Minutes

WORK ON THIS IS NOW MOVED TO THE INPUT GROUP

Meeting Minutes

Use this for decisions, in-process scratchpads, and other rough notes.

As of 21 Jan 2025 the Keyboard Only Access subgroup is asked to shift to the larger issue of Input. Please see Input subgroup Google Doc.

9 July 2024

For pre-breakout portion of call:

Questions Bruce had before subgroup activity breakouts.

Q1: Who can grant edit access? Can everyone at least make suggestions?

AC: The sub-group facilitator can grant access, and that is best to do for people in the sub-group (i.e. members of AG).

Q:2 How much deference to current numbering from <u>consolidated outcomes</u>? Can we reorder outcomes?

AC: Please keep the number of the main outcome that's being worked on, please discuss if you want to move onto a different outcome.

Q3: Consolidated outcomes doc provides outcome names and outcome text. Goal is to be "Plain english sentence of what the outcome should achieve" but is that much different than the outcome text?

AC: Goal does allow for a bit more text / explanation, the outcome should be concise. If we end up with very similar things, we can drop the goal. For example, keyboard focus had:

Outcome: The keyboard focus must be visually indicated [commanding, normative] **Goal:** Make it easier to spot the keyboard focus. [colloquial]

Q4: Is the model example (placeholder text) *only one* <u>outcome from 5.1.1</u>? (i.e., Image Alternatives: Images have equivalent text alternatives)

• If so, will we (keyboard) need *five* method decision trees? (Bruce thinks so.)

• If we have multiple method decision trees, do we need multiple *sets* of methods? (Bruce thinks *maybe* not; we can see how it goes...)

AC: I think the image alternative outcomes will be collapsed into one or two. Each outcome should have 1 decision tree if there is more than one method. We could share methods between outcomes, although it hasn't come up yet.

Notes

Attendance

- Bruce
- Detlev
- Azlan
- LoriO
- Theo
- Chuck
- Marco

For subgroup conversation:

- Check that you can make suggestions.
 - One goal for today is practice with template.
- Is prose for Goal any different than "the full outcome text from the list"?
 - That is, did we already start with "Plain english sentence of what the outcome should achieve"?
- What is "item of scope" and is it the same for all five outcomes?
 - Some things are inputs controls
 - Set of Web pages
 - Level of effort (6.2.1), item of scope is process
 - For Keyboard trap, viewport
 - Might be more granular, active area, pop-up

Area of the screen where the focus is, could have non model dialogItem of scope options:

Application Keyboard Command

We will need multiple decision trees.

Methods might be common to multiple outcomes.

Actions for asynchronous work

- Theo: Will work on a decision tree for non-conflict commands
- Bruce: Will review the 25 user needs from scratchpad and determine what hasn't been assigned. Table of user needs (25 rows) versus our 5 outcomes. [done]
 - o Bruce suggests starting with simpler outcomes, saving Keyboard Only for last.
- Unassigned/Future: Turn existing guidance (WCAG 2) into methods.

- Theo: Taking existing user needs, putting them into a table. 2nd column will outline
 other outcomes beyond the drivers that may better support the user's needs. Related
 outcomes.
- Have not started our methods, methods can be ported from understanding documents or other sources.
- Have not done the exercise of writing out the goals and the what to do's.

Items for reporting back on AG call

- Replicated outcomes (so we have 5)
- "Item of scope":
 - o Inputs
 - Process
 - Viewport or page
- Duplication versus overlap
- Filled in document with user needs from scratchpad to help support next week's work
- Asynchronous work can be adding methods,
- Anticipated issues:
 - O What is user agent responsibility?
 - O What is author's responsibility?
 - O Which are technology / platform dependent ?

Next week: If Theo has a decision tree (for not having conflicting keystrokes), that would be a good place to start and see if we can't expand the decision tree and address "bigger" challenges.

Suggestion from Alastair to maybe only pick one of five. Decision tree exercise should be *after* having worked on methods.

Table: 25 User Needs versus 5 outcomes

		User Need	6.2.1 comp. keyboard effort	6.2.2 consistent interaction	6.2.3 (unique) kb commands	6.2.4 keyboard only	6.2.5 no keyboard trap
1		As a user of assistive technology, I need my keyboard commands to remain consistent.		yes		yes	
2	2	As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.	yes			yes	

	User Need	6.2.1 comp. keyboard effort	6.2.2 consistent interaction	6.2.3 (unique) kb commands	6.2.4 keyboard only	6.2.5 no keyboard trap
3	As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.	yes			yes	
4	As a keyboard user, I should be able to perform all actions using a keyboard exclusively.				yes*	
5	As a keyboard user, I should not get unintentionally stuck.		yes		yes	yes
6	As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.			yes	yes	yes
7	As a keyboard user, timing should not prevent me from completing actions.	yes			yes	yes
8	As a keyboard user, I should not be required to press multiple keys at the same time to complete an action.				yes*	
9	As a keyboard user, I always know where I am in the interface.		yes		yes	
10	As a keyboard user, I am informed of non-standard commands before I need to make use of one.		yes	yes	yes	yes
11	As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.		yes	yes	yes	
12	As a keyboard user, the keyboard			yes	yes	

	User Need	6.2.1 comp. keyboard effort	6.2.2 consistent interaction	6.2.3 (unique) kb commands	6.2.4 keyboard only	6.2.5 no keyboard trap
	commands that I need to press don't interfere with my operating system and/or assistive technology.					
13	As a keyboard user, my choice of keyboard is fully supported.				yes*	
14	As a keyboard user, I should be provided with a method of skipping repetitive navigation links.	yes			yes	
15	As a keyboard user, I would like my user agent to provide keyboard shortcuts to quickly navigate around an interface.				yes*	
16	If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.		yes	yes	yes	
17	@@ As a keyboard user, I am warned about changes in context before they occur.					
18	@@ As a keyboard user, I am warned about changes in focus before they occur.					
19	As a keyboard user, I am not expected to use a certain viewport in order to use a keyboard.				yes*	
20	As a keyboard user, applications should be flexible in allowing me to use the keyboard commands I want to use (e.g. through personalisation).				yes*	
21	As a keyboard user, I shouldn't		yes		yes	yes

	User Need	6.2.1 comp. keyboard effort	6.2.2 consistent interaction	6.2.3 (unique) kb commands	6.2.4 keyboard only	6.2.5 no keyboard trap
	have my expected keyboard commands disabled.					
22	As a keyboard user, I should be able to access all scrollable regions of a view/app so I can read all text within these regions.				yes*	
23	As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.	yes			yes	
24	As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented.		yes	yes	yes	yes
25	@@ As a user, I don't want a keyboard shortcut to be the only way to do something (or, there should not be only one way to do something).					
tally		5	8	6	22	6

16 July 2024

Use this for decisions, in-process scratchpads, and other rough notes.

Notes

Attendance

- Chuck
- Azlan
- LoriO
- Sarah Horton
- Detlev Fischer
- Theo

Actions for asynchronous work

- Discussion about prerequisites
 - If this method/outcome is not present, there is no accessibility (issue will likely prevent affected users from completing task)
 - Rationale for rating severity, can review work from <u>Issue Severity</u>

23 July 2024

Use this for decisions, in-process scratchpads, and other rough notes.

Notes

Attendance

- Bruce
- Azlan
- Marco
- Jen
- Poornima Subramanian

Agenda

- Going forward, Bruce will try to provide an agenda (in this doc) before we talk.
 - Probably will *not* be sending out minutes.
 - But did everyone get Bruce's email sent 7/11 ?
 - o Confirm email address as needed.
- Review agenda, adding as necessary.
- Review doc insertions/changes since last week.
- Review actions for asynchronous work from last week.
- Method versus What to Do.

Actions for asynchronous work

- Bruce to send out minutes.
 - Email Jen her action item.
 - Ask Theo to write Goal and What to Do.
- Bruce to format 6.2.3 method as method in this document.

Items for reporting back to larger AG call

- Made progress on Goals and What to do
- One What to re-formatted as Methods
- We rename 6.2.3 to Custom keyboard command
- Any suggestions for plain language for "User Flows"?

30 July 2024

Use this for decisions, in-process scratchpads, and other rough notes.

Notes

Attendance

- Bruce has to leave AG call at 12:20
- Azlan
- Marco has to leave at 12:10 EST
- Jen sends regrets
- Lori O sends regrets
- Sarah Horton
- Chris Loiselle
- Chuck regets, to facilitation another subgroup

Agenda

- Anyone missing last week's notes? (Bruce sent 7/24).
- Review agenda, adding as necessary.
- Review any substantive asynchronous in this doc after last Tuesday.
 - Bruce did some editorial formatting.
 - Decision Tree for 6.2.3 Custom keyboard commands done. Methods need filling in, but that seems straight forward.
 - Jen drafted Goal and What to do for 6.2.1
 - Bruce drafted Goal and What to do for 6.2.2
- Discuss work plan going forward (pending email from AG chairs)
 - Look for time slot going forward.
- Review actions for asynchronous work from last week.

Actions for asynchronous work

• Bruce will send out minutes (done)

Items for reporting back to larger AG call

- Need more time.
- Any suggestions for plain language for "User Flows"?
- Wednesday 11:00 Boston to noon going forward, August 7th.

Chris notes from call 7/30: **Chris Loiselle - Oracle** to Everyone 11:10 AM listening today, happy to help moving forward!

Chris Loiselle - Oracle to Everyone 11:54 AM

for example Windows OS vs. non windows -

https://learn.microsoft.com/en-us/style-guide/a-z-word-list-term-collections/term-collections/keys-list-term-collections/term-collections/keys-list-term-collections/term-collections/keys-list-term-collections/term-collections/keys-list-term-collections/term-collections/keys-list-term-collections/term-collections/keys-list-term-collections/term-collections/keys-list-term-collections/term-collections/term-collections/keys-list-term-collections/term-collections/keys-list-term-collections/term-collections/keys-list-term-collection

-keyboard-shortcuts#keyboard-actions-and-access is that what you mean?

vs. https://support.apple.com/en-us/102650 and the diff of command key?

vs. allowing shortcut to be remapped?

https://www.w3.org/WAI/WCAG21/Understanding/character-key-shortcuts.html

vs. https://www.w3.org/WAI/ARIA/apg/practices/keyboard-interface/ and "F2" . Is F2 "well known"?

14 August 2024

Use this for decisions, in-process scratchpads, and other rough notes. We won't be using IRC for minutes. Understanding is that groups to proceed for eight weeks.

Notes

Attendance

- Bruce
- Marco
- Jen
- Chris

Agenda

- Recap status of work from last few weeks
- Is the time slot (Weds, 11 am U.S. Boston, same as AG) good enough?
 - Bruce had to cancel last week (8/14) and will be can't make it next week (8/21).
- Review agenda, adding as necessary.
- Review actions for asynchronous work from last week.
- Discussed Google Doc as sufficient tool

Actions for asynchronous work

- Bruce to send out minutes
- Bruces to check chairs expectations.
- Bruce to email version
- Bruce to note WCAG2 gap allowing complex keyboard shortcuts. Authors can invent shortcuts that override common keyboard shortcuts. (Theo's ^Z story.)
- Split documentation of keyboard shortcuts as separate method/requirement of 6.2.3.

Items for reporting back to larger AG call (8/20)

• Five outcomes, 6.2.1 Comparable keyboard effort, 6.2.2 Consistent keyboard interaction, 6.2.3 Keyboard commands [are unique], 6.2.4 Keyboard only, 6.2.5 No keyboard trap

- Meeting Wednesday 11:00 Boston
- Subgroup progress to date:
 - Mapping user needs to each outcome (1st meeting, lots of duplicates as to be expected)
 - Goal and What to Do for each outcome (2nd/3rd meeting)
 - Work for next several meeting is methods and decision trees
- 6.2.3 Custom keyboard command
 - Renamed from "Keyboard commands" to "Custom keyboard commands" to be more descriptive
 - Reoriented to only be about content. It is not feasible for keyboard platform commands to be comprehensively documented.
 - Decision tree close to done

20 August 2024

Use this for decisions, in-process scratchpads, and other rough notes..

Notes

Detlev shared example of keyboard trap, where after opening form and start typing. State of UI element prevented further keyboard interaction. (E.g. started typing name, and could not leave name field.) UI elements just taking focus is not enough. Another example is selections (e.g. checking a check box) opens up more fields.

You to Everyone 11:53 AM

https://docs.google.com/document/d/1uE2WCxPmvNopdCbuZQm_-cGyEdxEouRmZ8UUIIyuto U/edit#heading=h.7120wskf56qi

Graham to Everyone 11:54 AM hmmm seems I cant use Mic yeah should be ok

Lori Oakley to Everyone 11:59 AM I he to drop for another meeting, I'll be back shortly

Graham to Everyone 11:59 AM

my gut reaction is no, because otherwise we could argue nearly all things could be fixed with "x specific software"

I have used screen readers for years for testing etc. I still don't know how to activate "pass-through", are end users who rely on AT daily likely to know how to activate this?

Graham to Everyone 12:16 PM

Although this falls into "keyboard trap" probably, some custom elements retain focus within them (I am thinking Code editors), ESC should take you to the NEXT focusable item on the page, I have been trapped a few times where ESC takes you before and so you can't tab past it.

The issue is it isn't a "trap" by standard definitions.

If they are not tabbable then they are not focusable, so odds are they will not work for a screen reader anyway.

This isn't a "decision tree" item, it is more a testing methodology item. The same rules apply at all screen sizes.

Graham to Everyone 12:23 PM

@Bruce Bailey (USAB, he/him) can you open the chat please, nobody is seeing the messages I am writing and I need someone to be my voice today! thanks.

Tabbing is super important for those with dexterity / accuracy impairments such as Cerebral Palsy or Parkinsons, so the Tab test is essential anyway to facilitate those users.

Graham to Everyone 12:31 PM

Please see above message about code editors as this is related.

lease see above message about code editors as this is related.

The problem is I didn't add context as was responding to the conversation at the time so not sure how much of the above makes sense! haha

Graham to Everyone 12:36 PM

Shift-Tab - not necessarily, I have "skip links" throughout a document to skip long tables etc. I would not expect them to show on shift-tab necessarily.

And then that raises the question of should a "skip link" be available at the BOTTOM of a long table (of links for example) to skip it if going in reverse?

Graham 12:42 PM

There needs to be an element of "user expectation". If you cannot get passed something it MUST be a fail unless there is a clearly explained / visible way to get past that item.

We also need to consider people with cognitive impairments, and therefore "expected behavior" is important

OK, I think there needs to be a discussion about usage of chat at a group level. If we ever get a contributor who has a severe speech impairment / is mute their experience will be awful and not accessible.

Attendance

- Azlan
- Bruce
- Detlev
- Lori

- Poornma
- Graham

Items for reporting back to AG:

Q: Is scope strictly web or is it broader?

A: Trying to keep outcomes and decision trees generic.

Methods expected to be technology specific

21 August 2024

Use this for decisions, in-process scratchpads, and other rough notes. Notes added to inline comments.

28 August 2024

Use this for decision, in-progress scratpads, and other rough notes.

Attendance

Bruce Azlan Chris Jen Lori Giacomo Marco

Agenda

- Recap
 - Work since last week?
 - Bruce cleaned editorial, accepted deletions insertions for Decision Trees for <u>6.2.3</u>
 <u>Custom keyboard commands</u> and <u>6.2.4 Keyboard only</u> (thanks Detlev).
- Briefly discuss if heading structure for this doc should be redone?
- Suggestion for 8/28: Outline how 6.2.4 Keyboard only relates to other Outcomes?

From top, back in July:

• Usable

- 6.2.1 Comparable keyboard effort The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.
- 6.2.2 Consistent keyboard interaction Keyboard interface interactions are consistent.
- 6.2.3 Custom keyboard commands Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.

Operable (potential prerequisites)

 6.2.4 Keyboard only - All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

- 6.2.5 No keyboard trap If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.
- We think we are okay with current ordering and grouping as initially provided.
- Keyboard trap, because such blocker (non-interference to use WCAG2 terms) can't be part of Keyboard only.
- No keyboard trap is not something which can tested independently of Keyboard only.
- A separate audit for No keyboard trap would make it too easy to miss keyboard traps.
- A keyboard trap will block further audit
 - Is this a full stop?
 - Can auditor continue for other keyboard accessibility?
 - Keyboard trap might full block further testing. Trap would need to document in workflow, as critical failure. May be more than one keyboard trap!
 - From practical view, what is unit testing? Org using auditor would not want audit to stop!
- Separate decision trees between what author/UX designer/developer does versus audit?
 - o Developer needs same rules as auditor.
 - Decision trees seem to work for a design/develop perspective.

Actions for asynchronous work

- Bruce to add numbers and tiles to duplicative titles.
- Chris will add notes to this doc from Word document circulated last week. (done)
- Need glossary term for standard navigation keys.
- Bruce to clean up doc more.

More notes on glossary term:

- Standard or usual for app like Excel different from Open Office.
- Browsers and operating systems have default controls.
 - What are default for the application?
 - Use of Excel within a web browser? What is usual/common there?
 - O What are all users aware of?
 - First time Excel user probably uses arrow keys rather than enter and tab.
 - Might need two levels of this?
 - Think of how your mother uses the computer!

From chat: 11:17:51 From Chris Loiselle - Oracle to Everyone:

https://www.w3.org/WAI/standards-guidelines/wcag/wcag3-intro/#for-your-review for reference

11:46:48 From Chris Loiselle - Oracle to Everyone:

https://www.w3.org/WAI/ARIA/apg/practices/keyboard-interface/ uses A primary keyboard navigation convention common across all platforms ... might want to improvise off of their wordsmithing.

11:58:13 From Giacomo Petri to Everyone:

Fundamental Keyboard Navigation Conventions 11:58:16 From Giacomo Petri to Everyone: From aria practices

Regrets from Jen for 9/3.

04 September 2024

Use this for decision, in-progress scratpads, and other rough notes.

Attendance

Bruce Chris Lori. Giacomo Regrets from Jen and Theo.

Agenda

- Welcome, Introductions
- Recap since last week?
- AG call yesterday focussed on Assertions, Bruce would like that to be focus for today.
 - o Are assertions per outcome, or for a group of outcomes?

Notes

- Assertions should be specific to outcomes. General assertions will be addressed by larger group.
- Statement of fact that org/author did something.
- An ACR (from VPAT) is an assertion. (general)
- We use third-party testing. (general)
- We assert that we have keyboard shortcuts documented for our developers.
- Example from ALT group was existence of style guide.
 - Style guide would have to be vetted for completeness/accuracy.
 - See presentation materials 9/4 AG call.
- (Generic) Assertions need to apply to plug-ins used by website.
 - Website owners may have limited control over the 3rd party service and also requirement to use 3rd party service (from parent org, legally, only product available).
- (Generic) Customer requirement for accessibility statement, including details of testing.
- We as an organization have tested using <u>ANDI</u> for keyboard accessibility.
 - ANDI would help keyboard testing by https://www.section508.gov/training/web-software/andi-training-videos/focus-element/, however it is what company asserts on how they've used ANDI [what is an assertion for WCAG3]
 - How the org uses the tool
 - We have identified tool to do outcome X/Y/Z
- There are no keyboard traps.

- We validite using ANDI that keyboard focus can be moved to all interactive components, and that the keyboard focus can be moved away from that component.
- (Generic) Org. states what they are doing to come into compliance with an audit.
 - Can be treated as specific (to an SC) because web content audit has specific defects identified.
 - Developer (responding to audit) makes assertions about what they are doing to fix problems.
- See WAI statement generator: https://www.w3.org/WAI/planning/statements/generator/#create-efforts
- Another example: Welcome to the Accessibility Statement Generator built by Nomensa
- We tested the web content page, using XYZ methodes, and achieved ABC results.
 - Bug statements (defects) are tasks that need to get done.
- Example.: All of our non-decorative images have appropriate alt text.
- Enforcement:
 - Who is responsible for testing assertions? (Won't be W3C. Likely to be regulators.)

Noting that focusable is very intertwined with keyboard accessibility.

For reference (during call):

- 6.2.1 Comparable keyboard effort The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.
- 6.2.2 Consistent keyboard interaction Keyboard interface interactions are consistent.
- 6.2.3 Custom keyboard commands Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.
- 6.2.4 Keyboard only All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
- 6.2.5 No keyboard trap If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.

10 September 2024

Notes for AG call.

- Group meeting
 - Weekly, for an hour, Wednesdays 11:00 Eastern.

- Bruce is not available tomorrow, 9/11. Jen's hosting.
- Five related outcomes:
 - 6.2.1 Comparable keyboard effort
 - 6.2.2 Consistent keyboard interaction
 - 6.2.3 Custom keyboard commands
 - 6.2.4 Keyboard only
 - 6.2.5 No keyboard trap
- First exercise, mapping User Needs
 - See table above, 22 of 25 users needs versus 5 outcomes.
 - Missing only 3 from the initial set.
 - Keyboard only hits all 22.
- Comparable keyboard effort
 - Have not started.
- Consistent keyboard interaction
 - o Briefly started, did not get much traction, but might be easier now.
- Custom keyboard commands
 - Thrashed for bit, unpossible to document all OS/platform keyboard commands.
 - Solved by turning the outcome around!
 - Make the web app document any keyboard shortcuts it makes itself.
- Keyboard only
 - Something we have lots of experience from 2.x with success criterion 2.1.1, and decision tree common along.
 - See <u>6.2.4 decision tree above</u>.
- No keyboard trap
 - Decision tree and methods are concrete enough, but...
 - Won't catch some keyboard traps without testing for keyboard accessibility.
 - Example, entering text in form field exposes the keyboard trap when merely tabbing through form did not.
 - Is this outcome a parallel concurrent process with Keyboard only?
 - Part of "All functionality of the content is operable through a keyboard interface" includes not hitting a keyboard trap.
- Works parallels AG calls
 - So last week, 9/4 focus was on assertions.
- Assertions
 - Did not make much progress on outcome-specific assertions.
 - Could have outcome-specific for test methodology used.
 - We have identified tool X for do meeting outcomes X/Y/Z,
 - o Example, ANDI good for keyboard testing.
 - Could provide assertions related to its features.
 - Would be hard to disambiguate from testing for keyboard from testing for focus.
 - ACR from VPAT feels like an outcome-oriented assertion list.
 - From group conversation, probably won't return to assertions because they seem like a whole-of-WCAG3 issue.

- Unresolved question:
 - How to document dependance on AT features? (E.g., forms mode (and analogs) in JAWS, NVDA, Dragon.)
- Next steps?
 - Outline dependencies of 6.2.4 on others?
 - o New clean document?
 - Nice to have notes at the end of this document, but decision trees are hard to find with current formatting.

Notes from AG call

- Alastair noted: If everything can be objectively tested, that's ok!
- We need not return to assertions.

11 September 2024

Use this for decision, in-progress scratpads, and other rough notes. Jen Hosting, regrets from Bruce.

See comments from Chris added to doc 9/11.

17 September 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

- Welcome and recap
 - Comments in document scribed by Chris on 9/11.
 - Detlev mentioned on Tuesday touchpoint on AG call that "Keyboard Only" is an umbrella outcome for the others
- TPAC next week, who's going, are we meeting?
- Look at the preview build for Focus Appearance as model for Keyboard Only
- 6.2.1 Comparable keyboard effort, what do?

Notes

- Attendance: Bruce, Chris, Lori. Giacomo. Regrets from Jen.
- From last week: testing with ANDI, and discussion if testing with AT is subset?
 - Testing with only the keyboard is default for anyone using SR.
 - Where does that fit into decision tree?
 - See Chris' comments on 6.2.4 keyboard only decision tree.
- Recap from AG meeting yesterday, DetLev gave update
 - Explainer Google Doc and Editor's Draft
 - Look for (glossary) terms in particular especially relevant to subgroups.
 - WCAG3 draft for public comment to be published soon after TPAC

- WAI <u>WCAG 3 Introduction</u> is a stable URL for status
- All on the AG call agreed that the concrete examples were very helpful
- Some work overlaps with WCAG2 issues (single pointer in particular)
- TPAC
 - AG will be working on WCAG3 for hours
 - Public schedule is available, also by logging in to your W3C account
 - Free registration for Invited Experts should re-opening Monday 9/23
 - Remote-only attendance to AG calls is also free.
 - Email w3t-tpregister@w3.org (or an AG chair) if you have questions or difficulties.
 - Looking forward to the retrospective presentation
 - Joint meeting between APA, AG, and ARIA
 - Keyboard subgroup will not meet next week
- 6.2.1 Comparable keyboard effort
 - Group agrees to continuing deferring
 - Single pointer (from 2.2) might warrant revisiting when we come back to comparable effort analysis.
- Bruce drafted a goal and what to do for Custom Keyboard Commands
- Keyboard only as main or only Outcome?
 - No objections.
- Discussed Focus Appearance <u>Google Doc</u> and <u>preview build</u> as model for Keyboard Input.
 - Methods for Focus Appearance page includes Decision Tree and list of methods
 - Methods available links from page
 - Note parenthetical status of Enhanced, baseline, and prerequisites
 - List these in order of importance vs. the current mixed list out in numbered formatting.
 - Within the decision tree, links bring users to sub page vs. anchor links for methods.
 - Is that intended?
 - Discussed use of stop vs. no , continue ?
 - It is only baseline requirements at the moment that would have a stop
 - Giacomo shared Google doc version of Focus Appearance
- Google Docs not working great for our collaboration.
 - Word export is working well for post-meeting but is read-only.
- AG calls have been oriented towards repo and comments
 - Works okay for narrow topics
 - Not an appealing option for majority of AG participants for drafting
 - The github repo also hosts the <u>WCAG3 wiki</u> but that is probably not appealing to anyone but Bruce.
- Word online is not familiar to most subgroup members on the call.
- Slack might be an option.
 - Okay for Giacomo, Chris, Lori
 - o Bruce resisting Slack and other social media

Accessibility is of some concern

Action Items

- Bruce will send out email with Word version attached.
- Bruce will send out email with "goal" and "what to do" for Custom Keyboard Commands.
- Bruce draft on a decision tree and/or methods modelled after Focus Appearance, location and format TBD
- Everyone please keep thinking on topic for mechanics of collaboration.
- Next call is -10/2. 10/15.

15 October 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

- Welcome and recap
- The recent <u>SURVEY for participation in upcoming subgroups</u> might help this subgroup pick meeting time..
- Look at the preview build for Focus Appearance as model for Keyboard Only
- Updated Goal for <u>6.2.2 Consistent keyboard interaction</u> to:
 - Keyboard interaction including navigation and operation, is intuitive for users familiar with the platform.

Notes

Regrets from Theo, Gregg, Azlan

Attendance: Bruce, Giacomo, Chris, Jen

Consensus that Focus Appearance decision tree is a good model to emulate.

Some reference links:

- Current <u>WCAG 3 repo on GitHub</u> including issues / discussions / PRs
 - PR112 for <u>PR112</u> includes build preview of <u>How To</u> for Focus Appearance (and Text Alternatives
- Previous (now obsolete) Silver repo on GitHub but that has an
 - o index page to recent (September 11, 2024) editors drafts
- Accessibility Guidelines "Silver" (w3c.github.io)
- w3c/silver: Accessibility Guidelines "Silver" (github.com)
- Chris suggested <u>ARIA-AT</u> community group work on enabling AT Interoperability being relevant to our conversation of how keyboard compatibility includes voice recognition and keyboard emulation via switch input.

Regrets from Jen for next couple of weeks.

Action items:

- Bruce to ask for new Google doc to facilitate drafting.
- Bruce to clean up notes, send out email notes.

22 October 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

 copy/paste 6.24 to a <u>new document</u> – or maybe even better – another tab in this document.

Notes

Regrets from Jen, Chris,

Attendance: Bruce, Giacomo, Azlan, Gregg

Focus Appearance is not as ideal a model since it is more atomic than keyboard operable. Operable first, Useable second.

29 October 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

- Thanks to Azlan and Gregg for organizing document sections into tabs!
- Thank you to Gregg for drafting Composite Outcome and Decision Tree
 - Does group want to go through that on call?
- AG is asking subgroups to wrap up work on 11/5 or 11/12, and report
 - Our last working session next week 11/5 (Bruce not available 11/12 at 10:00 (U.S. Boston))
- Today, we also have 2nd hour of AG call.
- Bruce proposes to use time for mutual drafting.
- Need reporting bullet points for AG call today and on 11/12

Notes

Regrets: Theo, Jen

Attendance: Bruce, Chris, Giacomo, Gregg, Mary Jo, Lori, Sarah, Steve

For AG

- Decent progress on all 5 outcomes, but might not have any single one cleanly finished.
- What is unit of testing for decision tree?
 - As compared to Non Text content, focus appearance
 - Can't cycle through non-text content, or UIE which take focus
 - Complex to determine what is target of analysis
- Suggestion to use ACT vernacular for terms when developing decision trees

From AG

<alastairc> Bruce - you could set the scope as the "view", e.g. within the view, the following does not happen... or something to that effect.

5 November 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

- Our last working session, then 2nd half of WG call.
- Iterate on Composite Outcome.
- Clean up on the other five decision trees.
- Bullet points for presentation 11/12 to AG.

Attendance: Bruce, Giacomo, Jen; Bruce, Chris, Detley, Roldon, Nina

6.2.1 Comparable keyboard effort

Added some notes, need a lot of thought on how to quantify comparable 6.2.2

For AG

- From Composite Outcome
 - o Grouping of user needs sound
- Benefit for passing off is readable, composite outcome simpler
 - Next subgroup can take to next step
- Including focus and other issues makes decision tree too large
 - Current 2.x approach makes testing easier

19 November 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

Bruce gave overview during main AG session

• Subgroup meets for the last half hour of the AG call

Attendance: Gregg, Chris Bruce, Glenda

Group agrees with iterating on Composite Outcome.

Going forward: keep notes on discussion points and conclusions.

Going forward: Thursday Noon, Bruce to send out invite.

21 November 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Agenda

Pickup from <u>Goodwitch testing methodology</u>, goal is to have something for next PWD

No call 11/28.

Attendee: Bruce, Gregg, Glenda, Lori, Jen, Giacomo

Current WCAG 3 working draft has the following outcomes (some of which are covered by the above and some others we think we should address:

Comparable keyboard effort EXPLORATORY

The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.

EDITOR'S NOTE: Needs <u>additional research</u>

Consistent keyboard interaction EXPLORATORY

Keyboard interface interactions are consistent.

Focus in viewportEXPLORATORY

The focus does not move to a position outside the current viewport, unless a mechanism is available to return to the previous focus point.

Keyboard commandsEXPLORATORY

Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.

Keyboard focus location **EXPLORATORY**

The keyboard focus is visually indicated.

Keyboard only EXPLORATORY

All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

No keyboard trapEXPLORATORY

If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.

(WE HAD THE WRONG DOCUMENT. HERE IS THE CORRECT DOCUMENT

2.4.4 Keyboard operation

Users can navigate and operate content using only the keyboard focus.

Comparable keyboard effort **EXPLORATORY**

The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.

EDITOR'S NOTENeeds additional research

Conflicting keyboard commands **EXPLORATORY**

Authored keyboard commands do not conflict with platform commands or they can be remapped.

Consistent keyboard interaction **EXPLORATORY**

Keyboard interface interactions are consistent.

Keyboard mode **EXPLORATORY**

If the keyboard is non-hardware (such as a virtual keyboard), the keyboard input mode is indicated.

Keyboard only **EXPLORATORY**

All functionality must be accessible through the keyboard, except when a task requires input based on the user's specific input action.

No keyboard trap **EXPLORATORY**

If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component, or the component can be dismissed, with focus returning to the previous point.

Non-standard commands **EXPLORATORY**

The user is informed of non-standard authored keyboard commands.

26 November 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Attendance: Regrets: Bruce

Agenda

- No call 11/28. Next call 12/5 noon Boston (should already have invite).
- Pickup from Goodwitch testing methodology, goal is to have something for next PWD
- Bruce's suggestion: Iterate on Combine Outcome, model after PR112 Keyboard Focus and Clear Meaning, namely:
 - Foundational Requirements
 - o Supplemental Requirements
 - Assertions

3 December 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Attendance: Azlan, Bruce, Glenda, Francis, Lori O

Agenda

- Pickup from Goodwitch testing methodology, goal is to have something for next PWD
- Bruce's suggestion: Iterate on Combine Outcome, model after PR112 Keyboard Focus and Clear Meaning, namely:
 - Foundational Requirements
 - Supplemental Requirements
 - Assertions

Discussed individual outcomes versus composite, group agrees with composite.

Discussed GoodWitch versus Gregg Composite, agree to work from composite.

For next week, think on the Foundational / Supplemental approach from WCAG 3.0 Editors Draft.

5 December 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Attendance: Glenda, Lori O, Gregg

Regrets: Chris, Azlan, Jen, Bruce

Agenda

- Goal is to have something for next PWD similar to what is currently in Editor's Draft for
 - 2.1.1 Image alternatives
 - o 2.2.3 Clear meaning
 - o 2.3.1 Keyboard focus appearance
- See accordions and links to How To
- Note, outcomes characterized as Foundational, Supplemental, Assertions
- In ED, see: <u>2.4.4 Keyboard Operation</u>
- Iterate on <u>Combine Outcome</u>
- Are we okay with outcomes as written in ED?
 - From last week, Non-standard commands is different than Conflicting keyboard command and Consistent keyboard interaction – but should be in closer proximity

10 December 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Attendance: Bruce, Gregg, Azlan, Glenda, Francis

Regrets: Gregg may be on Jury Duty 12/12

Agenda

- Just one more stand-alone session (Thursday, Noon Boston)
- Bruce will poll for alternative time
- Review 2.4.4 Keyboard Operation from most recent (12/12/2024) Editors Draft?

We discussed the purpose of Decision Trees. We all like the mechanic. Confusing to have label of "Which foundation requirements apply?"

2.3.1 Keyboard Focus appearance is closest from Editors Draft.

Current Goal: Users can navigate and operate content using only the keyboard focus.

Proposed: Uses can

Comparable keyboard effort

The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.

Conflicting keyboard commands

Authored keyboard commands do not conflict with platform commands or they can be remapped.

Consistent keyboard interaction

Keyboard interface interactions are consistent.

Keyboard mode

If the keyboard is non-hardware (such as a virtual keyboard), the keyboard input mode is indicated.

Keyboard only

All functionality must be accessible through the keyboard, except when a task requires input based on the user's specific input action.

No keyboard trap

If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component, or the component can be dismissed, with focus returning to the previous point.

Non-standard commands

The user is informed of non-standard authored keyboard commands.

12 December 2024

This area is used for decisions, in-progress scratpads, and other rough notes.

Attendance: Bruce, Gregg, Glenda, Lori, Poornima, Sarah, Giacommo

Agenda

- Clean up colors/highlights on recent draft
- Next year: start defining standard function

17 December 2024

Attendance: John Kirkwood, Steve Faulkner, Francis Storr, Mike Beganyi

We met in AGWG's informal subgroup meeting. We didn't update the document but did note that virtual keyboards are essentially not mentioned. Has the subgroup looked at virtual keyboards and de-scoped them or is this an area that needs attention? There was discussion that:

1. They could be seen as OS-level and therefore de-scoped

- 2. Is there anything relating to custom keyboards (e.g. number pads, decimal pads, etc.) being used appropriately for inputting numbers, currency, etc.?
- Virtual keyboards could get tricky quickly: you have ones built into phones, tablets, "desktop" operating systems, etc., but then there are also <u>laser keyboards</u> which are more gimmicky.

7 January 2025

Attendance: Sarah, Azlan, Poornima, John Tolles, Gregg Vanderheidan, Bruce Bailey

2nd hour of Tuesday AGWG meeting. Folks on call were mostly new to the <u>Composite</u> draft, so we took advantage of that to improve its readability. Foundational Requirement and What to Do are largely (but not completely) duplicative, so that is a bit confusing. We started a fresh comparison to Focus Appearance from the most recent <u>WCAG3 public draft</u>.

14 January 2025

Attendance: Bruce, Giacomo, Glenda, Gregg

2nd hour of Tuesday AGWG meeting.

Tried using <u>keyboard doc from meeting</u>, but abandoned quickly. Group iterated on Foundational Requirements.

Note: Zoom call 16 January 11:00, see email invite from Kevin.

16 January 2025

Attendance: Bruce, Gregg, Azlan, Jen, Giacomo

Regrets: Glenda, Poornima

Zoom call 16 January 11:00

Bruce asks: are there pages where we do not have to worry about keyboard access? (Noting that "keyboard" and "keyboard interface" is standing in for other types of operation and AT.)

Thought experiment: HTML representation of Magna Carta, historical correspondence (e.g. love letter).

Headings would be important to navigation, but those are not input. List items being marked up correctly are another example (These are covered under 2.x SC 1.3.1, but are not input.)

Pages that have no actionable content.

Back button (and other similar functionality) are part of browser, so also out of scope).

Hypertext links: Very common (of course) and accessible.

Group iterated on testing of foundational requirements.

21 January 2025

The Keyboard Only Access subgroup is tasked with the larger issue of Input. Please see <u>Input subgroup Google Doc</u>.

User Needs

User Need(s) (I collapsed all of them into this list)

(When these were distributed across 4 outcomes -- many of these were repeated in different outcomes (demonstrating the overlap in the old outcomes.) Where something appeared under multiple outcomes the number of different outcomes (out of 4) that I appeared is shown in curly brackets and also colored turquoise [3].

1. VIEW/OPERATE ALL -

- 4 As a keyboard user, I should be able to perform all actions using a keyboard exclusively
- 19 As a keyboard user, I am not expected to use a certain viewport in order to use a keyboard
- 22 As a keyboard user, I should be able to access all scrollable regions of a view/app so I can read all text within these regions
- 25 As a user, I don't want a keyboard shortcut to be the only way to do something. (Or, there should not be only one way to do something.)

2. NOT GET STUCK -

- 5 As a keyboard user, I should not get unintentionally stuck (3)
- As a keyboard user, scrolling is not trapped with screen zooming
- As a keyboard user, I should be able to identify the exit from the active component e.g. modal, menu
- As a keyboard user, I could use consistent (standard) keyboard commands to exit out of the active component e.g. 'ESC command to close a modal'
- NEW User Need: Nothing pops up that blocks view of focus or that cannot be dismissed or moved using the keyboard only

3. SAME COMPLEXITY-STEPS

- 3 As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.
- 23 As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.
- 2 As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.
- 15 As a keyboard user, I would like my user agent to provide keyboard shortcuts to quickly navigate around an interface.

4. TIMING NOT BARRIER -

7 As a keyboard user, timing should not prevent me from completing actions.

5. SKIP REPETITIVE -

 14 As a keyboard user, I should be provided with a method of skipping repetitive navigation links {2}

6. KNOW WHERE I AM -

- 9 As a keyboard user, I always know where I am in the interface (2)
- 18 As a keyboard user, I am warned about changes in focus before they occur.

o 17 As a keyboard user, I am warned about changes in context before they occur.

7. CONSISTENT, STANDARD or DESCRIBED -

- 1 As a user of assistive technology, I need my keyboard commands to remain consistent.
- As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- As a keyboard user, I shouldn't have my expected keyboard commands disabled
 (2)
- 10 As a keyboard user, I am informed of non-standard commands before I need to make use of one. <a>{4}
- 11 As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- 24 As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented.
- 16 If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands. <a>{2}

8. NOT CONFLICT WITH AT -

- 6 As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- 12 As a keyboard user, the keyboard commands that I need to press don't interfere with my operating system and/or assistive technology
- 21 As a keyboard user, I shouldn't have my expected keyboard commands disabled.

9. CAN CUSTOMIZE.

- 20 As a keyboard user, applications should be flexible in allowing me to use the keyboard commands I want to use (e.g. through personalisation)
- 10. The following are solved by operating systems or the fact that software gets its keystrokes from the "keyboard interface/API" and not from the keyboard -- so there is no need to address these with WCAG3 requirements
 - 8 As a keyboard user, I should not be required to press multiple keys at the same time to complete an action
 - (The StickyKeys feature is available on all keyboards directly (on screen keyboards) or via the OS -- and StickyKeys allows you to press modifier keys in sequence to create key combinations - so no need for it to be an author requirement)
 - 13 As a keyboard user, my choice of keyboard is fully supported
 - (All web content gets its keystrokes from a keyboard interface so it has no idea what keyboard generated them. So no need or possibility for author control of this)

OUTCOME & Foundation Requirements

COMPOSITE OUTCOME APPROACH: Keyboard Only Access

As of 21 Jan 2025 the Keyboard Only Access subgroup is asked to shift to the larger issue of Input. Please see Input subgroup Google Doc.

This page is now OBSOLETE - GO to INPUTS page for further work on this topic

Outcome:

Except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints:

It is possible to perceive, navigate, and operate the whole page using the keyboard only, where:

- keyboard navigation commands are either standard or they are described by the author prior to their being needed,
- navigation movement is in a logical order and be consistent moving forward and backward
- navigation allows access to all viewable and actionable items
- navigation focus does not get trapped anywhere
 - The focus can move in and out of any content using the keyboard only.

Supplemental recommendation

 Keyboard effort: The number of input commands required to complete a task using the keyboard is not excessively or unnecessarily more than the number of commands when using other input modalities (pointer or voice). Users who cannot use a pointing device (e.g., mouse or touchpad) - or who just use a keyboard or alternate keyboard (including Morse Code and other "keyboards without keys") can fully view and fully operate a web page.

NOTE: This deals with keyboard navigation directly - and does not cover keyboard navigation when the keyboard commands are being absorbed by an assistive technology (e.g. a person using a screen reader) - and the assistive technology (e.g. screen reader) is actually doing the navigation of the page.

Normative Text

Guideline

Users can navigate and operate content using only the keyboard

Which foundational requirements apply?

- this does not seem to make sense? These questions (in current published Working draft) look like "which foundational requirements need to be met?" So they are actually conditions for applying the requirements.
- Anything that Relates to 'whether or not this thing would apply to a page or wigit' should be included in the provision because it would be normative. That is, it would be a condition for whether or not the requirement applied. So any information about whether or not they apply should be in the provision, not above
- If this is going to be testing, whether or not they are met then we would suggest that it be located below (in the What to do" section, rather than above the requirements
- Finally, we don't think that it should no longer say continue if you pass. They
 should all just say whether or not it fails on that particular item. If something
 fails on one item, we don't think that people should stop checking to see whether
 or not the rest of the items are met or not.

Foundational Requirements

- a. Actionable using standard keys or Described: If any keyboard action essential to navigate, view and operate the page is not in the <u>list of standard keyboard navigation</u> <u>techniques</u> then it is described on the page where it is required or on a page earlier in the process where it is required.
 - i. Best practices
 - Avoid non-standard keys where the standard keys would be just as effective

- If you do use non-standard keys for efficiency also support navigation with standard keys
- Allow user to change any non-standard keyboard commands
- Have a description of any non-standard techniques available on every page they are needed so no memory is required.
 - Very best practice is that any non-standard are described or reminded wherever they are needed.

- NOTES:

- Any platform related functions are not the responsibility of the author as long as they are not defeated by the content
 - Examples
 - Tab and shift tab to move through elements
 - stickyKeys function that allows single key activation of multi-key commands
- b. **Bidirectional:** It is possible to move forward and backward at each point in the navigation.
- c. **Not Trapped:** it is always possible to navigate away from an element using standard keys after navigating to it or entering or activating the element
 - i. (Not a trap if there is a standard or described way to exit the element and proceed and go back in navigation order)
- d. No Unwarned jumps: The navigation focus does not suddenly change in a non-typical fashion without warning
- e. **Responsive:** If page uses responsive design (e.g. tabs collapse into hamburger menu) the page remains fully keyboard navigable
- f. **All content available :** All content that is available to pointer users (including hovers, right clicks, etc.) or via audio (voice or other), gesture, camera input, or other means is available using keyboard navigation only
- g. **All elements actionable:** All elements that can be controlled or activated by pointer, audio (voice or other), gesture, camera input, or other means can be controlled or activated from the keyboard

(Disabled and {unchangable} controls are not included in tab order - per HTML standard)

What to do

Test Procedure

Navigate bidirectionally to all items in the page, and activate any that are actionable using the keyboard only noting whether each of the following is true (all are required):

a. Actionable using standard keys or Described:

- i. If any keyboard action essential to navigate the page is not in the <u>list of standard</u> <u>keyboard navigation techniques</u> then it is described on the page where it is required or on a page earlier in the process where it is required.
- ii. Procedure
- 1. Navigate through all of the content including all actionable items the page
- Check whether you can reach all of the items without being trapped and using only commands that are found in this <u>list of</u> <u>standard keyboard navigation techniques.</u>
- Check whether any keyboard commands that are needed but that are not in the list are described on the page where it is required or on a page earlier in the process where it is required.
- 4. If check 2 or 3 is true then pass.

iii. Best practices

- Avoid non-standard keys where the standard keys would be just as effective
- If you do use non-standard keys for efficiency also support navigation with standard keys
- Allow user to change any non-standard keyboard commands
- Have a description of any non-standard techniques available on every page they are needed so no memory is required.
 - Very best practice is that any non-standard are described or reminded wherever they are needed.

NOTES:

- Any platform related functions are not the responsibility of the author as long as they are not defeated by the content
 - Examples
 - Tab and shift tab to move through elements
 - stickyKeys function that allows single key activation of multi-key commands
- Bidirectional: It is possible to move forward and backward at each point in the navigation. {False = fail}
- Not Trapped: it is always possible to navigate away from an element using standard keys after navigating to it or entering or activating the element {False = fail}
 - i. (Not a trap if there is a standard or described way to exit the element and proceed and go back in navigation order)
- d. No Unwarned jumps: The navigation focus does not suddenly change in a non-typical fashion without warning {False = fail}
- e. **Responsive:** If page uses responsive design (e.g. tabs collapse into hamburger menu) the page remains fully keyboard navigable **{False = fail}**
- **f. All content available :** All content that is available to pointer users (including hovers, right clicks, etc.) is available using keyboard navigation only {False = fail}

- g. **All elements actionable:** All elements that can be controlled or activated by pointer, voice or other means can be controlled or activated from the keyboard **{False = fail}**
 - i. (Disabled and {unchangable} controls are not included in tab order per HTML standard)

Knowing only the standard keyboard navigation techniques and what is described on the page, check that you can view and operate all of the content on the page using a keyboard only and that the navigation is consistent across pages.

SEE ALSO 2.4.3 focus order - because it impacts reading order and overlaps with [Users can navigate and operate content using only the] Keyboard

Notes from 7 January

For each [page/view] with actionable elements:

Note: "keyboard" includes keyboard emulation such as switch input.

Can we reorder/sort items?

- Bidirectional
- Standard Keys or Described
- Not Trapped
- Logical order
- No Unwarned jumps
- Responsive
- All content available
- All elements actionable

Are some dependent on others?

Is all content available to a keyboard? No, stop, fail.

Definitions

Definitions

STANDARD KEYBOARD NAVIGATION FUNCTIONS

In order to require that vendors document any techniques that are not standard - we must define what we mean by **standard keyboard functions** (that they do **not** need to document). The following is proposed as being such a definition.

REQUIREMENTS for inclusion in Definition of Standard Keyboard Navigation Commands

- 1. They are commonly known and used by keyboard-only navigators
- present on all keyboards (PC, Mac, Chrombook, etc.) that are not pointer operated (e.g. onscreen keyboards- that are operated by a mouse/finger/pointer already)

DEFINITION

Standard Keyboard Navigation Commands

Only the following navigation commands, which can be issued from all standard keyboards with simple single or key combinations without any timing requirements and which use only the following keys, which are present on all standard non-pointer-based keyboards; TAB, ESC, RETURN, SPACEBAR, SHIFT, CTRL, ALT/OPTION,0-9:

a. Tab

- i. moves you forward to next actionable element
- ii. If in a text field it may enter a TAB in the text or it may indent text or bullet
 - 1. (in this case CTRL-TAB or ESC then TAB is used to move to next actionable item)

b. ShiftTab

- i. moves backwards to previous actionable element
- ii. If in a text field it may reduce indent of text or bullet
 - 1. (in this case CTRL-TAB or ESC then TAB is used to move to next actionable item)

c. ArrowKeys

- i. move you forward and backward (or up and down) within a group of selections after you have entered a group or control (e.g. dropdown list)
 - 1. move you up or down in a list or dropdown
- ii. move you back and forth (or up and down) in a text field
- iii. move you around in a table or grid
- iv. move slider back and forth after having selected/entered it

d. Home, End

i. Home moves focus to the first focusable item, End moves focus to the last focusable item in a component.

e. F6, Shift+F6

Moves focus to a next view/group, with Shift to a previous view/group.
 Also allows skipping out of complex controls (a component which exists of many other components, like an editable table).

f. SpaceBar

- i. Activate item you are focused on (unless you are in a text entry field in which case it types a space)
- ii. toggle "toggle controls" on and off
- iii. When it in text field it moves you around one character or line at a time.

g. Enter/Return =

- i. Activates item item you are
- ii. Activates the default button on page (e.g submit, or OK)?

h. CTRL- Enter/Return

i. (anything here?)

i. Escape

- i. Exit or deselect something you have entered.
- j. (??? ANY OTHER CTRL-KEY COMMANDS? Or
 - i. CTRL-TAB for moving between groups?)

NOTE 1: Other commands listed in ARIA can also be supported <u>but are not "standard keyboard navigation commands"</u> if not listed above because they do not use keys present on all keyboard or are not commonly known. They can be optional ways to do something but do not qualify as "standard keyboard navigation" commands for use here) (https://www.w3.org/WAI/ARIA/apg/patterns/,

https://www.w3.org/WAI/ARIA/apg/practices/keyboard-interface

NOTE 2: Other common keyboard commands for navigation that would not qualify for list above because they do not exist on all keyboards are Ctrl+Home, Ctrl+End, F6 (to navigate between frames), Shift+Home, Shift+End. Providing these in *addition* to full navigation without them is good to do however if they allow faster access when the user is using one of these keyboards.

For complex interactive elements --

- WHAT ARE THE STANDARD METHODS?
- DO THEY VARY BY PLATFORM?
- DO WE DEFINE THESE -- OR JUST LET THEM BE SINCE THEY ARE SUBSETS OF THE KEYS ABOVE? OR JUST GENERICALLY LIST THEM UNDER THE ITEMS ABOVE (GV: I TOOK A CRACK AT DOING THIS ABOVE. Does the above cover the below?)

k. Tab listS

- i. ArrowKeys? Modified? TabKeys?
- Select list (combo boxes)
 - i. Arrowkey and spacebar ??
- m. Grids (interactive)
 - i. (need to define what grids means)
 - ii. How to navigate?
- n. Slider arrow keys vs. using thumbs
 - i. Expected behavior of these controls
- o. Media controls,
- p. Toggle switches
- q. Radio buttons
 - Expected keyboard commands to select and move on to next area outside of choice

NON-STANDARD BUT COMMONLY DEFINED KEYBOARD NAVIGATION COMMANDS

MEDIA CONTROLS

Change in non-typical fashion

DRAFT from AG call

Keyboard (from AG meeting, 14 Jan 2025)

Guideline

Users who cannot use a pointing device (e.g., mouse or touchpad) - or who just use a keyboard or alternate keyboard (including Morse Code and other "keyboards without keys") can fully view and fully operate functionality in a view.

NOTE: This deals with keyboard navigation directly - and does not cover keyboard navigation when the keyboard commands are being absorbed by an assistive technology (e.g. a person using a screen reader) - and the assistive technology (e.g. screen reader) is actually doing the navigation of the view.

Foundational Requirements

For each active user-interface element, all enumerated items below must be true:

- 1. Can the functionality of the user-interface element can be activated by a <u>keyboard</u> interface?
 - o Yes, keyboard access is enabled, continue.
 - No, fail. Keep testing for other failures.
- 2. Is it always possible to navigate away from an element after navigating to it or entering or activating the element?
 - Yes, keyboard focus is not trapped, continue.
 - o No. fail.
- 3. Is it possible to move forward or backward at each point in the view?
 - Yes, bi-directional keyboard use is enabled, pass.
 - o No, fail.

Exception

 Where the underlying function requires input that depends on the path of the user's movement and not just the endpoints, this guideline is not applicable.

Testing of the Foundational Requirements

Navigate bidirectionally to all user-interface elements in the view, and activate any that are actionable using the keyboard.

- All functionality in the view must be operable with the keyboard interface.

- The keyboard focus must not be trapped.

Foundational requirements

- Logical order: Keyboard focus movements follow a logical pattern.
- No Unwarned jumps: The navigation focus does not suddenly change in a non-typical fashion without warning.
- **Responsive:** If a view uses responsive design (e.g. navigation tabs at top of screen collapse into hamburger menu) the view remains fully keyboard navigable at each breakpoint.
- Standard Keys or Described: All keyboard actions are either in the list of standard keyboard navigation techniques or described before the place where the user needs to use them.

_

Method - Keyboard access is enabled

User-interface elements can be controlled with the keyboard.

Techniques

- HTML: Use default HTML markup
- HTML: Use keyboard-compatible events in scripts.
- Non-markup: Use keyboard-compatible events in scripts.

Method - Keyboard focus is not trapped

It is always possible to navigate away from an element using standard keys after navigating to it or entering or activating the element

 NOTE: It is not considered a trap if there is a standard or described way to exit the element and proceed and go back in navigation order

Techniques

-

Method - bi-directional keyboard use is enabled

It is possible to move forward or backward at each point in the view.

Techniques

_

BONEYARD

Keyboard Input silver subgroup

6.2 Keyboard Input (Developing to Exploratory)

Background

- Update 10/8: Meeting on Tuesdays 10:00 Eastern starting 10/15 through to 11/19.
- Update (9/6): Keyboard subgroup meeting via Zoom on Wednesdays 11:00 Eastern starting 8/14.
 - o For invite, please email bailey at access hyphen board dot u s
- Process
- Subgroup Handbook
- Keyboard Support Scratchpad
- Consolidated Outcomes
 - Usable
 - Comparable keyboard effort The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.
 - Consistent keyboard interaction Keyboard interface interactions are consistent.
 - **Keyboard commands** Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.
 - Operable (potential prerequisites)
 - **Keyboard only** All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.
 - No keyboard trap If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.
- Focus Appearance <u>decision tree</u> is a good model to emulate.

Could we have one outcome that encapsulates Comparable keyboard effort, Consistent keyboard interaction, Keyboard only, and No keyboard trap? "Keyboard only" being the core or umbrella expectation. "Keyboard only" may be a less challenging start. Custom keyboard command and No keyboard trap are the most tangible.

Work specifically on "Keyboard Commands"?

Then brainstorm methods that will give endpoints to a decision tree.

Additional Background

Closest WCAG 2.x Guideline

- Guideline 2.1: Make all functionality available from the keyboard.
- Parent category from consolidated outcomes
 - o 6 Input / Operation

Outcome: 6.2.1 Comparable keyboard effort

The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.

Goal

The input effort required to complete a task should be equivalent regardless of the input modality being used.

What to do

Complete tasks using only keyboard commands. Assess the number of commands required to complete tasks using only keyboard commands. Assess the effort needed to complete tasks using other modalities (such as using a mouse). Assess if the level of effort is similar between keyboard and other modality.

User Need(s)

- As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.
- As a keyboard user, I should be provided with a method of skipping repetitive navigation links
- As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.
- As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.
- As a keyboard user, timing should not prevent me from completing actions.

Outcome: 6.2.2 Consistent keyboard interaction

Keyboard interface interactions are consistent.

Goal

The keyboard experience when using web content is what I have come to expect. If the keyboard interaction is not conventional, then I am informed ahead of time about the work-arounds needed

What to do

You would know you have met the outcome if the content requires only use of tab, shift-tab, arrow keys, spacebar, enter, and escape keys for navigation. Alternatively, instructions would detail keyboard use of the content, and would not use keystrokes required by assistive technology.

You would know you have met the outcome if the content requires only the fundamental keyboard conventions. Instructions would detail keyboard use of the content. Keystrokes would not use keystrokes required by assistive technology.

User Need(s)

- As a keyboard user, I always know where I am in the interface.
- As a user of assistive technology, I need my keyboard commands to remain consistent.
- As a keyboard user, I should not get unintentionally stuck.
- If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.
- As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented

Outcome: 6.2.3 Custom keyboard commands

Application keyboard commands do not conflict with platform commands, and the user is informed of non-standard commands.

Goal

Keyboard interaction including navigation and operation, is intuitive for users familiar with the platform.

What to do

- As an author, any keyboard commands implemented in content do not conflict with keyboard usage of the platform.
- As an author, documentation for any keyboard commands implemented in content are readily available to users.
- An author would know that they have met this outcome if they have not added keyboard shortcut or accelerator keys to their content.

- An author would know that they have met this outcome if all added keyboard shortcuts are documented and readily available in the content.
- Content authors who add keyboard shortcuts must be well familiar with using a platform via only the keyboard.

User Need(s)

For each application keyboard command, there is not an overlap with platform commands.

Strongly related to:

- As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- As a keyboard user, the keyboard commands that I need to press don't interfere with my
 operating system and/or assistive technology
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
- As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented

Outcome: 6.2.4 Keyboard only

All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

Goal

All user flows can be completed by keyboard only.

What to do

This outcome would be met if you are able to complete all tasks by only the use of keyboard actions for both navigation and input. Exception is permissible where the action of completing a task is not only dependent upon the task having been completed but also requires specific user movement (e.g. requirement that the user must drag an object to follow specified path).

User Need(s)

- 4 As a keyboard user, I should be able to perform all actions using a keyboard exclusively
- 5 As a keyboard user, I should not get unintentionally stuck.
- 7 As a keyboard user, timing should not prevent me from completing actions.
- 9 As a keyboard user, I always know where I am in the interface

- 14 As a keyboard user, I should be provided with a method of skipping repetitive navigation links
- 19 As a keyboard user, I am not expected to use a certain viewport in order to use a keyboard
- 22 As a keyboard user, I should be able to access all scrollable regions of a view/app so
 I can read all text within these regions
- 23 As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.
- 2 As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.
- 3 As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.
- 1 As a user of assistive technology, I need my keyboard commands to remain consistent.
- 6 As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- 8 As a keyboard user, I should not be required to press multiple keys at the same time to complete an action
- 10 As a keyboard user, I am informed of non-standard commands before I need to make use of one
- 11 As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- 12 As a keyboard user, the keyboard commands that I need to press don't interfere with my operating system and/or assistive technology
- 13 As a keyboard user, my choice of keyboard is fully supported
- 15 As a keyboard user, I would like my user agent to provide keyboard shortcuts to quickly navigate around an interface.
- 20 As a keyboard user, applications should be flexible in allowing me to use the keyboard commands I want to use (e.g. through personalisation)
- 21 As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- 24 As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented
- 16 If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.

Decision Tree for 6.2.4 Keyboard only

- 1. Check if tabbing can cycle through all UIE and return back to where started.
 - a. Check if the page has any UIE that can be focused and activated with pointer input but not with the keyboard. If this is the case, the outcome fails.
 - b. Interact with each UIE, for example, activate it to change state, go to other pages, execute functions, or provide sample text input. Check if this is possible with the keyboard. If not, the outcome fails.
 - c. What is content requirement for exposing interactive features so they are usable via AT? (for example "forms mode")

- i. May be part of method. Relates to AT as UA.
- d. Shift-tab also needs to be tested
- 2. For elements that can be activated or followed (execute a link) or submitting a button
 - a. Enter or spacebar work to trigger / activate that element
- 3. For complex interactive elements
 - a. Tab list
 - i. Can arrows be used for interaction?
 - b. Select list (combo boxes)
 - i. Can arrows be used for interaction?
 - c. Grids (interactive)
 - d. Slider arrow keys vs. using thumbs
 - i. Expected behaviour of these controls
 - e. Media controls, toggle switches
 - f. Radio buttons expected keyboard commands to select and move on to next area outside of choice
- 4. Actionable elements that receive focus, can be activated with space bar, enter key or other documented keys
 - a. Disabled vs. readonly? Nature of an element involves activation, how is this processed?
 - i. Readonly focus but not editable
 - ii. Disabled wouldn't receive focus
- 5. Are there UI elements which are responsive to mouse but not keyboard?
 - a. Importance of focus and activation
 - i. Works with keyboard only is it AT provider need to update a keyboard only issue on vendor side? Does this fail 2.1.1 or 4.1.2 or otherwise?
 - 1. Dependent on code vs. AT issue.
- 6. Check viewport at reduced size.
 - a. With responsive design, on smaller screens, it is common for navigation elements to collapse into a "hamburger menu". If applicable, can this menu be navigated by keyboard?
 - b. One technique for different viewport sizes, which might be otherwise overlooked (even when default viewport size).
- 7. Focus order also impacts reading order, overlaps with keyboard (2.x 2.4.3 focus order)

Outcome: 6.2.5 No keyboard trap

If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.

Goal

Users are able to navigate content using only the keyboard without getting stuck. Users are able to access components and content and identify the exit (or) ways to exit clearly using keyboard-only.

What to do

Ensure that standard keyboard operation may be used for navigating through content, including moving focus from components. When non-standard keyboard operation is needed for navigation or to move focus from components, those are documented before being encountered.

User Need(s)

- As a keyboard user, I should not get unintentionally stuck.
- As a keyboard user, timing should not prevent me from completing actions.
- As a keyboard user, scrolling is not trapped with screen zooming
- As a keyboard user, I should be able to identify the exit from the active component e.g. modal, menu
- As a keyboard user, I could use consistent (standard) keyboard commands to exit out of the active component e.g. 'ESC command to close a modal'
- As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
- As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented.

Unassigned User Needs

User needs from previous scratchpad that were *not* brought over from the <u>User Need section of older scratchpad</u>. These might belong to another Outcome or Guideline.

- As a keyboard user, I am warned about changes in context before they occur. [17]
- As a keyboard user, I am warned about changes in focus before they occur. [18]
- As a user, I don't want a keyboard shortcut to be the only way to do something. (Or, there should not be only one way to do something.) [25]

Decision Tree for 6.2.5 No keyboard trap

- 1. Can tab through entire page, not getting stopped on UI element
 - a. Even if shift-tab to get out, traditionally this failure under 2.x 2.4.2.focus order
 - b. Might count as Keyboard Trap for 3.x
- 2. With 2.x "key trap" appears only after entering some content.
- 3. With 2.x, after skipping entry (leave empty), focus resets to top of page or returns focus to element.

Methods

Sample Method Decision Tree (text alternatives for images)

- 1. Does the organization have a style guide which includes guidance on all types of text-alternatives, and a policy that editors are required to follow the style guide?
 - 1.1. Yes, See <u>Text Alternative Style Guide Assertion</u> (Enhanced), continue
 - 1.2. No, continue
- 2. Would removing the image impact how people understand the page?
 - 2.1. No, see <u>Decorative Image Methods</u> (Enhanced) [stop]
 - 2.2. Yes, continue
- 3. Can a concise and plain-text description provide an equivalent to the image?
 - 3.1. No, see Complex Image Methods (Baseline) [stop]
 - 3.2. Yes, continue
- 4. Is the image a control?
 - 4.1. Yes, see <u>Image as Control Methods</u> (Baseline) [stop]
 - 4.2. No, continue
- 5. Is the image one of a set of images which provide multiple views of the same subject matter?
 - 5.1. Yes, see Multiple Image Views Methods (Baseline) [stop]
 - 5.2. No, see <u>Image Description Methods</u> (Baseline)
- 6. Is the image used for conveying information (e.g. a content image)?
 - 6.1. See <u>Image Description Method</u> (Baseline) [stop]
- 7. Any other image scenario
 - 7.1. See Alternative method [stop]

Method Decision Tree for 6.2.3 Custom keyboard command

6.2.3 Custom keyboard commands (take 2, 8/14)

For each type of action where the content sets a keyboard command.:

- 1. Does the content owner document unique commands
 - a. Yes
 - b. No
- 2. Does the content allow a setting for changing keyboard commands?
 - a. Yes "Override content level command", Stop
 - b. No, continue.

- 3. Does the content allow a setting for disabling all keyboard commands?
 - a. Yes [method]
 - b. No.
- 4. Does the platform allow means control over keyboard command.
 - a. Yes [method]
 - b. No, see Alternative Means

Alternative means:

- 1. Content does not meet this requirement.
- 2. Accessibility supported might allow Assistive Technology which has a "pass thru" function.
- 3.

Method list

[Method Name]

Level

[One of Prerequisite/Baseline/Enhanced]

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]
- (HTML, EPUB) Using alt attributes so the image is ignored. Example

Override Platform Level Command

Level

[One of Prerequisite/Baseline/Enhanced]

Baseline

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]
- (HTML, EPUB) Using alt attributes so the image is ignored. Example

Document Unique Command

Comprehensive platform list does not exist.

Commands vary from platform to platform.

Example copy/paste is probably easy, but other keyboard functionality can be problematic.

Level

[One of Prerequisite/Baseline/Enhanced]
Baseline

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]
- (HTML, EPUB) Using alt attributes so the image is ignored. Example

[Assertion name]

Level

[One of Prerequisite/Baseline/Enhanced]

Test

• [Example] The organization has a documented style guide which includes guidance on text-alternatives, and a policy that editors are required to follow the style guide.

•

Alternative Means

What to do when no method meets outcome for Custom keyboard commands

If none of the above methods match the scenario you are faced with, you can meet the outcome in another way. It must meet the outcome and general information under "what to do".

Blank Template

Outcome: 6.2.4 Keyboard Only

Background

- Process
- Subgroup Handbook
- <u>Scratchpad</u> (6.2 Keyboard Input (Developing to Exploratory)

Outcome:

[The full outcome text from the list.]

Goal

[Plain english sentence of what the outcome should achieve, e.g. "Non-text information is available to more people."]

What to do

[In as plain-language as possible, outline how you would know if you have met the outcome. E.g. a general test procedure.]

User Need(s)

[List from the scratchpad any that are relevant to this outcome.]

Method Decision Tree

[Create a decision tree of scenarios. This should provide a logical path to the appropriate method. Always finish with a catch-all path. The method text should link to the method description further down the document.

The alt-text example is left below to assist.]

For each image:

- 1. Does the organization have a style guide which includes guidance on all types of text-alternatives, and a policy that editors are required to follow the style guide?
 - 1.1. Yes, See <u>Text Alternative Style Guide Assertion</u> (Enhanced), continue
 - 1.2. No, continue
- 2. Would removing the image impact how people understand the page?
 - 2.1. No, see <u>Decorative Image Methods</u> (Enhanced) [stop]
 - 2.2. Yes, continue
- 3. Can a concise and plain-text description provide an equivalent to the image?
 - 3.1. No, see Complex Image Methods (Baseline) [stop]
 - 3.2. Yes, continue
- 4. Is the image a control?
 - 4.1. Yes, see <u>Image as Control Methods</u> (Baseline) [stop]
 - 4.2. No, continue
- 5. Is the image one of a set of images which provide multiple views of the same subject matter?
 - 5.1. Yes, see Multiple Image Views Methods (Baseline) [stop]
 - 5.2. No, see <u>Image Description Methods</u> (Baseline)
- 6. Is the image used for conveying information (e.g. a content image)?
 - 6.1. See Image Description Method (Baseline) [stop]
- 7. Any other image scenario
 - 7.1. See <u>Alternative method</u> [stop]

For each [item of scope]:

Method list

[Method Name]

Level

[One of Prerequisite/Baseline/Enhanced]

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]
- (HTML, EPUB) Using alt attributes so the image is ignored. Example

[Method Name]

Level

[One of Prerequisite/Baseline/Enhanced]

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]
- (HTML, EPUB) Using alt attributes so the image is ignored. Example

[Method Name]

Level

[One of Prerequisite/Baseline/Enhanced]

Method

[Description of method, not technology specific]

Techniques

[List of technology/platform specific techniques.]

- ([Platforms]) [Method title]
- (HTML, EPUB) Using alt attributes so the image is ignored. Example

[Assertion name]

Level

[One of Prerequisite/Baseline/Enhanced]

Test

• [Example] The organization has a documented style guide which includes guidance on text-alternatives, and a policy that editors are required to follow the style guide.

•

Alternative method

If none of the above methods match the scenario you are faced with, you can meet the outcome in another way. It must meet the outcome and general information under "what to do".

Meeting Minutes

Use this for decisions, in-process scratchpads, and other rough notes.

Model Template

Focus Appearance (Exploratory to Developing)

Prior work on this topic: Focus appearance draft and testing material

[Guideline within which this Outcome is contained: "The *point of interaction* must be visually indicated"]

Outcome

The keyboard focus must have a sufficient visual indicator.

Editor's note: We would like to conduct research once we have an initial pass/fail set of examples to see if we are making decisions that support various disabilities. Please get in touch if you could facilitate this research.

Goal

Users can see which element has keyboard focus.

What to do

Ensure focusable elements include focus indicators that are visible for people using an input device that relies on the keyboard focus to navigate.

An outline, or indicator which surrounds the focusable element is generally the most easily visible.

All indicators for keyboard-focusable elements must be:

- Present: There is a visible indicator of keyboard focus.
- Not obscured: The focus indicator is not obscured or partially obscured (more than 50%, TBC)
- Persistent: The focus indicator persists while the element has focus, but does not persist after the element loses focus.
- Distinctive: The keyboard focus indicator uses a style that is distinct from the style of other controls, so that the item in focus can be distinguished without reference to the non-focused state.
- Sufficiently visible: According to the specific method (below), the indicator must be visually discernible whilst navigating.

Note: The tests are written for the visuals apparent in the interface, there could be multiple ways of implementing those visuals

Decision Tree and Methods

Method decision tree

For each focusable item:

- 1. Is there a focus indicator?
 - a. Yes, continue
 - b. No, fail.
- 2. Does the organization have a style guide which includes guidance on all types of text-alternatives, and a policy that editors are required to follow the style guide?
 - a. Yes, See Text Alternative Style Guide Assertion (Enhanced), continue
 - b. No, continue
- 3. Is the user-agent default focus indicator used? (Prerequisite)
 - a. Yes, see <u>default indicators</u> (continue)
 - b. No, continue.
- Is the user-agent default focus indicator used, and meets contrast across backgrounds?
 (Baseline)
 - a. Yes, see <u>default focus indicator and checking contrast</u> (stop)
 - b. No, continue
- 5. Is the indicator using lines or a bounding box? (Baseline)
 - a. Yes, see <u>Line / bounding box indicators</u> (stop)
 - b. No, continue
- 6. Does the indicator use a change of color of the element? (Baseline)
 - a. Yes, see Change of color (stop)
 - b. No, continue
- 7. Does the indicator use an icon or shape as the indicator? (Baseline)
 - a. Yes, see loon/shape (stop)
 - b. No, continue.
- 8. Does the indicator use a tooltip or additional content as the indicator? (Baseline)
 - a. Yes, see Tooltip/additional content.

Is there a supplementary indicator? (Enhanced)

- i. Yes, see <u>Supplementary indicators</u> (stop)
- ii. No, continue
- b. No, continue.
- 9. Does the indicator use a change of size of the focusable element as the indicator?

(Baseline)

- a. Yes, see Size change (stop)
- b. No, continue.
- 10. Does the indicator use a combination of the above indicators?
 - a. Pick one of the methods above to meet conformance (stop)
 - b. No, continue
- 11. Does the indicator use another method? (Baseline)
 - a. Yes, see Other indicators (stop)

Note: Links which appear on focus (e.g. skip links) are considered an example of "Additional content".

Methods

Focus indicators Follow an Organizational Style Guide - Assertion

Level: Enhanced

Assertion:

 The organization has a documented style guide which includes guidance on focus indicators, and a policy that designers and developers are required to follow the style guide. The style guide includes focus indicators that meet one of the baseline scenarios in the decision tree.

Using the default focus indicator

Method: The platform's default indicator is used.

Level: Prerequisite

Techniques

- **Platform:** Provide a focus indicator that meets the requirements for all focus indicators and bounding box indicators.
- Author technique (HTML, EPUB): Don't block the default user agent indicator and ensure that default indicator is present. The default indicator is not edited by the author.

Default focus indicator and checking contrast

Method: The platform's default indicator is used, and where the backgrounds are changed, the indicator contrast is checked.

Level: Baseline

 Author technique (HTML, EPUB): Don't block the default user agent indicator and ensure that default indicator is present and contrasts with background.
 [See line indicators technique/test]

Potential Platform suggestions for focus indicators:

- **Present**: There is a visible keyboard focus indicator.
- **Be persistent**: The focus indicator is persistent while the element has focus.
- **Unfocused**: The focus indicator does not persist after the element loses focus.
- Distinct Style: The keyboard focus indicator uses a style that is distinct from the style of
 other controls, so that the item in focus can be distinguished without reference to the
 non-focused state.
- Contrast: Provide an indicator that maximizes chance of sufficient contrast in high contrast and dark mode (two contrast indicator or indicator that changes contrast based on surrounding colors)
- **In Viewport**: By default (if the author hasn't introduced javascript to change viewport and the user has not changed the position of the viewport after focus), the focus indicator should be visible within the viewport.

Line / bounding box indicators

Method: The authored focus indicator uses a line or bounding box.

Level: Baseline

Techniques

 Author technique (HTML): A bounding box focus indicator using outline, border or box-shadow.

[Baseline level]

 Author technique (HTML): A single line focus indicator using border, box-shadow, text-decoration, or graphic.
 [Baseline level]

Tests/requirements within the method:

- **Sufficient Size:** The thickness/length of the focus indicator change is [a sufficient size to be noticeable].
 - Editor's note: We will use a metric of 2px of the perimeter -20% as a starting point. This metric needs usability testing, please get in touch if you could help facilitate this.
 - Note: Text within a control is not considered part of the size metric.
- Change of contrast: The contrast change of the pixels that indicate the keyboard focus is [sufficient] in a) standard mode, b) dark mode, and c) high contrast mode.
 - Editor's note: The contrast metric is to be determined, but assumed to be similar to a large-text level.
- Adjacent Contrast: The focus indicator has sufficient contrast with the adjacent colors in a) standard mode, b) dark mode, and c) high contrast mode

- Editor's Note: There is a relationship between the area of the focus indicator and contrast that will need to be clarified. E.g. a higher contrast area may not need to be as large.
- Note: If the focus indicator is adjacent to the focusable element and does not contrast with the element, it can use the "size change" method.
- Distinct Style: The keyboard focus indicator uses a style that is distinct from the style of
 other controls, so that the item in focus can be distinguished without reference to the
 non-focused state.
- **Adjacent:** The indicator is closer to the focused element than any other focusable element. (The center of the focused element to the center of the indicator.)

Change of color

Method: The authored focus indicator uses a change of color within the control.

Level: Baseline

- **Author technique (General):** Change the user-interface-component background by [some minimum amount].
- Author technique (General): Change the user-interface-component border color by [some minimum amount]

Tests/requirements within the methods:

- **Sufficient Size:** The thickness/length of the focus indicator change is [a sufficient size to be noticeable].
 - Editor's note: We will use a metric of 2px of the perimeter -20% as a starting point. This metric needs usability testing, please get in touch if you could help facilitate this.
 - Note: Text within a control is not considered part of the size metric.
- Change of contrast: The contrast change of the pixels that indicate the keyboard focus is [sufficient] in a) standard mode, b) dark mode, and c) high contrast mode.
 - Editor's note: The contrast metric is to be determined, but assumed to be similar to a large-text level.
- Distinct Style: The keyboard focus indicator uses a style that is distinct from the style of
 other controls, so that the item in focus can be distinguished without reference to the
 non-focused state.
- **Multiple controls:** There must be at least 3 visible controls of the same style in the view/page.
 - Note: In order for a change of color to be used, there needs to be a basis for comparison, otherwise another method is needed.

Icon/shape

Method: The authored focus indicator uses the addition/removal of an icon or shape.

Level: Baseline

- Author technique (General): Provide an icon to indicate the user interface component with focus.
- Author technique (General): Provide a shape to indicate the user interface component with focus.

Tests/requirements within the methods:

- **Sufficient Size:** The thickness/length of the focus indicator change is [a sufficient size to be noticeable].
 - Editor's note: We will use a metric of 2px of the perimeter -20% as a starting point. This metric needs usability testing, please get in touch if you could help facilitate this.
 - Note: Text within a control is not considered part of the size metric.
- **Change of contrast:** The contrast change of the pixels that indicate the keyboard focus is [sufficient] in a) standard mode, b) dark mode, and c) high contrast mode.
 - Editor's note: The contrast metric is to be determined, but assumed to be similar to a large-text level.
- Adjacent Contrast: The focus indicator has sufficient contrast with the adjacent colors in a) standard mode, b) dark mode, and c) high contrast mode
 - Editor's Note: There is a relationship between the area of the focus indicator and contrast that will need to be clarified. E.g. a higher contrast area may not need to be as large.
 - Note: If the focus indicator is adjacent to the focusable element and does not contrast with the element, it can use the "size change" method.
- Adjacent: The indicator is closer to the focused element than any other focusable element. (The center of the focused element to the center of the indicator.)

Tooltip/additional content

Method: The authored focus indicator uses additional content.

Level: Baseline.

• Author technique (General): Display a tooltip to indicate the user interface component with focus.

Tests/requirements within the methods:

- **Sufficient Size:** The thickness/length of the focus indicator change is [a sufficient size to be noticeable].
 - Editor's note: We will use a metric of 2px of the perimeter -20% as a starting point. This metric needs usability testing, please get in touch if you could help facilitate this.
 - O Note: Text within a control is not considered part of the size metric.
- Change of contrast: The contrast change of the pixels that indicate the keyboard focus

is [sufficient] in a) standard mode, b) dark mode, and c) high contrast mode.

- Editor's note: The contrast metric is to be determined, but assumed to be similar to a large-text level.
- Adjacent Contrast: The focus indicator has sufficient contrast with the adjacent colors in a) standard mode, b) dark mode, and c) high contrast mode
 - Editor's Note: There is a relationship between the area of the focus indicator and contrast that will need to be clarified. E.g. a higher contrast area may not need to be as large.
 - Note: If the focus indicator is adjacent to the focusable element and does not contrast with the element, it can use the "size change" method.

Size change

Method: The authored focus indicator uses a change of size of the focusable element.

Level: Baseline

 Author technique (General): Increase the size of the user interface component to indicate focus

Tests/requirements within the methods:

- **Sufficient Size:** The discernable area of the focus indicator change is [sufficient to be noticeable]
 - Editor's note: We will use a metric of 10% as a starting point. This metric needs usability testing, please get in touch if you could help facilitate this.
 - Note: "discernable area" means the focused element has a discernible border or background, it cannot just be text.
 - Note: Text within a control is not considered part of the size metric.
- **Contrast:** The contrast change of the pixels that indicate the keyboard focus is sufficient in a) standard mode, b) dark mode, and c) high contrast mode

Other focus indicators:

Method: Using a custom indicator that does not fit into any other method.

Level: Baseline

For an indicator which does not fit into other branches of the decision tree, the requirements for meeting the outcome are.

- **Sufficient Size**: The thickness/length of the focus indicator change is sufficient to be noticeable (metric TBC, use 2px perimeter of the (visually apparent) control as the default, should be relative to the visual size of the control.)
 - Note: Add the complexity of the adjacent contrast vs size of indicator.
- **Change of contrast**: The contrast change of the pixels that indicate the keyboard focus is sufficient in a) standard mode, b) dark mode, and c) high contrast mode.

- Adjacent Contrast: The focus indicator has sufficient contrast with the adjacent colors in a) standard mode, b) dark mode, and c) high contrast mode
 - Note: There is a relationship between the area of the focus indicator and contrast that will need to be clarified
 - There is a relationship between sufficient size and adjacent contrast as well when the contrast is the same between the indicator and adjacent item
 - If the focus indicator is adjacent to the focusable element and does not contrast with the element, it may pass via "size change" below.
- Distinct Style: The keyboard focus indicator uses a style that is distinct from the style of
 other controls, so that the item in focus can be distinguished without reference to the
 non-focused state.
- **Adjacent**: The indicator is closer to the focused element than any other focusable element. (The center of the focused element to the centre of the indicator.)

Supplementary indicators

Method: In addition to the focus indicator, there is another change to the focusable element. **Level:** Enhanced

 Author method (General): Add a supplementary change to the element with focus, such as an outline or background change.
 [Enhanced level]

6.2.1 Comparable keyboard effort

Outcome: 6.2.1 Comparable keyboard effort

The number of input commands required to complete a task using the keyboard is similar to the number of input commands when using other input modalities.

Goal

The input effort required to complete a task should be equivalent regardless of the input modality being used.

What to do

Complete tasks using only keyboard commands. Assess the number of commands required to complete tasks using only keyboard commands. Assess the effort needed to complete tasks using other modalities (such as using a mouse). Assess if the level of effort is similar between keyboard and other modality.

User Need(s)

- As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.
- As a keyboard user, I should be provided with a method of skipping repetitive navigation links
- As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.
- As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.
- As a keyboard user, timing should not prevent me from completing actions.

Note: Included in this outcome should be how much effort is needed in advance to customize the content. For example, ability to set customized keyboards is a pass for 6.2.3, but might fail 6.2.1 Comparable Effort because it is so much work before one can even start.

Prerequisite for testing is (1) user has average or above skill and (2) is familiar with the content.

We know that, usually, using a keyboard is not going to be as efficient as using a mouse.

Outcome might address need for content to provide extra support for keyboard user. For example, navigating by headings, having descriptive heading.

Number of keystrokes must be proportional to the number of elements.

Consider example of very long article filled with hundreds of links. What is responsible for author with facilitation of keyboard navigation?

6.2.2 Consistent Keyboard Interaction

Outcome: 6.2.2 Consistent keyboard interaction

Keyboard interface interactions are consistent.

Goal

The keyboard experience when using web content is what I have come to expect. If the keyboard interaction is not conventional, then I am informed ahead of time about the work-arounds needed

What to do

You would know you have met the outcome if the content requires only use of tab, shift-tab, arrow keys, spacebar, enter, and escape keys for navigation. Alternatively, instructions would detail keyboard use of the content, and would not use keystrokes required by assistive technology.

You would know you have met the outcome if the content requires only the fundamental keyboard conventions. Instructions would detail keyboard use of the content. Keystrokes would not use keystrokes required by assistive technology.

User Need(s)

- As a keyboard user, I always know where I am in the interface.
- As a user of assistive technology, I need my keyboard commands to remain consistent.
- As a keyboard user, I should not get unintentionally stuck.
- If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.
- As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented

Method Decision Tree for 6.2.2 Consistent Keyboard Interaction

Methods

[Create a decision tree of scenarios. This should provide a logical path to the appropriate method. Always finish with a catch-all path. The method text should link to the method description further down the document.

The alt-text example is left below to assist.]

:

Does the organization have a style guide which includes guidance and references for standard keyboard behaviour for the platform? Does the organization have a policy that developers are required to follow the style keyboard conventions for the platform?

Yes, See Text Alternative Style Guide Assertion (Enhanced), continue No, continue

For each UIE which is operable via a keyboard:

Would removing the image impact how people understand the page? No, see Decorative Image Methods (Enhanced) [stop] Yes, continue

Can a concise and plain-text description provide an equivalent to the image? No, see Complex Image Methods (Baseline) [stop] Yes, continue

Is the image a control? Yes, see Image as Control Methods (Baseline) [stop] No, continue

Is the image one of a set of images which provide multiple views of the same subject matter? Yes, see Multiple Image Views Methods (Baseline) [stop]
No, see Image Description Methods (Baseline)

Is the image used for conveying information (e.g. a content image)? See Image Description Method (Baseline) [stop]

Any other image scenario See Alternative method [stop]

6.2.3 Custom Keyboard Command

Outcome: 6.2.3 Custom keyboard commands

- 1. Application keyboard commands do not conflict with platform commands.
- 2. The user is informed of non-standard commands.

Goal

Keyboard interaction including navigation and operation, is intuitive for users familiar with the platform.

What to do

- As an author, any keyboard commands implemented in content do not conflict with keyboard usage of the platform.
- As an author, documentation for any keyboard commands implemented in content are readily available to users.
- An author would know that they have met this outcome if they have not added keyboard shortcut or accelerator keys to their content.
- An author would know that they have met this outcome if all added keyboard shortcuts are documented and readily available in the content.
- Content authors who add keyboard shortcuts must be well familiar with using a platform via only the keyboard.
 - Keyboard shortcuts should also be programmable if possible or they may conflict with AT commands
 - If programmable, they should be able to be saved for next use
 - This should be in either web application preferences or through use of browser / cookies
 - The author should also reference https://www.w3.org/TR/WCAG22/#character-key-shortcuts

User Need(s)

- 1. For each application keyboard command, there is not an overlap with platform commands.
- 2. For each application keyboard command, there is not an overlap with Assistive Technology (AT) commands

Strongly related to:

- As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- As a keyboard user, the keyboard commands that I need to press don't interfere with my operating system and/or assistive technology
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
 - This would be done via on screen / page information or related documentation per the above

- As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- If an author implements a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented

Method Decision Tree for 6.2.3 Custom keyboard command

6.2.3 Custom keyboard commands

For each type of control where the author specifies a keyboard command.:

- 1. Is the keyboard command acting as default for the platform.
 - a. Yes, pass.
 - b. No, continue.
- 2. Does the content owner document unique commands
 - a. Yes, continue.
 - b. No, skip to step 4.
- 3. Does the content allow a setting for changing keyboard commands for the content?
 - a. Yes, pass, method for changing keyboard command...
 - b. No, continue.
- 4. Does the content allow a setting for disabling all keyboard commands?
 - a. Yes, pass, method for disabling all keyboard commands.
 - b. No, continue.
- 5. Does the platform allow a means for control over keyboard command.
 - a. Yes [method]
 - b. No, see Alternative Means

Alternative means:

- 1. Content does not meet this requirement.
- 2. Accessibility supported might allow Assistive Technology which has a "pass thru" function.

3.

References from 10/29:

- Developing a Keyboard Interface | APG | WAI | W3C
- Grid (Interactive Tabular Data and Layout Containers) Pattern

6.2.4 Keyboard

Outcome: 6.2.4 Keyboard only

All functionality can be performed through the keyboard interface only, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

Goal

All user flows can be completed by keyboard only.

What to do

This outcome would be met if you are able to complete all tasks by only the use of keyboard actions for both navigation and input. Exception is permissible where the action of completing a task is not only dependent upon the task having been completed but also requires specific user movement (e.g. requirement that the user must drag an object to follow specified path).

User Need(s)

- 4 As a keyboard user, I should be able to perform all actions using a keyboard exclusively
- 5 As a keyboard user, I should not get unintentionally stuck.
- 7 As a keyboard user, timing should not prevent me from completing actions.
- 9 As a keyboard user, I always know where I am in the interface
- 14 As a keyboard user, I should be provided with a method of skipping repetitive navigation links
- 19 As a keyboard user, I am not expected to use a certain viewport in order to use a keyboard
- 22 As a keyboard user, I should be able to access all scrollable regions of a view/app so
 I can read all text within these regions
- 23 As a keyboard user, I need to be able to access content in an equivalent way as other input modalities.
- 2 As a user of a keyboard, I should be able to complete actions in a comparable amount of time to other users.
- 3 As a user of a keyboard, the complexity and number of input commands required to complete a task should be comparable with those of a user of other methods of input.
- 1 As a user of assistive technology, I need my keyboard commands to remain consistent.
- 6 As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- 8 As a keyboard user, I should not be required to press multiple keys at the same time to complete an action
- 10 As a keyboard user, I am informed of non-standard commands before I need to make use of one

- 11 As a keyboard user, information on available shortcuts is provided in context and in a way I can perceive.
- 12 As a keyboard user, the keyboard commands that I need to press don't interfere with my operating system and/or assistive technology
- 13 As a keyboard user, my choice of keyboard is fully supported
- 15 As a keyboard user, I would like my user agent to provide keyboard shortcuts to quickly navigate around an interface.
- 20 As a keyboard user, applications should be flexible in allowing me to use the keyboard commands I want to use (e.g. through personalisation)
- 21 As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- 24 As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented
- 16 If I implement a keyboard device within my site/app, that control must respond correctly to standard keyboard commands.

[new draft] Decision Tree for 6.2.4 Keyboard only

[old] Decision Tree for 6.2.4 Keyboard only

- 1. Check if tabbing can cycle through all User Interface Elements and return back to where one started.
 - a. Check if the page has any UIE that can be focused and activated with pointer input but not with the keyboard. If this is the case, the outcome fails.
 - b. Interact with each UIE, for example, activate it to change state, go to other pages, execute functions, or provide sample text input. Check if this is possible with the keyboard. If not, the outcome fails.
 - c. What is content requirement for exposing interactive features so they are usable via AT? (for example "forms mode")
 - i. May be part of method. Relates to AT as UA.
 - d. Shift-tab also needs to be tested
- 2. For elements that can be activated or followed (execute a link) or submitting a button
 - a. Enter or spacebar work to trigger / activate that element
- 3. For complex interactive elements
 - a. Tab list
 - i. Can arrows be used for interaction?
 - b. Select list (combo boxes)
 - i. Can arrows be used for interaction?
 - c. Grids (interactive)
 - d. Slider arrow keys vs. using thumbs

- i. Expected behavior of these controls
- e. Media controls, toggle switches
- f. Radio buttons expected keyboard commands to select and move on to next area outside of choice
- 4. Actionable elements that receive focus, can be activated with space bar, enter key or other documented keys
 - a. Disabled vs. readonly? Nature of an element involves activation, how is this processed?
 - i. Readonly focus but not editable
 - ii. Disabled wouldn't receive focus
- 5. Are there UI elements which are responsive to mouse but not keyboard?
 - a. Importance of focus and activation
 - i. Works with keyboard only is it AT provider need to update a keyboard only issue on vendor side? Does this fail 2.1.1 or 4.1.2 or otherwise?
 - 1. Dependent on code vs. AT issue.
- 6. Check viewport at reduced size.
 - a. With responsive design, on smaller screens, it is common for navigation elements to collapse into a "hamburger menu". If applicable, can this menu be navigated by keyboard?
 - b. One technique for different viewport sizes, which might be otherwise overlooked (even when default viewport size).
- 7. Focus order also impacts reading order, overlaps with keyboard (2.x 2.4.3 focus order)

6.2.5 No keyboard trap

Outcome: 6.2.5 No keyboard trap

If keyboard focus can be moved to an interactive component, then the keyboard focus can be moved away from that component.

Goal

Users are able to navigate content using only the keyboard without getting stuck. Users are able to access components and content and identify the exit (or) ways to exit clearly using keyboard-only.

What to do

Ensure that standard keyboard operation may be used for navigating through content, including moving focus from components. When non-standard keyboard operation is needed for navigation or to move focus from components, those are documented before being encountered.

User Need(s)

- As a keyboard user, I should not get unintentionally stuck.
- As a keyboard user, timing should not prevent me from completing actions.
- As a keyboard user, scrolling is not trapped with screen zooming
- As a keyboard user, I should be able to identify the exit from the active component e.g. modal, menu
- As a keyboard user, I could use consistent (standard) keyboard commands to exit out of the active component e.g. 'ESC command to close a modal'
- As a keyboard user, keyboard shortcuts and assistive technologies do not interfere with each other.
- As a keyboard user, I am informed of non-standard commands before I need to make use of one
- As a keyboard user, I shouldn't have my expected keyboard commands disabled.
- As a keyboard user, when I reach a custom implementation of a common component, all keyboard commands that exist are implemented.

Decision Tree for 6.2.5 No keyboard trap

- 1. Can tab through entire page, not getting stopped on UI element
 - a. Even if shift-tab to get out, traditionally this failure under 2.x 2.4.2.focus order
 - b. Might count as Keyboard Trap for 3.x
- 2. With 2.x "key trap" appears only after entering some content.
- 3. With 2.x, after skipping entry (leave empty), focus resets to top of page or returns focus to element.

Goodwitch Testing Methodology:

Goodwitch Testing Methodology:

Test with the screen reader turned OFF and ALSO with the screen reader turned ON.

- 1. Visually explore the screen to discover all of the functionality that is available to someone using a mouse.
- 2. Ensure that the equivalent functionality is available using only the keyboard.
- 3. If an interactive element is not keyboard accessible it may still pass this checkpoint if at least one of the following is true:
 - a. Equivalent functionality is available on the same screen in a keyboard-accessible way.
 - An alternate accessible version of the functionality is available on a different screen AND the link to the accessible alternate version is clearly identified on the current screen. (More information about Understanding Conforming Alternate Versions.)

4. Exception (rare):

- **a.** The underlying function requires input that depends on the path of the user's movement and not just the endpoints.
 - i. The input required for these activities cannot currently be done from the keyboard in any known fashion without requiring an inordinate number of keystrokes.
 - **ii.** Examples include activities like: freehand drawing, watercolor painting, and flying a helicopter through an obstacle course.

5. **KEYBOARD TRAP**

- Using only the keyboard as you navigate through the page, verify that keyboard focus is never trapped (stuck) on or in a component.
 - Use the TAB key to navigate focus among buttons, links, form inputs, links, menu items, tab panels, accordions, and other interactive UI elements.
 - Use SHIFT-TAB to navigate the focus in the reverse direction through all interactive UI elements.
 - o If TAB or SHIFT-TAB do not work, also try any and all of the following:
 - Standard exit keys (Enter, Spacebar, ESC)
 - Up, down, right, left arrow keys
- If moving away from a component requires more than TAB or SHIFT-TAB, unmodified arrow keys, Enter, Spacebar, or ESC, verify that the user is provided instructions for moving focus away.

Goodwitch testing method ends.