Script for Canvas RCE Accessibility - Structure (Heading and Lists)

Screen: *Narrator navigates to https://canvas.vt.edu/courses/69527/pages/structure-non-example-1?module_item_id=844607 in the Brave browser.*

Narrator: This is the Wikipedia page on Web Accessibility stripped of all formatting. Suppose I showed this to you and asked you to identify the three methods of performing an accessibility audit. How long would it take you to do it?

Screen: *Narrator scrolls down to the bottom of the page*

Narrator: I don't know exactly, but I'm pretty sure it would take you longer than if the document was formatted correctly with headings and lists so you could more quickly scan the page for relevant information.

Let's take a look at the same page with some *visual* formatting applied.

Screen: *Narrator clicks the **Next button** to navigate to https://canvas.vt.edu/courses/69527/pages/structure-non-example-2?module_item_id=844608 in the Brave browser.*

Narrator: Now, what strategies might you use to find the relevant information on this page? I don't know about you, but I would probably scan the page for headings related to web accessibility audits.

Screen: *Narrator scrolls down until he reaches the pseudo-heading, **Web accessibility audits** and triple-clicks on it to select it.*

Narrator: Ah, there it is! Now what was the question exactly? Rob wanted me to name the three methods of performing an accessibility audit.

Screen: *Narrator selects the pseudo-bulleted list below **Web accessibility audits**.*

Narrator: Oh, I see it; there is a list of three items under that heading. I bet that's it.

As you can see, providing some visual structure to a document can make a world of difference in finding the relevant information quickly.

However, despite the fact that we have communicated the structure of the document to sighted folks, blind people who use screen reading software would not have those same cues available

to them if *they* tried to access this page. If we look at how those headings were created, we can understand why.

*Screen: Narrator clicks the **Edit button** and scrolls down to the pseudo-heading, **Web accessibility audits**, and selects it.*

Narrator: The way I chose to make this text look like a heading was, first, by making it bold by pressing the **Bold button**

*Screen: Narrator hovers over the **Bold button***

And, second, by making the text bigger using the **Font Sizes button menu**.

*Screen: Narrator expands **Font Sizes button menu***

None of the formatting created this way is communicated to those using screen readers. To communicate that something is a heading to these users, you need to use the **Paragraph button menu**. If you expand that menu, you can choose the heading level that matches your document structure--in this case, **Header 2**.

*Screen: Narrator expands the **Paragraph button menu** and chooses **Header 2***

Narrator: It looks the same, but now, behind the scenes, the HTML tag has been updated so that a screen reader user will be told they are encountering a heading.

Screen: Narrator hovers mouse over H2 tag in the breadcrumb beneath the editor.

Narrator: The same principle applies to lists. Under **Web accessibility audits**, I see what appears to be a bulleted list, but when I look more closely, I can see that each list item is really just some text with an asterisk in front of it. If I make it a *true* list by clicking the **Bullet list button** *(Screen: click the **Bullet list button**)*, HTML tags will be added behind the scenes that will make it so that someone using a screen reader will actually be told they are in a list when they encounter it and how many items are in the list.

Let's take a look at the same page with these semantic headings and lists added.

Screen: Narrator navigates to https://canvas.vt.edu/courses/69527/pages/structure-example?module_item_id=844609 in the Safari browser.

Narrator: I'm going to turn on VoiceOver, the screen reader that comes with macOS so I can show you how quickly an experienced screen reader user could navigate to the needed information in a well-structured document. The first thing VoiceOver will do upon loading is read the page title.

Screen: *Narrator turns on VoiceOver (VO) by pressing CMD+F5 and lets it read the page title before silencing it by pressing the Control key.*

VoiceOver: VoiceOver on Safari, Structure (example): REC Examples, window

Narrator: I can then use a keyboard shortcut to bring up a list of headings.

Screen: *Narrator presses CTRL+ALT+U, then uses the right arrow key until the Headings list shows up.*

VoiceOver: Headings menu

Narrator: If I want, I can press the up or down arrow keys to read through the list of headings to get a general sense of the page.

Screen: *Narrator presses down arrow a few times.*

VoiceOver: heading level 1 Structure (example), heading level 2 Assistive technologies used for web browsing, heading level 2 Guidelines on accessible web design

Narrator: Even better, I can start typing to filter the list to just the heading I want. So, for instance, now that I have the Headings list open, I can start to type in the word "audit" and it will only show the headings with that word in it.

Screen: *Narrator types "aud"*

VoiceOver: heading level 1 … heading level 3 Authoring tool … heading level 2 Website accessibility audits 1 item

Narrator: Once I've found the right heading, I jump directly to that part of the page.

Screen: *Narrator presses the **Enter key***

VoiceOver: heading level 2 Website accessibility audits

Narrator: Now I might guess that there was a list under this heading, since the question was to name the three methods of performing an accessibility audit. I can use a twisting gesture on the trackpad of my MacBook Pro to bring up what is called the **Web Rotor** and select **List** from it.

Screen: *Narrator twists on the trackpad until Lists is selected.*

VoiceOver: [switching noise] Lists

Narrator: Then, if I swipe down on my trackpad, it will take me to the next list on the page.

Screen: *Narrator swipes down once*

VoiceOver: list 3 items, You are currently in a list

Narrator: Notice it said there were three items in the list. I bet that's the three items Rob asked me about! OK, I'm gonna turn off VoiceOver now.

Screen: *Narrator turns off VO (CMD+F5)*

VoiceOver: VoiceOver off

Narrator: I just showed you how to add headings and lists in Canvas's rich content editor, but you can do the same thing in Microsoft Word.

Screen: *Narrator switches to the document, accessibility-non-examples.docx, in Microsoft Word*

Narrator: Here we have a document that has fake headings and lists just like what we saw in Canvas and we can solve the problem in a similar way. We just place our cursor on the line we want to serve as a heading *(**Screen:** Narrator places cursor on first line)*, then expand the **Styles pane** *(**Screen:** Narrator expands styles button)* and choose the heading level that is most appropriate for our content, in this case, **Heading 1** *(**Screen:** Narrator chooses Heading 1)*.

Similarly, in our Word document, we see a bullet list that is using asterisks, which we can fix by selecting the list items and clicking the **Bullets button**.

Screen: *Narrator selects the list items and clicks the **Bullets button***.

Narrator: The same principle applies to numbered lists either in Word or Canvas. If you can select the numbers in a numbered list, then it is not semantically a list that will be communicated as such using a screen reader.

Screen: *Narrator selects the number 1 in the numbered list*

Narrator: We can fix that by selecting our numbered list items and clicking the **Numbering button**

Screen: *Narrator selects the numbered list items and clicks the **Numbering button***.

Narrator: I hope this video gave you some sense of the barriers folks might be presented with when encountering a poorly structured document, and that you now have a better sense of what you can do to ensure your learning materials are accessible.