Design Considerations, by Disability Type

1. Blindness

Input: Keyboard only, in most cases
Output: Audio (via a screen reader)

Design Consideration	Why?
All content must be presented in text or via a text equivalent (e.g. alt text for images or other non-text objects).	Screen readers cannot read non-text content (e.g. images) directly, but they can read alt text that you provide.
All functionality must be available using only the keyboard (Note: be sure to test with the screen reader turned on, because there are subtle differences in keyboard behaviors when the screen reader is on)	Even though most blind users can physically use a mouse or trackpad, it doesn't do them much good because they can't see where the mouse pointer is. It is more effective for them to navigate by the keyboard
The content must use markup with **good structure and semantics** (headings, landmarks, tables, lists, etc.)	Screen reader users often often pull up lists of headings, landmarks, and other semantic elements to help them understand what is on the page. They can also navigate by these elements (e.g. jump directly to the main content landmark, or to a specific heading)
All **custom controls** (e.g. expand/collapse buttons, media player volume control, dialogs, etc.) must have the correct **name/label**, **role** (either with HTML or with ARIA), and **value**, and must change value when appropriate (e.g. aria-expanded="false" changes to aria-expanded="true" after activating the button)	Unlike native HTML elements, custom controls have no semantic parts natively, so screen readers can't tell users what the widget is, and can't update users on the properties of the widget unless you supply that information via ARIA names, roles, states, and properties.
Users must receive immediate feedback after all actions, via their screen reader. Silence after activating a feature is always bad!	Examples of feedback: Expanded/collapsed region, value changed on a control (e.g. on a slider), successful/unsuccessful form submission, notification that a new "page" has loaded in single-page applications, etc.
Videos require audio descriptions (additional narration of visual content) if the video's original audio track (dialog, sounds, narration) does not explain everything that a blind person would need to know to understand the video.	Blind users can hear the dialog, narration and other sounds in videos, but they can't see the visual parts of a video, so if the visual parts convey important information, those parts will need to be described out loud for blind users to understand them.
On mobile devices: • **All features require a click action**. • **Custom swipe actions on web pages will not work with the screen reader turned on**	When a blind screen reader user is on a mobile device, swipe actions are used by the screen reading software. All features (controls, widgets) on a mobile web page require a click action to work at all.

2. Low Vision

Input: Mouse, keyboard (no special methods)
Output: Screen magnifier, and in some cases audio (screen reader) also

Users who need to zoom in:
The pinch-to-zoom feature must not be disabled
(avoid <meta name="viewport" content="user-scalable=no">)
Use real text as much as possible (not text inside of images), because magnified images aren't as clear as magnified real text

Users with low contrast vision:
All text must pass contrast guidelines against the background (verify using Deque's aXe accessibility browser extension or similar)
Links, buttons, and controls must have a visible :focus state, and should have a visible :hover state also that helps users to know where the keyboard/mouse focus is. The default browser :focus state is acceptable per the WCAG guidelines, but users with low vision benefit greatly from enhanced CSS :focus and :hover states.

The user interface should provide a clear visual distinction between content (e.g. text) and controls (e.g. buttons, links, etc.) so users know which items are actionable.

3. Colorblind

Input: Mouse, keyboard (no special methods)
Output: Screen (no special methods)

All information must be understandable without needing to distinguish between colors.

Reds and greens are especially problematic when used as the only way to convey information.

4. Deafblind

Input: Keyboard only, or a braille input device in some cases
Output: Refreshable braille output device (via a screen reader)

All of the considerations for blindness apply.

All of the considerations for deafness apply.

In addition, a transcript must be provided for audio and video content.

5. Deaf & Hard of Hearing

Input: Mouse, keyboard (no special methods)
Output: Screen (no special methods)

All videos must have captions.

All audio-only content must have transcripts.

Sign language interpretation of videos can be very helpful.

6. Motor/Mobility

Input: Varies widely: Keyboard (or similar), mouse (or similar), voice, etc.
Output: Screen (no special methods)

Sighted keyboard users (or those who use devices that emulate keyboards):
All functionality must be available using only the keyboard.

Links, buttons, and controls must have a :focus state that is visible to sighted users, and should have a visible :hover state also that helps users to know where the keyboard/mouse focus is.

Enhance visual focus indicator if possible.
Users with slow movements:

With session time-outs, warn users before the time expires (e.g. an accessible dialog or alert), and give them the option to extend the session. Ensure the warning itself allows for slow responses. A recommended minimum response time is 2 minutes.

Users with movements that are difficult to control (e.g. tremors, spasms):
Provide large click targets (links, buttons, controls).

7. Seizures

Input: Mouse, keyboard (no special methods)
Output: Screen (no special methods)

Don’t include videos, animations, or transitions with flashing light sequences of 3x per second or more.

8. Cognitive

Input: Mouse, keyboard (no special methods)
Output: Audio (screen reader)

Users with lower comprehension:
Simplify the interface as much as possible.

Simple the content as much as possible.

Keep videos and audio as short as possible.

Limit the number of choices on the screen.

Provide help features.

Design for ease of use.

Test the usability of the interface with actual users, preferably including users with cognitive disabilities.

Users with memory loss:
Retain information across screens, and within a path.

Provide help features.

Users with distractibility:
Reduce or eliminate distractions (be careful with ads, carousels, intrusive audio, intrusive video, etc.).

Users with difficulty reading (dyslexia, etc.):
Supplement text with illustrations, videos, audio, etc.

Avoid the highest level of contrast for text against the background (e.g. black on white), and
instead choose colors that are a slight step down in contrast (e.g. dark gray text against white or
off-white) BUT be sure to stay within the contrast range that people with low vision need.

9. Speech

Input: Mouse, Keyboard (no special methods)
Output: Screen (no special methods)

Don’t depend on voice input (e.g. in mobile apps, custom widgets, games, etc.