and select the "Display Alt Attributes" option. Alt attributes for all images on the page will be displayed. Navigate to form content on the page, and verify that icons that convey required status for fields have an alt value of "Required". Verify that the icons are implemented as part of the form label.

Log an issue for each required field icon that does not have a text alternative.

- **Success Criterion:** 1.3.1 Info and Relationships (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome)

USERS HAVE THE ABILITY TO PREVENT ERRORS ON FORMS

Navigate to a form. Verify that users are provided with enough information to submit the form without errors.

Log an issue for each form that does not provide sufficient information to complete it successfully.

- **Success Criterion:** 3.3.2 Labels or Instructions (level A)
- **Tool used:** Visual check

FORMS ARE NOT AUTOMATICALLY SUBMITTED

Navigate to a form and enter valid data into it. Verify that the form is not automatically submitted at any time during data entry or editing, and requires the user to interact with a button to submit it.

Log an issue for each form that submits automatically without the user explicitly submitting the form.

- **Success Criterion:** 3.2.2 On Input (level A)
- **Tool used:** Visual check, keyboard

SUBMIT BUTTONS IN FORMS RELY ON AN INPUT TYPE SUBMIT ELEMENT OR A BUTTON ELEMENT

Using the Web Developer toolbar, go to the Forms tab and select the "Display Form Details" option. Details about all form elements on the page will be displayed. Verify that the control to submit the form uses a button element, or an input element with type="submit".

Log an issue for each form that does not have a button or submit input.

- **Success Criterion:** 3.2.2 On Input (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome)

USERS HAVE THE ABILITY TO CORRECT ERRORS ON FORMS

Navigate to a form. Submit the form with errors. Verify that users are provided with the opportunity to correct those errors.

Log an issue for each form that does not provide the opportunity to correct errors.

- **Success Criterion:** 3.3.3 Error Suggestion (level AA)
- **Tool used:** Visual check

WHEN ERRORS ARE RETURNED ON SUBMIT, FOCUS IS VISUALLY SET DIRECTLY TO THE ERROR MESSAGE

Navigate to a form. Submit the form with data missing or entered incorrectly. Verify that errors are returned, and that a visible error message is displayed.

Log an issue for each form that does not visually display an error message.

- **Success Criterion:** 3.3.1 Error Identification (level A)
- **Tool used:** Visual check

WHEN ERRORS ARE RETURNED ON SUBMIT, FOCUS IS PROGRAMMATICALLY SHIFTED DIRECTLY TO THE ERROR MESSAGE

Navigate to a form. Submit the form with data missing or entered incorrectly. Verify that errors are returned. Press the Tab key. Verify that focus moves to the first focusable element after the error message.

Log an issue for each form that does not programmatically shift focus to an error message.

- **Success Criterion:** 3.3.1 Error Identification (level A)
- Tool used: Keyboard

WHEN ERRORS ARE RETURNED, CLEAR INFORMATION IS PROVIDED REGARDING ANY DETECTED ERRORS

Navigate to a form. Submit the form with data missing or entered incorrectly. Verify that errors are returned. Verify that that the user is notified of errors. Verify that the information provided is as clear as possible to correct the errors.

Log an issue for each error message that does not provide clear information to fix the associated errors.

- **Success Criterion:** 3.3.3 Error Suggestion (level AA)
- **Tool used:** Visual check

ERRORS AND ALERTS ARE VISUALLY INDICATED THE SAME WAY SITE-WIDE

Navigate to a form. Submit the form with data missing or entered incorrectly. Verify that errors are returned. Verify that that the user is notified of errors. Verify that the way error notification occurs is visually consistent across the page and on forms of this kind across the site.

Log an issue for each error message that is visually inconsistent with the rest of the site.

- **Success Criterion:** 3.2.4 Consistent Identification (level AA)
- **Tool used:** Visual check

ERROR MESSAGES AND ALERTS ARE PROGRAMMATICALLY INDICATED THE SAME WAY ACROSS THE SITE

Navigate to a form. Submit the form with data missing or entered incorrectly. Verify that errors are returned. Verify that that the user is notified of errors. Verify that the way error notification occurs is programmatically consistent across the page and on forms of this kind across the site.

Log an issue for each error message that is programmatically inconsistent with the rest of the site.

- **Success Criterion:** 3.2.4 Consistent Identification (level AA)
- **Tool used:** Visual check, keyboard, mouse

CUES ARE PROVIDED TO HELP USERS CORRECT ERRORS

Navigate to a form. Submit the form with data missing or entered incorrectly. Verify that errors are returned. Verify that that the user is notified of errors. Verify that the user is given a way to navigate to fields that have errors, such as a list of descriptive links that jump to each field with an error.

Log an issue for each error that lacks a cue for user correction.

- **Success Criterion:** 3.3.3 Error Suggestion (level A)
- **Tool used:** Visual check, keyboard

ERROR MESSAGING VISUALLY DISPLAYS NEXT TO THE FORM FIELD

Navigate to a form. Submit the form with data missing or entered incorrectly. Verify that errors are returned. Verify that an error message appears next to each field that has an error, or next to each group of fields that share an error.

Log an issue for each field with an error that does not have a visible adjacent error message

- **Success Criterion:** 3.3.2 Labels or Instructions (level A)
- **Tool used:** Visual check

INSTRUCTIONS TO FIX FORM ERRORS ARE EXPLAINED IN TEXT INSIDE THE CORRESPONDING LABEL ELEMENT

Navigate to a form. Submit the form with data missing or entered incorrectly. Verify that errors are returned. Verify that an error message appears next to each field that has an error. Using the Web Developer toolbar, go to the Forms tab and select the "View Form Information" option. Information about all form fields on the page will be listed out, grouped by form. For each form field that displayed an error, verify that the label include the associated error message text

Log an issue for each field that has an error displayed but which does not have the error text included in its label.

- **Success Criterion:** 3.3.2 Labels or Instructions (level A)
- **Tool used:** Web Developer toolbar (available for Firefox or Chrome)

A CONFIRMATION SCREEN IS DISPLAYED PRIOR TO ANY FINAL FORM SUBMISSION

Navigate to a form. Fill out the form with incorrect data and submit it. If the form does not return errors, verify that the user is given the opportunity to verify the data they have entered before the data is submitted.

- **Success Criterion:** 3.3.4 Error Prevention (Legal, Financial, Data) (level AA)
- **Tool used:** Visual check

NO RESPONSES REQUIRE A SPECIFIED, UNCHANGEABLE PERIOD OF TIME TO ANSWER

Navigate to a form. If the form or specific fields in the form have a time limit for responding, verify that *at least one* of the following is true:

1. There is an accessible way for users to **turn off** the time limit before they encounter it.
2. There is an accessible way for users to **adjust the time limit** before they encounter it, allowing them at least ten times the normal amount of time.
3. There is an accessible, easy way for users to **extend the time limit** to at least ten times the normal amount before time expires, and they are given at least 20 seconds to do so.
4. The time cannot be extended because it is a **real-time event or extending it would invalidate** what the user is trying to accomplish.

5. The time limit is **longer than 20 hours**.

    - **Success Criterion:** 2.2.1 Timing Adjustable (level A)
    - **Tool used:** Visual check