1. Introduction to Focus

Focus is the control of the user from the clipboard and the keyboard. The act of clicking on a text field focuses the field. Tab moves to the next available control.

WCAG 2.0 Checklist 2.1.1 = All page functionality is available using the keyboard, unless the functionality cannot be accomplished in any known way using a keyboard.

1. What is Focus?

Focus determines where keyboard events go in the page.

Tab moves focus forward, shift + tab moves focus backwards, and arrow keys to move focus around within a component.

Tab Order should be logical. Input button and select are implicitly focusable.

(automatically in the tab order and have their own built-in keyboard event handling)

1. Quiz: Experiencing Focus

Space bar to uncheck a box. Form just works with focus behavior.

1. DOM Order Matters

CSS can allow things to appear in the DOM in a different order than the Tab Order will be.

1. Quiz: Fixing DOM Order
2. Using Tabindex

Tabindex attribute = “-1” can be focused programmatically with JS.

Document.querySelector(‘#modal’).focus()

Tabindex = “0” is in the natural focus order and programmable.

Tabindex > 0 will be in the order which they appear. {discouraged}

1. Deciding whats in focus

“Is this something the user is going to interact with?” – before adding a tabindex.

1. Quiz: Which Elements Should Have Focus?
2. Managing Focus

Animated scrolls/changing content without refreshing you should give a tabindex of -1 to the appropriate header, and then call it’s focus method after the user takes action upon the header.

1. Quiz: Manage Focus Yourself
2. Skip Links

Jump around rather than tabbing through everything!

<a href=”#maincontent” class=”skip-link” > Skip to main content</a>

<main id=”maincontent” tabindex=”-1”>



1. Focus in Complex Components

<https://www.w3.org/TR/wai-aria-practices/>

1. Keyboard Design Patterns

Roving focus = set tabindex to -1 on all items except the currently selected is 0.

1. Quiz: Implementing Keyboard Event Listeners
2. Offscreen Content

Focus can move to an offscreenobject.

Document.activeElement in console.

Accessibility Developer Tools

Display: none or visibility:hidden if object is offscreen in CSS.

Display: block or visibility:visible if object appears.

1. Quiz: Implementing Offscreen Content

<https://github.com/udacity/ud891/tree/gh-pages/lesson2-focus>

1. Modals and Keyboard Traps

Tabs can get stuck in keyboard traps. 2.1.2 = NO keyboard traps.

Temporary keyboard trap is desirable with modals.

Modals are the boss battle of accessibility.

1. Lesson 2 Outro

Accessible websites are important for all users!