

APPENDIX A

Accessible Heuristics

ACCESSIBILITY HEURISTICS, V1.4

10 General Rules of thumb for Accessible Design¹

Table A-1.

INTERACTION METHODS AND MODALITIES	Users can efficiently interact with the system using the input method of their choosing (i.e., mouse, keyboard, touch, etc.).
NAVIGATION AND WAYFINDING	Users can easily navigate, find content, and determine where they are at all times within the system.
STRUCTURE AND SEMANTICS	Users can make sense of the structure of the content on each page and understand how to operate within the system.

(continued)

¹Accessibility Heuristics Denis Boudreau, Aparna Pasi, Caitlin Geaier <https://www.24a11y.com/2018/unlocking-accessibility-for-ux-ui-designers/> - https://drive.google.com/file/d/1QkURByXUk4NOt17jw6VtyCui4_Jj1P6d/edit

APPENDIX A: ACCESSIBLE HEURISTICS

Table A-1. *(continued)*

ERROR PREVENTION AND STATES	Interactive controls have persistent, meaningful instructions to help prevent mistakes and provide users with clear error states which indicate what the problems are and how to fix them whenever errors are returned.
CONTRAST AND LEGIBILITY	Text and other meaningful information can be easily distinguished and read by users of the system.
LANGUAGE AND READABILITY	Content on the page can easily be read and understood by users of the system.
PREDICTABILITY AND CONSISTENCY	The purpose of each element is predictable, and how each element relates to the system as a whole is clear and meaningful, to avoid confusion for the users.
TIMING AND PRESERVATION	Users are given enough time to complete their tasks and do not lose information if their time (i.e., a session) runs out.
MOVEMENT AND FLASHING	Elements on the page that move, flash, or animate in other ways can be stopped, and do not distract or harm the users.
VISUAL AND AUDITORY ALTERNATIVES	Purely visual or auditory content that conveys information has text-based alternatives for users who can't see or hear.

A11Y Style Guide²

Table A-2.

Basic Card	<ul style="list-style-type: none"> • Make sure you have tabbing focus indicators for all elements that need to be highlighted. • Every <code></code> you add to your site needs to have an alt attribute. If the image is informational, set the <code>alt</code> equal to a <i>descriptive</i> alternative for that image. • If the image is decorative or redundant to adjacent text, set <code>alt=""</code>, which conveys to assistive technology users that the image isn't necessary for understanding the page. • Avoid using generic strings like photo, image, or icon as alt values, as they don't communicate valuable content to the user. Be as descriptive as possible. You can add <code>class="visuallyhidden"</code> with descriptive text to give more context to a button or link's purpose.
------------	---

(continued)

²A11Y Style Guide <https://a11y-style-guide.com/style-guide/>

Table A-2. *(continued)*

Forms— Checkboxes	<ul style="list-style-type: none"> • The <code><fieldset></code> surrounds the entire grouping of checkboxes or radio buttons. The <code><legend></code> provides a description for the grouping. • Some assistive technology reads the legend text for each fieldset, so it should be brief and descriptive. This helps someone using assistive technology to understand the question they are answering with the group of checkboxes. • WAI-ARIA provides a grouping role that functions similarly to <code>fieldset</code> and <code>legend</code>. In option #2 the <code>div</code> element has <code>role=group</code> to indicate that the contained elements are members of a group and the <code>aria-labelledby</code> attribute references the id for text that will serve as the label for the group. Note: This method is not supported by all browser/AT devices.
Buttons	<ul style="list-style-type: none"> • A <code><button></code> tag does not need anything special to work. Use <code><button></code> when you can, but it is possible to use other elements as long as you add <code>role="button"</code> and add JavaScript to replicate the button functionality. • Just like links, you can add <code>class="visuallyhidden"</code> with descriptive text to give more context to the button's purpose. • If a button contains an <code></code> element, make sure to set its <code>alt</code> attribute. If it contains an icon, use <code>aria-label</code> to describe the icon instead. • You can use <code><input type="image"></code> to make a graphical button. It takes a <code>src</code> and <code>alt</code> attribute just like traditional images. • Button states are important, not just button styling! If you are only toggling classes to visually manage state of your components, you are likely not appropriately conveying that state to users of assistive technologies.

(continued)

Table A-2. *(continued)*

Colors	<ul style="list-style-type: none"> • Avoid using color <i>only</i> to communicate information. In the case of links, use another identifier such as bold or underline to indicate a link vs. using color alone. • Some users have difficulty reading text if there is too little contrast between foreground and background. To meet Level AA, text must have a contrast ratio of at least 4.5:1 (or 3:1 for large text). In order to meet the guidelines at the stricter Level AAA, the contrast ratio must be at least 7:1 (or 4.5:1 for large text).
Link Focus	<ul style="list-style-type: none"> • Do not set your site's link focus to <code>outline: none</code>. Never. Ever. • If you have multiple form fields on your site or you see the dreaded <code>outline: none</code> in your base code, you can reset the browser defaults by adding the code: <pre style="margin-left: 20px;">a:focus { outline: auto 2px Highlight; // for non- webkit browsers outline: auto 5px -webkit-focus-ring-color; // for webkit browsers }</pre> • Of course, you can create your own focus styles to match your theme or to make the default browser styles more prominent, just make sure they are visible by tabbing and obvious to your users.

(continued)

Table A-2. *(continued)*

Read More Links	<ul style="list-style-type: none">• Add id selectors to titles or paragraphs and use <code>aria-labelledby=""</code> to link to the title text (Example #1).• Add descriptive text with <code>aria-label=""</code> directly in the link (Example #2).• Add id selectors to titles or paragraphs and use <code>aria-describedby=""</code> to link to the title text (Example #3).• Use the class <code>visuallyhidden</code> to visually hide more information about the link (Example #4).• Turn read more links into buttons, when you can, since they allow for more labeling options.
Skip Links	<ul style="list-style-type: none">• Provide ways for users to skip to important sections of your website. This will help users using screen readers navigate your site easier and more efficiently.• There is no real theming rule when it comes to styling your skip links, as such a lot of websites tend to hide them with <code>class="visuallyhidden"</code> with a special focus attribute so that sighted keyboard only users are able to see them.• It is important to see that the link points to a valid HTML ID, as often this is overlooked in implementation.• On the a11y Style Guide site we have used ‘Jump to main content’, but other valid examples of “skip” links include:

(continued)

Table A-2. *(continued)*

Typography	<ul style="list-style-type: none"> • Avoid small font sizes. • Use relative units for font size, such as ems or rems. While modern browsers can smoothly zoom pixel-based layouts, sizing type in relative units ensures an entire layout can be scaled up or down by simply updating the font size of the body element. • A design should allow typography to be magnified up to 200% by the user without clipping or distorting content. • Select basic, simple, easily readable fonts and use a limited number of fonts. • Avoid small font sizes and use relative units for font size (ems or rems). • Limit the use of font variations such as bold, italics, and ALL CAPITAL LETTERS (caps are similar to screaming to screen readers). • Don't rely only on the appearance of the font (color, shape, font variation, placement, etc.) to convey meaning. • Avoid blinking or moving text. • Use real text rather than text within graphics.
Audio	<ul style="list-style-type: none"> • Build your media with accessibility in mind! It is much easier to work accessibility into the beginning than trying to tack it on later. This is true of all components, but especially for media components. • Make sure your player is accessible and includes control elements to pause, stop, and play your media. • Do not auto-play your media. This can cause confusion as well as annoyance. • Make sure your media has alternative methods to get to the content. For example, add transcripts of your audio files.

(continued)

Table A-2. (continued)

Image Gallery	<ul style="list-style-type: none">• Every <code></code> you add to your site needs to have an alt attribute. If the image is informational, set the <code>alt</code> equal to a <i>descriptive</i> alternative for that image.• Avoid using generic strings like photo, image, or icon as alt values, as they don't communicate valuable content to the user. Be as descriptive as possible.
Images	<ul style="list-style-type: none">• Every <code></code> you add to your site needs to have an alt attribute. If the image is informational, set the <code>alt</code> equal to a <i>descriptive</i> alternative for that image.• If the image is decorative or redundant to adjacent text, set <code>alt=""</code>, which conveys to assistive technology users that the image isn't necessary for understanding the page.• Avoid using generic strings like photo, image, or icon as alt values, as they don't communicate valuable content to the user. Be as descriptive as possible.• Make sure any text in images of text is at least 14 points and has good contrast with the background.
Site Logo	<ul style="list-style-type: none">• Every <code></code> you add to your site needs to have an alt attribute. If the image is informational, set the <code>alt</code> equal to a <i>descriptive</i> alternative for that image.• Avoid using generic strings like photo, image, or icon as alt values, as they don't communicate valuable content to the user. Be as descriptive as possible.• Make sure any text in images of text is at least 14 points and has good contrast with the background.• When using images as links, one must consider how the alt text will be read back to users of assistive technology. The following examples showcase appropriate alt text for logos when used as a link, or as a stand-alone image asset.

(continued)

Table A-2. *(continued)*

SVG's	<ul style="list-style-type: none"> • SVGs are scalable vector graphics and can be made used for icons, images, logos, etc. SVG content is scalable and scales without any reduction in visual quality. • The best way to make SVGs accessible to assistive technologies (AT) like screen readers and speech recognition tools is to put it directly into your HTML using the <svg> tag • Avoid using <embed>, <object>, or elements as they are not as supported by browsers as inline SVG • Include a <title> and <description> in your SVG markup • Use <code>aria-labelledby=""</code> and reference the id values of the title and description elements • Give your SVGs a job with the <code>role=""</code> attribute • To “hide” elements from a screen reader in an SVG add <code>role="presentation"</code>
Videos	<ul style="list-style-type: none"> • Build your media with accessibility in mind! It is much easier to work accessibility into the beginning than trying to tack it on later. This is true of all components, but especially for media components. • Make sure your player is accessible and includes control elements to pause, stop, and play your media. • Do not auto-play your media. This can cause confusion as well as annoyance. • Make sure your media has alternative methods to get to the content. For example, add captions to your videos and/or provide a transcript for your users to read. • Make sure your media does not cause seizures! Use the Photosensitive Epilepsy Analysis Tool (PEAT) to check your media before you add it to your web site.

(continued)

Table A-2. *(continued)*

Navigation— Accordion	<ul style="list-style-type: none"> • Buttons are used as the accordions so that they are tab-able by keyboard users and accessible to screen readers. • Each accordion button and related content has a unique id associated with its <code>aria-controls</code>. • Each button has an <code>aria-expanded</code> attribute on it that is toggled between true and false. If <code>aria-expanded='true'</code>, the content associated with it is shown, and if <code>aria-expanded='false'</code> the content is hidden.
Breadcrumbs	<ul style="list-style-type: none"> • Place the breadcrumb in a <code><nav></code> element when you can. • If you do not use a <code><nav></code> element, you need to add <code>role="navigation"</code> to the markup. Note: this role is implied when you use the <code><nav></code> element, so it is a bit redundant to use both at the same time. • The markup includes an <code>aria-label="breadcrumbs"</code> to describe the type of navigation. • Add <code>aria-current="page"</code> to the link that points to the current page. This will tell assistive technology (AT) devices that the focused link is pointing to the current page.
Basic Navigation	<ul style="list-style-type: none"> • All navigation elements should have a <code><nav></code> tag. • If you have to use a more generic element such as a <code><div></code>, add <code>role="navigation"</code> to every navbar to explicitly identify it as a landmark region for users of assistive technologies. • Menus should be labeled according to their individual function. You can use <code>class="visuallyhidden"></code>, <code>aria-label=""</code>, or <code>aria-labelledby=""</code> to easily add context to the different menus on your site.

(continued)

Table A-2. (continued)

Dropdown	<ul style="list-style-type: none"> • All navigation elements should have a <code><nav></code> tag.
Navigation	<ul style="list-style-type: none"> • If you have to use a more generic element such as a <code><div></code>, add <code>role="navigation"</code> to every navbar to explicitly identify it as a landmark region for users of assistive technologies. • Menus should be labeled according to their individual function. You can use <code>class="visuallyhidden"></code>, <code>aria-label=""</code>, or <code>aria-labelledby=""</code> to easily add context to the different menus on your site.
Footer	<ul style="list-style-type: none"> • All navigation elements should have a <code><nav></code> tag.
Navigation	<ul style="list-style-type: none"> • If you have to use a more generic element such as a <code><div></code>, add <code>role="navigation"</code> to every navbar to explicitly identify it as a landmark region for users of assistive technologies. • Menus should be labeled according to their individual function. You can use <code>class="visuallyhidden"></code>, <code>aria-label=""</code>, or <code>aria-labelledby=""</code> to easily add context to the different menus on your site.
Mobile	<ul style="list-style-type: none"> • Use the <code><button></code> element for your mobile navigation icon.
Navigation	<ul style="list-style-type: none"> • If you use an icon that is purely decoration, declare <code>alt=""</code>, as no additional information is needed. If the icon you are using <i>is</i> important to the functionality, then supply additional <code>alt="_descriptive text goes here_"</code> information. • It is helpful to all users when you add descriptive text when displaying a mobile icon to give more context to the button's purpose. • Place mobile open/close buttons within the <code><nav></code> element and use them to toggle state of another child wrapper of the nav. This will ensure that the navigation landmark is still discoverable by screen readers, even if it is in a closed/hidden state.

(continued)

Table A-2. *(continued)*

Pagination	<ul style="list-style-type: none">• Place the pager in a <nav> element when you can.• If you do not use a <nav> element, you need to add <code>role="navigation"</code> to the markup. Note: this role is implied when you use the <nav> element so it is a bit redundant to use both at the same time.• The markup includes an <code>aria-label="pagination"</code> to describe the type of navigation.• Add <code>aria-current="page"</code> to the link that points to the current page. This will tell AT that the focused link is pointing to the current page.• Add <code>aria-disabled="true"</code> to the link when the link is disabled.
Headings	<ul style="list-style-type: none">• Navigating through the <h1> and <h2> give a user an overview of a page and how its content is structured. The <h3> through <h6> elements provide a quick understanding of the details in each section.• Heading tags should be in order. That means an <h1> is followed by an <h2>, an <h2> is followed by a <h2> or <h3>, and so on. It <i>is</i> okay to skip heading levels when going up in order (ex. <h4> to <h1>).• Keep heading tags consistent. Inconsistently implementing headings can create confusion and frustration for users using assistive technologies.• Do not style text to give the visual appearance of headings—use actual heading tags.

(continued)

Table A-2. *(continued)*

Lists	<ul style="list-style-type: none"> • Creating accessible lists is fairly straightforward and easy <i>if</i> you use the correct markup. • Use <code>ol</code> markup to group ordered lists; use <code>ul</code> markup to group unordered lists; and use <code>dl</code> markup to group terms with their definitions. • Simple comma-separated lists may not need list markup, but longer lists or groups of links should have it.
Tables	<ul style="list-style-type: none"> • Tables with one header and simple data are fairly accessible out of the box and may not need additional accessibility updates. Always use the simplest table configuration possible. • When your tables get more complex, use the <code><th></code> element to identify the header cells by adding a <code>scope</code> attribute. For header rows use <code><th scope="row"></code>. For header columns use <code><th scope="col"></code> • Add the optional <code><caption></code> element before the table content to give users more information on the table contents. Think of it as a headline for your table.
Tabs - Draft	<ul style="list-style-type: none"> • Be mindful of using tabs, as they are less discoverable by design. • Once a tab button is focused, other tabs can be selected with the arrow keys. • A tab's contents can be accessed via <code>Tab</code> (if there are focusable elements in the tab well) or <code>PgDn</code> (if there are no focusable elements in the tab well). • Be sure to update the values of the <code>aria-posinset</code> and <code>aria-setsize</code> attributes if you have more than three tabs.

Periodic Table of Semantics by Gerard Cohen

Periodic Table of Semantics by Gerard Cohen <https://gerardkcohen.github.io/periodic-table-of-semantics.html>

Assistive Technologies and Adaptive Strategies

From the IAAP (International Association of Accessibility Professionals) Study Guide—ICT stands for Information and Communication Technology

Color Blindness: Examples of Assistive Technologies and Adaptive Strategies

Domain	Challenges	Solutions
General	Certain color combinations—red and green in particular—can be difficult to distinguish	Materials can be designed in a way that does not depend on color as a way to convey information.

Reading Disabilities and ICT: Examples of Assistive Technologies and Adaptive Strategies

Domain	Challenges	Solutions
ICT	Often perceive words as floating and not in a line	Can use a special font developed for dyslexia which weights the letters down and makes similar figures appear differently May be granted additional time to complete tasks
ICT	Often perceive words differently than others such as seeing p b d q as the same letter	Can use a special font developed for dyslexia which weights the letters down and makes similar figures appear differently May be granted additional time to complete tasks
ICT	Often require additional time to read and process content	Can extend time-outs and return to the same location on the page Can use a screen reader to get content in an auditory method to reinforce what is being seen Can use screen readers which highlight the word or phrase being read to assist with tracking Can use enhanced visible focus indicators to keep track of their position on the page Can use special programs or dictionaries which present words with pictures May be granted additional time to complete tasks

(continued)

APPENDIX A: ACCESSIBLE HEURISTICS

Domain	Challenges	Solutions
ICT	Often have the burden of deciphering content from the way it is presented	May apply a custom style sheet
ICT	May have difficulty solving problems presented through security features such as CAPTCHA	Ability to change the type of problem presented
ICT	May have difficulty processing content through visual means	Can use a screen reader to get content in an auditory method to reinforce what is being seen May be granted additional time to complete tasks
ICT	May have a hard time spelling words correctly	Can use a spelling and grammar checker

Blindness: Examples of Assistive Technologies and Adaptive Strategies

Domain	Challenges	Solutions
ICT	Cannot see digital or electronic interfaces (computers, automated teller machines (ATMs), mobile devices, airport kiosks, televisions, printers, copiers, phones, GPS devices, etc.)	Screen readers can read interfaces and content out loud to users by converting digital text to synthesized speech, but only if they have been designed to be accessible. Self-voicing interfaces and applications can communicate to users without the need for a screen reader, but these are appropriate mostly for broadcasting information, because they usually do not use or interact with the interface or content as screen readers do. Refreshable braille output devices use screen readers to convert digital text to braille. These devices are typically expensive, and only a minority of blind people know how to read Braille.
ICT	Cannot use screen readers on digital content and interfaces not designed with accessibility in mind	Interface designers and content authors can edit the markup to make it compatible with the assistive technologies used by blind people.

(continued)

APPENDIX A: ACCESSIBLE HEURISTICS

Domain	Challenges	Solutions
Architecture and Built Environment	Cannot see when walking	<p>Can help blind people feel their surroundings as they walk.</p> <p>Service animals (e.g., “Seeing Eye” dogs), trained to assist blind people, help them navigate their surroundings.</p> <p>GPS-based walking instructions with an audio interface, either automated or via a remote human navigator.</p> <p>Raised tiles on the ground to indicate the edge of a platform, a pathway along a sidewalk, the beginning of a staircase, etc. Eliminate low-hanging architectural features that a blind person could bump into.</p> <p>Clear pathways without obstructions in hallways, sidewalks.</p>
Architecture and Built Environment	Cannot see signs or other text on buildings or other areas in the built environment	<p>Map and geolocation applications on mobile devices can announce the names and descriptions of buildings and other location-related information.</p> <p>Braille labels and descriptions on entrances, rooms, bathrooms, historical markers, and other points of interest can allow blind people to explore and understand their surroundings, as long as the person knows braille, and as long as the braille labels are easy to find.</p> <p>Tactile models of the exterior of buildings or of floorplans of the interior of buildings help blind people form a mental map of their surroundings.</p>

Domain	Challenges	Solutions
Consumer and Industrial Products	Cannot see or feel the controls on flat interfaces on consumer devices such as microwaves, ovens, dishwashers, etc.	Alternative interfaces with knobs or other tactile controls. Audio interfaces. Remote control through applications on mobile devices.
Consumer and Industrial Products	Cannot read the text on the containers or packaging for consumer items such as medicine, toothpaste, shampoo, sunscreen, hand cream, personal care products, foods, drinks, and candy	Embossed braille (or braille stickers) on packaging and product containers help consumers identify items both in the store and after purchase.

(continued)

APPENDIX A: ACCESSIBLE HEURISTICS

Domain	Challenges	Solutions
Consumer and Industrial Products	Cannot read money to determine its value	<p>Applications on mobile devices can photograph the money and read the value to blind people.</p> <p>Paper bills and coins could be manufactured in different sizes, shapes, or textures to allow blind people to distinguish the value based on touch.</p> <p>Non-cache systems of payment can allow blind people to make financial transactions via computers, mobile devices, or on-site payment hardware with screen readers or self-voicing output.</p>
Consumer and Industrial Products	Cannot read books, magazines, posters, postal mail, or other printed materials	<p>Optical character recognition software can convert scanned images of text into digital text readable by screen readers. The accuracy of the conversion depends on the quality of the original document, as well as font choices, line spacing, and the quality of the conversion software itself.</p> <p>Information can be placed online or in other digital formats to allow blind people to read the materials using their own assistive technologies.</p>

Low Vision: Examples of Assistive Technologies and Adaptive Strategies

Domain	Challenges	Solutions
General	Small text can be hard to read.	<p>Screen magnifiers can enlarge the items on the screen to make them easier to read.</p> <p>Utilities to enhance contrast, change colors, or alter other aspects of visual appearance can improve legibility.</p> <p>Screen readers can supplement screen magnifiers by reading interfaces and content out loud to users through synthesized speech, but only if the digital information has been designed to be accessible. Self-voicing interfaces (on ATMs, kiosks, transportation systems, etc.) and applications can communicate to users without the need for a screen reader, but these are appropriate mostly for broadcasting information, because they usually do not use or interact with the interface or content as screen readers do.</p> <p>Alternative large print versions of small print text can make printed materials easier to read.</p> <p>Alternative digital versions (web, mobile applications, etc.) of printed materials can give users the ability to read the materials using their own assistive technologies.</p>
General	Low-contrast text can be hard to read.	<p>Software or hardware options can enhance the contrast of digital text. Interface designers and content creators can choose color combinations with high-enough contrast to easily read.</p>

Math and ICT: Examples of Assistive Technologies and Adaptive Strategies

Domain	Challenges	Solutions
ICT	Inability to distinguish right from left in graphic images	Can read data in a data table or text description as an alternative to graphic representations of data when an alternative is provided May be granted additional time to complete tasks
ICT	Inability to copy graphs, figures, and diagrams	Can use speech-to-text to verbalize instructions for completing homework and test questions when the questions are designed to accessibility guidelines and text-to-speech assistive technology can access content May be granted additional time to complete tasks
ICT	Inability to perform calculations	Can use an accessibility accommodation link to a reference sheet with common equations when provided Can use an onscreen calculator as an accommodation May be granted additional time to complete tasks

Speech and Language: Examples of Assistive Technologies and Adaptive Strategies³

Domain	Challenges	Solutions
ICT	Often require repeated exposure to content before long-term memory processing and comprehension takes place	Screen readers can read interfaces and content out loud to users by converting digital text to synthesized speech, but only if they have been designed to be accessible. Users can adjust rate of speech; vary voice and pitch for repeated yet varied exposure to content. May be granted additional time to complete tasks
ICT	Cannot use screen readers on digital content and interfaces not designed with accessibility in mind	Interface designers and content authors can edit the markup to make it compatible with the assistive technologies used by those with speech and language disabilities May be granted additional time to complete tasks
ICT	May have difficulty writing understandable text	May use programs with writing templates, organizational tools, word prediction, and spell checkers May use speech-to-text programs May be granted additional time to complete tasks

³IAAP Exam Prep IAAP CPACC BOK 2017_062317
<https://www.accessibilityassociation.org/exampreparation>

WCAG 2 for Designers⁴

Success Criteria	Area of focus
1.3.1 Info and Relationships (A) (2.0)	<ul style="list-style-type: none">• Organize pages using properly nested HTML headings.• Use ARIA landmarks and labels to identify regions of a page.• Reserve tables for tabular data, use table headers appropriately, and use table captions.
1.3.4 Orientation (AA) (2.1)	<ul style="list-style-type: none">• All content and functionality should be available regardless of whether a mobile device is oriented vertically or horizontally, unless the orientation of the device is absolutely essential.
1.4.1 Use of Color (A) (2.0)	<ul style="list-style-type: none">• When the color of words, backgrounds, or other content is used to convey information, also include the information in text.• Links should always be easily identifiable through noncolor means, including both default and hover states. The easiest and most conventional way to signify links is underlining.• Required fields and fields with errors must include some noncolor ways to identify them.
1.4.2 Audio Control (A) (2.0)	<ul style="list-style-type: none">• Do not have audio that plays automatically on the page. When providing audio, also provide an easy way to disable the audio and adjust the volume.

⁴WCAG 2 Checklist for designers <https://usability.yale.edu/web-accessibility/articles/wcag2-checklist/designers>

Success Criteria	Area of focus
1.4.3 Contrast (Minimum) (AA) (2.0)	<ul style="list-style-type: none"> Text (including images of text) have a contrast ratio of at least 4.5:1. For text and images of at least 24px and normal weight or 19px and bold, use a contrast ratio that is at least 3:1.
1.4.5 Images of Text (AA) (2.0)	<ul style="list-style-type: none"> Avoid images of text, except in cases such as logos.
1.4.10 Reflow (AA) (2.1)	<ul style="list-style-type: none"> Provide responsive style sheets such that content can be displayed at 320px wide without horizontal scrolling. (Content which must be displayed in two dimensions, such as maps and data tables, may have horizontal scrolling.)
1.4.11 Non-text Contrast (AA) (2.1)	<ul style="list-style-type: none"> Color contrast for graphics and interactive UI components must be at least 3:1 so that different parts can be distinguished. When providing custom states for elements (e.g., hover, active, focus), color contrast for those states should be at least 3:1.
1.4.13 Content on Hover or Focus (AA) (2.1)	<ul style="list-style-type: none"> For content that appears on hover and focus: the content should be dismissible with the escape key; the content itself can be hovered over; and the content remains available unless it is dismissed, it is no longer relevant, or the user removes hover and focus. To the extent possible, content that appears on hover or focus should not obscure other content, unless it presents a form input error or can be dismissed with the escape key.

(continued)

Success Criteria	Area of focus
2.1.1 Keyboard (A) (2.0)	<ul style="list-style-type: none"> All functionality should be available to a keyboard without requiring specific timing of keystrokes, unless the functionality cannot be provided by a keyboard alone.
2.2.2 Pause, Stop, Hide (A) (2.0)	<ul style="list-style-type: none"> Items on the page should not automatically move, blink, scroll, or update, including carousels. If content does automatically move, blink, scroll, or update, provide a way to pause, stop, or hide the moving, blinking, scrolling, or updating.
2.3.1 Three Flashes or Below Threshold (A) (2.0)	<ul style="list-style-type: none"> Do not provide any content that flashes more than three times in any 1-second period.
2.4.3 Focus Order (A) (2.0)	<ul style="list-style-type: none"> Create a logical tab order through links, form controls, and interactive objects. When inserting content into the DOM, insert the content immediately after the triggering element, or use scripting to manage focus in an intuitive way. When triggering dialogs and menus, make sure those elements follow their trigger in the focus order in an intuitive way. When content is dismissed or removed, place focus back on the trigger.
2.4.4 Link Purpose (In Context) (A) (2.0)	<ul style="list-style-type: none"> The purpose of each link can be determined from the link text alone, or from the link text and the containing paragraph, list item, or table cell, or the link text and the title attribute.
2.4.5 Multiple Ways (AA) (2.0)	<ul style="list-style-type: none"> Each web site should include at least two of the following: a list of related pages; table of contents; site map; search; or list of all pages.

Success Criteria	Area of focus
2.4.6 Headings and Labels (AA) (2.0)	<ul style="list-style-type: none"> • Ensure that on each page, headings, landmark labels, and form labels are unique unless the structure provides adequate differentiation between them.
2.4.7 Focus Visible (AA) (2.0)(link is external)	<ul style="list-style-type: none"> • Provide keyboard focus styles that are highly visible, and make sure that a visible element has focus at all times when using a keyboard. Do not rely on browser default focus styles.
2.5.1 Pointer Gestures (A) (2.1)	<ul style="list-style-type: none"> • Do not require multipoint or path-based gestures (e.g., pinching, swiping, dragging) for functionality unless the gesture is essential to the functionality.
2.5.2 Pointer Cancellation (A) (2.1)	<ul style="list-style-type: none"> • Avoid triggering functionality on down-events, such as onmousedown. Use events such as onclick instead. • If a function is triggered on an up-event (e.g., onmouseup), provide a way to abort or undo the function.
2.5.4 Motion Actuation (A) (2.1)	<ul style="list-style-type: none"> • Avoid activating functionality through motion (e.g., shaking a phone). If motion triggers functionality, provide a way to disable the motion trigger, and provide an alternative way to activate the functionality.
3.2.1 On Focus (A) (2.0)	<ul style="list-style-type: none"> • When the focus changes, the page should not cause a change in page content, spawn a new browser window, submit a form, cause further change in focus, or cause any other change that disorients the user.

(continued)

Success Criteria	Area of focus
3.2.2 On Input (A) (2.0)	<ul style="list-style-type: none"> • When a user inputs information or interacts with a control, the page should not cause a change in page content, spawn a new browser window, submit a form, cause further change in focus, or cause any other change that disorients the user. If an input causes such a change, the user must be informed ahead of time.
3.2.3 Consistent Navigation (AA) (2.0)	<ul style="list-style-type: none"> • When components are repeated across web page, they should appear in the same relative order with regard to other repeated components on each web page where they appear. • When a navigation menu is presented on multiple pages, the links should appear in the same order on each page.
3.2.4 Consistent Identification (AA) (2.0)	<ul style="list-style-type: none"> • When components have the same functionality across several web pages, the components are labeled consistently on each page
3.3.1 Error Identification (A) (2.0)	<ul style="list-style-type: none"> • Programmatically indicate required fields using the required or aria-required att • Programmatically indicate required fields using the required or aria-required attributes. Also, visually indicate required fields in the form’s instructions or form labels. Do not indicate required fields for CSS alone. • Make errors easy to discover, identify, and correct.

Success Criteria	Area of focus
3.3.2 Labels or Instructions (A) (2.0)	<ul style="list-style-type: none">• Use semantic, descriptive labels for inputs. Visually position labels in a consistent way that makes associating labels with form controls easy. Do not rely on placeholder text in lieu of an HTML label.• Provide text instructions at the beginning of a form or set of fields that describes the necessary input.
3.3.3 Error Suggestion (AA) (2.0)	<ul style="list-style-type: none">• If an input error is detected and if suggestions for correction are known, provide suggestions for fixing the submission.
3.3.4 Error Prevention (Legal, Financial, Data) (AA) (2.0)	<ul style="list-style-type: none">• Provide easy ways to confirm, correct, or reverse a user action where a mistake would cause a serious real-world consequence (e.g., submitting financial data, entering into a legal agreement, submitting test data, or making a transaction).

International Laws Web Accessibility⁵

Country	Law Name	Law Type
Australia	Disability Discrimination Act 1992	Non-discrimination
Australia	Procurement Standard Guidance	Procurement Recommendation
Canada	Human Rights Act	Non-discrimination
Canada	Policy on Communications and Federal Identity	Mandatory policy
China	Law on the Protection of Persons with Disabilities 1990, as amended	Accessibility law
China	Voluntary Web Accessibility Standard	Recommendation
Denmark	Agreement on the use of open standards for software in the public sector	Mandatory policy
European Union	Web and Mobile Accessibility Directive	Accessibility law
European Union	European Accessibility Act (proposed)	Proposed law
European Union	The European Union's Directive 2016/2012	Accessibility law
Finland	Act on Electronic Services and Communication in the Public Sector	Accessibility law
France	Law N° 2005-102 Article 47	Accessibility law

⁵International Web Accessibility and Policies <https://dynamapper.com/blog/27-accessibility-testing/532-international-web-accessibility-laws-and-policies>

Country	Law Name	Law Type
France	Order of 29 April 2015 on the general accessibility framework for public administrations	Accessibility law
France	Law N° 2016-1321 Article 106	Digital Governance law
Germany	Act on Equal Opportunities for Disabled Persons of 2002	Non-discrimination law
Germany	Federal Ordinance on Barrier-Free Information Technology	Accessibility policy
Hong Kong	Guidelines on Dissemination of Information through Government Websites	Mandatory policy
India	Rights of Persons with Disabilities Act, 2016 (RPD)	Non-discrimination law
India	Guidelines for Indian Government Websites	Mandatory policy
Ireland	The Disability Act, 2005	Accessibility law
Ireland	Equal Status Acts 2000 to 2004	Non-discrimination law
Ireland	Employment Equality Acts 1998 and 2004	Non-discrimination law
Israel	Equal Rights of Persons with Disabilities Act, as amended	Non-discrimination law
Italy	Law 9 January 2004, n. 4 “Provisions to support the access of disabled people to IT tools” (Stanca Law)	Accessibility law

(continued)

APPENDIX A: ACCESSIBLE HEURISTICS

Country	Law Name	Law Type
Japan	Basic Act on the Formation of an Advanced Information and Telecommunications Network Society	Accessibility law
Netherlands	Procurement Law 2012	Procurement law
Netherlands	Policy in the Netherlands	Mandatory policy
New Zealand	Human Rights Act 1993, including amendments	Non-discrimination law
New Zealand	Online Practice Guidelines	Mandatory policy
Norway	Regulations on universal design of ICT	Non-discrimination law
Republic of Korea	Act on Welfare of Persons with Disabilities	Non-discrimination law
Sweden	Discrimination Act (2008:567)	Non-discrimination law
Switzerland	Federal Law on the Elimination of Inequalities for Persons with Disabilities, as amended	Non-discrimination law
Taiwan	Web Accessibility Guidelines 2.0	Mandatory policy
United Kingdom	Equality Act 2010	Non-discrimination law
United States	Section 508 of the US Rehabilitation Act of 1973, as amended	Procurement law, Accessibility law
United States	Americans with Disabilities Act of 1990 (ADA), as amended	Non-discrimination law
United States	Section 504 of the US Rehabilitation Act of 1973, as amended	Non-discrimination law
United States	Section 255 of the Telecommunications Act of 1996	Non-discrimination law

Additional Reading

Designing with Accessibility in Mind

- *A Web for Everyone: Designing Accessible User Experiences* by Sarah Horton
- *Accessibility for Everyone* by [Laura Kalbag](#)
- *Accessibility Handbook* by Katie Cunningham
- Accessible UX Principles by Whitney Quesenbery
<https://rosenfeldmedia.com/a-web-for-everyone/accessible-ux-principles-and-guidelines/>
- *Just Ask: Integrating Accessibility Throughout Design* by [Shawn, Lawton Henry](#)
- *Mismatch: How Inclusion Shapes Design* by Kat Holmes
- *Design Meets Disability* by [Graham Pullin](#)
- *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability* by Steve Krug
- Adobe Accessibility <https://www.adobe.com/accessibility/resources.html>
- Amazon Accessibility for Fire <https://www.amazon.com/gp/feature.html/?&docId=1000632481>
- Accessibility with Microsoft products <https://www.amazon.com/gp/feature.html/?&docId=1000632481>

Development

- ARIA best practices <https://www.w3.org/wiki/PF/ARIA/BestPractices>
- Inclusive Design Patterns - Coding Accessibility Into Web Design by [Heydon Pickering](#)
- *Apps for All: Coding Accessible Web Applications*, by Heydon Pickering
- *Responsive Design Workflow (Paperback)* by Stephan Hay
- *Practical Approaches For Designing Accessible Websites* by Smashing Magazine
- *Engineering Software for Accessibility (Developer Reference)* 1st Edition by Microsoft Corporation
- Github Accessibility <https://github.com/collections/web-accessibility>
- Mozilla Accessibility for Developers <https://developer.mozilla.org/en-US/docs/Web/Accessibility>
- Google Accessibility for Developers <https://developers.google.com/web/fundamentals/accessibility/>

Web Content Accessibility Guidelines

- Web Content Accessibility Guidelines 2.0
<https://www.w3.org/TR/WCAG20/>
- Web Content Accessibility Guidelines 2.1
<https://www.w3.org/TR/WCAG21/>
- WebAIM <https://webaim.org>

International Resources for Accessibility

- International Association of Accessibility Professionals
<https://www.accessibilityassociation.org>
- United Nations Article 9 Accessibility <https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities/article-9-accessibility.html>

Legal

- *eQuality: The Struggle for Web Accessibility by Persons with Cognitive Disabilities (Cambridge Disability Law and Policy Series)* by Peter Blanck
- *Making Computers Accessible: Disability Rights and Digital Technology* by Elizabeth R. Petrick
- *Structured Negotiation: A Winning Alternative to Lawsuits* by Lainey Feingold

Design Principles

- *Building Design Systems: Unify User Experiences through a Shared Design Language* by Sarrah Vesselov and Taurie Davis
- Material Design - Accessibility <https://material.io/design/usability/accessibility.html>
- Human Interface Guidelines Accessibility - <https://developer.apple.com/design/human-interface-guidelines/accessibility/overview/introduction/>

User Research

- *Just enough research* by Erika Hall
- *Observing the User Experience: A Practitioner's Guide to User Research* – by Goodman Ph.D. School of Information University of California Berkeley, Elizabeth (Author), Mike Kuniavsky (Author), Andrea Moed (Author)
- *Interviewing Users: How to Uncover Compelling Insights* – by Steve Portigal
- *Validating Product Ideas Through Lean User Research* By Tomer Sharon
- *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, Second Edition* 2nd Edition by Jeffrey Rubin (Author), Dana Chisnell (Contributor), Jared Spool (Contributor)

- *Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions* – by Bruce Hanington (Author), Bella Martin (Author)

Planning and Implementation

- Planning for Accessibility <https://alistapart.com/article/planning-for-accessibility/>
- Accessibility Planning and Resource Guide for Cultural Administrators <https://www.arts.gov/accessibility/accessibility-resources/publications-checklists/accessibility-planning-and-resource>
- Accessibility in Practice <https://developer.paciellogroup.com/blog/2014/03/accessibility-practice-process-driven-approach-accessibility/>
- Accessibility Prioritization <https://www.deque.com/blog/accessibility-prioritization-laying-foundation-strategic-plan/>

Usability Testing

- How to Conduct Usability Studies for Accessibility <https://www.nngroup.com/reports/how-to-conduct-usability-studies-accessibility/>
- Tips for Conducting Usability Studies with Participants with Disabilities <https://www.smashingmagazine.com/2018/03/tips-conducting-usability-studies-participants-disabilities/>

APPENDIX A: ACCESSIBLE HEURISTICS

- Usability and Accessibility: Looking at User Experience through Two Lenses <https://www.usability.gov/get-involved/blog/2013/01/accessibility-and-usability.html>
- Think like an Accessible UX researcher part 3: five common mistakes in usability testing and how to avoid them <https://developer.paciellogroup.com/blog/2019/04/think-like-an-accessible-ux-researcher-part-3/>
- Accessibility Testing Tutorial: What is, Tools & Examples <https://www.guru99.com/accessibility-testing.html>
- Accessibility: Usability for All <https://www.interaction-design.org/literature/article/accessibility-usability-for-all>

Media

- 3Play Media <https://www.nad.org/resources/technology/>
- Federal Social Media Accessibility Toolkit <https://digital.gov/resources/federal-social-media-accessibility-toolkit-hackpad/>
- BBC Future Media Standards and Guidelines <https://www.bbc.co.uk/guidelines/futuremedia/accessibility/>

Web Resources

- a11y Project <https://a11yproject.com>
- a11y Style Guide <https://a11y-style-guide.com/style-guide/>
- Knowability <https://knowbility.org>
- Digital a11y <https://www.digitala11y.com/web-accessibility-resources/>
- 100daysofa11y <https://100daysofa11y.com/category/resources/>

Mixed Realities

- Augmented reality and accessibility <https://www.interaction-design.org/literature/article/accessibility-usability-for-all>
- Able Gamers <https://ablegamers.org/thoughts-on-accessibility-and-vr/>
- Augmented reality and Accessibility W3C https://www.w3.org/WAI/APA/task-forces/research-questions/wiki/Augmented_Reality_and_Accessibility
- Accessibility with VR & AR <https://knowbility.org/programs/accessu/2019/accessibility-with-vr-ar/>
- Virtual reality and Accessibility References W#C https://www.w3.org/WAI/APA/task-forces/research-questions/wiki/Virtual_Reality_and_Accessibility_References

Voice User Interface

- *Voice User Interface Design* by James P. Giangola, [Jennifer Balogh](#)
- *Designing Voice User Interfaces: Principles of Conversational Experiences* 1st Edition by [Cathy Pearl](#)
- Tips for Designing Accessibility in Voice User Interfaces <https://uxdesign.cc/tips-for-accessibility-in-conversational-interfaces-8e11c58b31f6>
- Building Chatbots with Python: Using Natural Language Processing and Machine Learning Paperback – by [Sumit Raj](#)
- So you want to be a voice designer <https://medium.com/@muppetaphrodite/so-you-want-to-be-a-voice-designer-22cdb3cc5b92>
- Designing for Voice <https://uxplanet.org/designing-for-voice-c6259f07c49c>
- Designing for the future with Voice Prototypes <https://uxplanet.org/designing-for-voice-c6259f07c49c>

Content and Writing for Accessibility

- *Designing Connected Content: Plan and Model Digital Products for Today and Tomorrow (Voices That Matter)* 1st Edition by Carrie Hane, [Mike Atherton](#)
- *Content Strategy for the Web*, 2nd Edition 2nd Edition by Kristina Halvorson, Melissa Rach

- *Ensuring Digital Accessibility through Process and Policy* by Jonathan Lazar, Daniel F. Goldstein, et al.
- Tips for getting started: Writing for Web Accessibility
- 7 Guidelines for writing accessible microcopy <https://blog.prototypr.io/7-guidelines-for-writing-accessible-microcopy-8d52575f5d8e>
- Writing for Accessibility <https://styleguide.mailchimp.com/writing-for-accessibility/>

Accessibility Consultants (The Author does not endorse or recommend products or services. Please do your own research into companies)

- Deque <https://www.deque.com>
- Equal Entry <https://equalentry.com>
- The Carol Center for the Blind <https://carroll.org>
- NFB International Braille and Technology Center for the Blind <https://nfb.org/programs-services/center-excellence-nonvisual-access/international-braille-and-technology-center>
- Knowability <https://knowability.org>
- Interactive Accessibility <https://interactiveaccessibility.com>
- National Association of the Deaf <https://www.nad.org/resources/technology/>

APPENDIX A: ACCESSIBLE HEURISTICS

- Level Access <https://www.levelaccess.com>
- Paciello Group <https://www.paciellogroup.com>
- Tenon <https://tenon.io>
- Web Aim <https://webaim.org>

Disability Organizations

- CSUN (California State University Northridge) <https://www.csun.edu/cod/>
- Disabled List <https://www.disabledlist.org/>
- NFB International Braille and Technology Center for the Blind <https://nfb.org/programs-services/center-excellence-nonvisual-access/international-braille-and-technology-center>
- National Association of the Deaf <https://www.nad.org/resources/technology/>
- Lighthouse <http://lighthouse-sf.org>

Education and Technology

- Association of University Centers on Disabilities: A network of interdisciplinary centers advancing policy and practice for and with individuals with developmental and other disabilities, their families, and communities.
- **Center for Applied Special Technology:** Advocates for educational products, classroom practices, and products that are inclusive to all.

- **National Association of Special Education Teachers:** National membership organization dedicated to rendering all possible support and assistance to those preparing for or teaching in the field of special education.
- **Assistive Technology Industry Association:** A not-for-profit membership organization of manufacturers, sellers, and providers of technology-based assistive devices and services.
- **Microsoft Partners in Assistive Technology:** Third-party manufacturers of screen readers, magnifiers, and specialty accessibility hardware that are compatible with Microsoft technology.
- **The Policy Surveillance Program:** The goal of this program is to increase the use of policy surveillance and legal mapping as tools for improving the nation's health. More and more, researchers, policy-makers, public health practitioners, and the media are recognizing the need for access to reliable information about laws and policies that influence the public's health.
- **Trace Research and Development Center:** Conducts research and advocates for accessibility to new and emerging technologies.

Employment and Business

- **Broad Futures:** Works to empower young adults with learning disabilities through employment.
- **Incight:** A nonprofit that aims to eliminate stigma associated with disability and expand inclusion by helping students and jobseekers in particular.
- **Partnership on Employment and Accessible Technology:** Multifaceted initiative to foster collaboration and action around accessible technology in the workplace. Guided by a consortium of policy and technology leaders, PEAT works to help employers, IT companies, and others to understand why it pays to build and buy accessible technology, and how to do so.
- **Our Ability, Building the Business Case for Employment of People with Disabilities:** Our Ability consults with businesses to leverage the successful employment of individuals with disabilities.
- **US Business Leadership Network:** A national nonprofit that unites business around disability inclusion in the workplace, supply chain, and marketplace. More than 130 corporate partners look to USBLN for guidance on disability inclusion, including recruitment and outreach, supplier diversity, and accessibility.

Family and Social Services

- **American Association of Caregiving Youth:** National resource for children who sacrifice their education, health, well-being, and childhood to provide care for family members who are ill, injured, elderly, or disabled.

- **Community Options:** Supports people with disabilities through developing residential, employment, and other support services.
- **Easter Seals:** Founded in 1919, offers therapy, early intervention services, camps and employment placement and helps children and adults with disabilities, caregivers, veterans, and seniors be at their best as they live, learn, work, and play.
- **Through the Looking Glass:** Pioneers research, training, and services for families in which a child, parent, or grandparent has a disability or medical issue. This nonprofit organization emerged from the independent living movement.

Harassment and Hate Crimes

- **All Walks Of Life:** Focuses on prevention of violence to people with disabilities with information and a moderated email list.
- **US Equal Employment Opportunity Commission:** Federal agency that investigates cases of disability discrimination.⁶

⁶National Center on Disability Journalism <https://ncdj.org/resources/organizations/>

Index

A, B

A11Y style guide

- audio, 225
- basic card, 221
- basic navigation, 228
- breadcrumbs, 228
- buttons, 222
- colors, 223
- dropdown navigation, 229
- footer navigation, 229
- forms—checkboxes, 222
- headings, 230
- images, 226
- link focus, 223
- lists, 231
- mobile navigation, 229
- navigation—accordion, 228
- pagination, 230
- read more links, 224
- site logo, 226
- skip links, 224
- SVG's, 227
- tables, 231
- tabs-draft, 231
- typography, 225
- videos, 227

ABC Science case study

- accessibility window, Twitter, 56
- CAPTCHA security tests, 54
- dark patterns, 55
- Disability Discrimination Act, 58
- WCAG 2.1, 57
- webform instructions, 56

A/B testing, 106

- Access and ability, Cooper Hewitt museum, 193, 195

Accessibility, 2

- consultants, 259
- errors, 58

Accessibility for all

- content managers, 113
- user research
 - bias buddies, 118, 119
 - big questions, 115, 116
 - collaboration, 118
 - evidence-based decisions, 117
 - goal, 114, 115
 - messy imperfection, 117
 - prototyping, 114
 - time and money, 116

INDEX

- Accessible content
 - KIS, 95
 - Skip to Content, 95
 - time, use and read, 97, 98
 - writing, 96, 97
 - Accessible design, rules of
 - thumb, 219, 220
 - Accessible Rich Internet
 - Applications (ARIA),
22, 41, 109
 - Adaptive technology, 123
 - Airbnb design, 86
 - American Federation for the
Blind (AFB), 32
 - Americans with Disabilities
Act (ADA), 62, 79–81, 148
 - Annoyance, 43
 - accessible, better, 59, 60
 - conference room,
example, 43–45
 - dark patterns, 58
 - Annoyances and solutions
 - buttons/links, 50, 51
 - page, too long to load, 46
 - poor content layout, 46–48
 - pop up/modals, 48, 49
 - too much text, 49
 - Another Lens framework,
86, 87
 - Artificial intelligence (AI), 210
 - capability, 211
 - download page, 210
 - Assistive technologies, 32, 132
 - Assistive technologies and adaptive
strategies
 - color blindness, 232
 - architecture and built
environment, 236
 - consumer and industrial
products, 237, 238
 - ICT, 235
 - vision, 239
 - math and ICT, 240
 - reading disabilities and
ICT, 233, 234
 - speech and language, 241
- Assistive technology (AT)
- vs.* adaptive technology, 123–125
 - definition, 122
 - future, 143
 - google search, 122
 - keyboard
 - AccessKey, 134
 - searching text, 134
 - web pages, 133
 - zoom, fonts and colors, 135
 - mobile
 - accessibility features,
136, 137
 - voice assistants, 137–143
 - readers (*see* Screen reader)
 - web browsers, 132, 133
- Augmented Reality (AR), 199, 206
- Autoplay
- ABC Science (*see* ABC Science
case study)

dark patterns, 53
 Netflix, 51, 52
 user control, 52

C

Captioning options
 accessible, 69
 automatic
 closed caption, 71
 Google, 70
 Success Criterion 1.2.2,
 71, 72
 YouTube, 69, 70
 closed, 68
 engaging, 69
 searchable, 69
 3PlayMedia, 68
 voice recognition, 73, 74
 Yahoo sports, 67, 68
 Cascading Style Sheets (CSS),
 22, 30, 31, 37, 40
 Color blindness, 232
 blue–yellow, 15
 Color Rush, 12, 18
 monochromacy, 15, 16
 project, 11, 12
 red–green, 14
 types, 13
 Concept testing, 105
 Content and writing, 258, 259
 Cruise control, 191, 192
 Curb cut, 189–191
 Customer feedback, 105

D

Deaf people, gaming
 experiences, 204
 Designing, accessibility, 251
 motor disabilities, 2
 user experiences, 1
 visual disabilities, 2
 Design thinking, 8, 9
 Digital experiences, 99
 Digital Services Georgia, case
 study, 107
 Digital Services Georgia platform
 color contrast, 108
 font legibility, 108
 keyboard-only navigation, 109
 screen reader, 109
 semantic markup, 108
 Disability
 medical model, 4
 organizations, 260
 social model, 4, 5

E, F

Education and skill
 teaching, 202, 203
 Education and technology, 260, 261
 employment and business, 262
 family and social
 services, 262, 263
 harassment and hate crimes, 263
 Email surveys, 107
 Empathy maps, 9–11

INDEX

Empathy, values, vision, context,
and communication
(EVVCC) framework, 84
areas, 85
design strategy, 86
eQuality, 253
Evaluative research, 107
Exclusion, design
disability, 102
user research methods, 103
Eyetracking, 104

G

Generative research, 107
Gesture-based designs, 198
Google slides
captions, 214–216
landing page, 214
microphone settings, 214
Graphics processing unit
(GPU), 201
Guiding principles
balance, bias, 87
confirmation bias, 88
lenses, identification, 88
growth mindset, 89, 90
opposite, consideration, 87, 89

H

Haptics, 198
Head-mounted display
(HMD), 199, 200

Healthcare, 200
HumanWare Brailliant braille
display, 197, 198
Hyper Text Markup Language
(HTML), 30
accessibility, example
headings, 36
text description,
image, 37
title page, 35, 36
WCAG 2.1 color, 37, 38
elements, 34, 35
headings, 35
resources, 33

I

Inclusive design
defined, 7
design tips, 91
do's, don'ts, 92–94
Microsoft, 6
research, 119
Inclusive experiences
blue–yellow color
blindness, 15
color blindness, 13
Color Rush, 18
cone monochromacy, 15
design, 11, 12
NFL game, 17
rod monochromacy, 16
Color Rush, 12
red–green color blindness, 14

Inclusivity/accessibility
 create culture
 avoid discrimination, legal
 complaints, 148
 benefits, 147
 build positive PR, 149
 higher-quality code, 151
 increase usability, 150
 SEO, 150
 Tidbits, 151, 152
 wider audience,
 reach, 148
 design systems, 153, 154
 implementation, 154
 organization structure, 155

Information and communication
 technology (ICT),
 232, 233, 240

Instagram, 66, 67

International Laws Web
 Accessibility, 248–250

International resources, 253

Internet, 25, 79, 107, 188

J

JavaScript
 ARIA, 41
 definition, 40

K

Keep it simple (KIS), 95

L

Legal, design principles, 254
 Live captions, 212

M, N, O

Math and ICT, 240
 Media, 256
 Mixed reality (MR), 199, 257
 Motor disabilities, 2, 74, 93, 148
 Motor skills, 203
 Movers and Shakers, 208, 209
 Muscle recovery, 203

P, Q

Perceivable, operable,
 understandable, and robust
 (POUR), 61, 64, 65, 82
 captioning (*see* Captioning
 options)
 operable
 gestures, 75, 76
 keyboard accessibility,
 74, 75
 task time, 77
 perceivable
 alternative text,
 Instagram, 66
 Alt text, 67
 robust, 78, 79
 understandable, error, 77, 78

INDEX

Photosensitive Epilepsy Analysis
tool (PEAT), 227, 262
Planning and implementation, 255
Professional sports, 200

R

Reading disabilities, 233, 234
ReCAPTCHA, 57
Recruiting people,
disabilities, 101–102
Rehabilitation Act, 62, 250
Research plan, 100, 101
Route planning, 203

S

Screen magnification
software, 126
Screen reader
head pointers, 129
input devices, 129
JAWS software, 126
motion tracking/eye
tracking, 130, 131
screen magnification
software, 126, 127
speech input software, 129
switch controller, 132
text readers, 128
user flow, 125
Search engine optimization
(SEO), 150
Skydiving/skateboarding, 203

Speech and language, 241
Standards and legislation
Asia Pacific, 63
Canada, 62
Europe, 63
United States, 62
US digital accessibility, 62

T

Technology designed, disabled
people
cruise control, 191, 192
curb cuts, 189–191
screen reader, 192, 193
Telephone, 187, 188
Text messaging, 189

U

Universal Design for Learning
(UDL), 7
Usability testing, 255, 256
benefits, 165
case study
BBC iPlayer, 168–171
challenge, 172
guidelines
annotated UX, 184, 185
copy, 182
IPlayer, 181
standards, 180, 181
mobile application test
plan, 166, 167

- sample test plan, 166
- UX principles
 - add value, 177–179
 - familiarity design, 176
 - user choices, 174, 175
 - user control, 176
- User-centered design (UCD), 8
- User interface, 151, 258
- User research methods,
 - 254, 255
 - card sorting, 106
 - desirability studies, 105
 - intercept surveys, 106
 - interviews, 104
 - participatory design, 104
 - remote panel studies, 105
 - usability lab studies, 104

V

- Virtual reality (VR), 199
 - computers, 201
 - headsets, 200
 - low vision, 207
- Virtual travel, 200
- Vision Aid, BrainPort, 196, 197
- Visual appearance
 - Classic 2 theme, 110
 - Friendly theme, 111
 - Portal theme, 112
- Visual disabilities, 2, 148
- Voice recognition, 73, 125
- Voice User Interfaces (VUI), 137

W

- W5H approach, 5, 6, 20
- WAI Policy page, 64
- WCAG 2
 - audio control (A) (2.0), 242
 - consistent navigation (AA) (2.0), 246
 - error identification (A) (2.0), 246
 - error prevention (AA) (2.0), 247
 - flashes or below threshold (A) (2.0), 244
 - focus order (A) (2.0), 244
 - heading, labels (AA) (2.0), 245
 - info, relationships (A) (2.0), 242
 - link purpose (A) (2.0), 244
 - motion actuation (A) (2.1), 245
 - non-text contrast (AA) (2.1), 243
 - pointer gestures (A) (2.1), 245
 - reflow (AA) (2.1), 243
 - use of color (A) (2.0), 242
- Web accessibility, ADA
 - accessibility lawsuits, 79
 - Beyoncé’s official web site, 81, 82
 - Harvard and MIT, 80
 - target, 80
 - Winn Dixie, 81
- Web Accessibility Initiative (WAI), 64, 65

INDEX

WebAIM, 65, 108

Web Content, 253

categories, 26, 27

code do, 30

CSS, 31

HTML, 30

information

architecture, 27–30

pixels code, 31

Web content accessibility guidelines
(WCAG), 45, 61, 148

Web resources, 257

World Health Organization, 102

X, Y, Z

XR Access, 204, 205

XR prototypes, 206