Keyboard accessibility & :focus

Stuck in a focus trap

Tim Damen

DevCon 2025

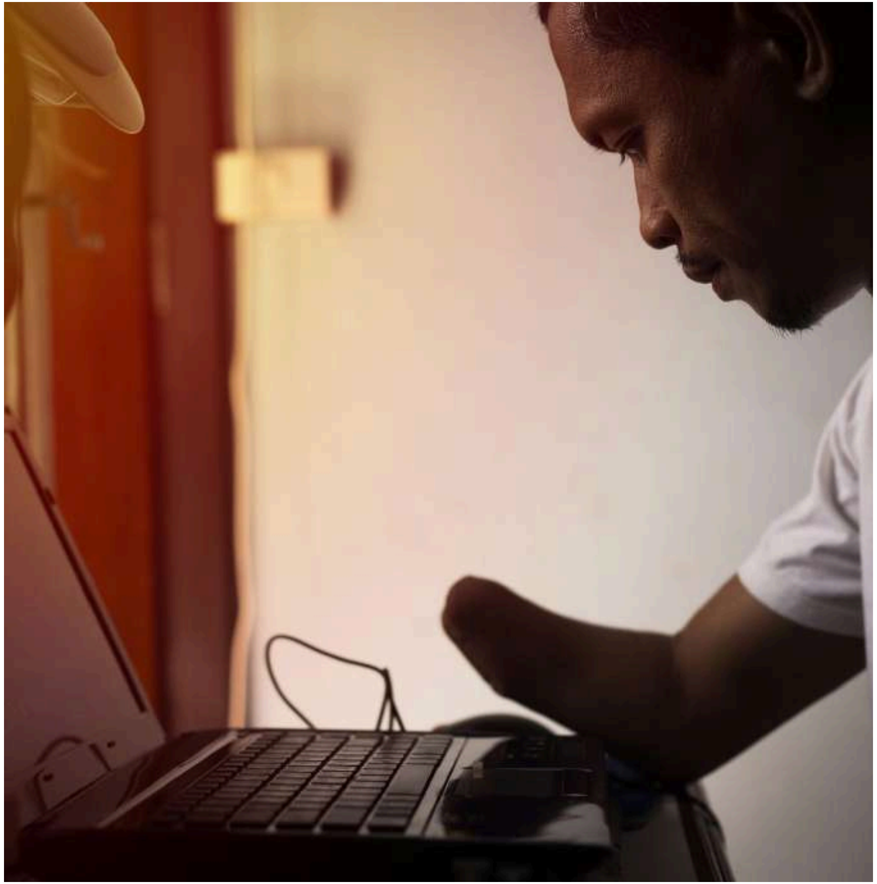
01-04-2025

Keyboard accessibility & :focus

- People using the web with a keyboard
- Shortcuts
- Interactive elements
- Focus rings
- Focus traps
- Tabindex (bonus?)

A lot of people use keyboards to navigate websites and apps.

- Motor disabilities
- Vision issues–including blindness
- With a broken hand
- While quickly filling out a form 🙋‍♂️🙋‍♀️🙋‍♂️



Banks are specifically mentioned in the European Accessibility Act

Due 28 June 2025

Tim Damen

- 🧑‍💻 Front-end Developer
- ♿ Accessibility Lead & Advocate
- 🎙️ Podcaster @ focustrap

Ensure everything that can be done with a mouse, can be done with a keyboard

(also other devices that present as keyboard)

People who don’t use a mouse, use focus styles to see where they are

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry’s standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

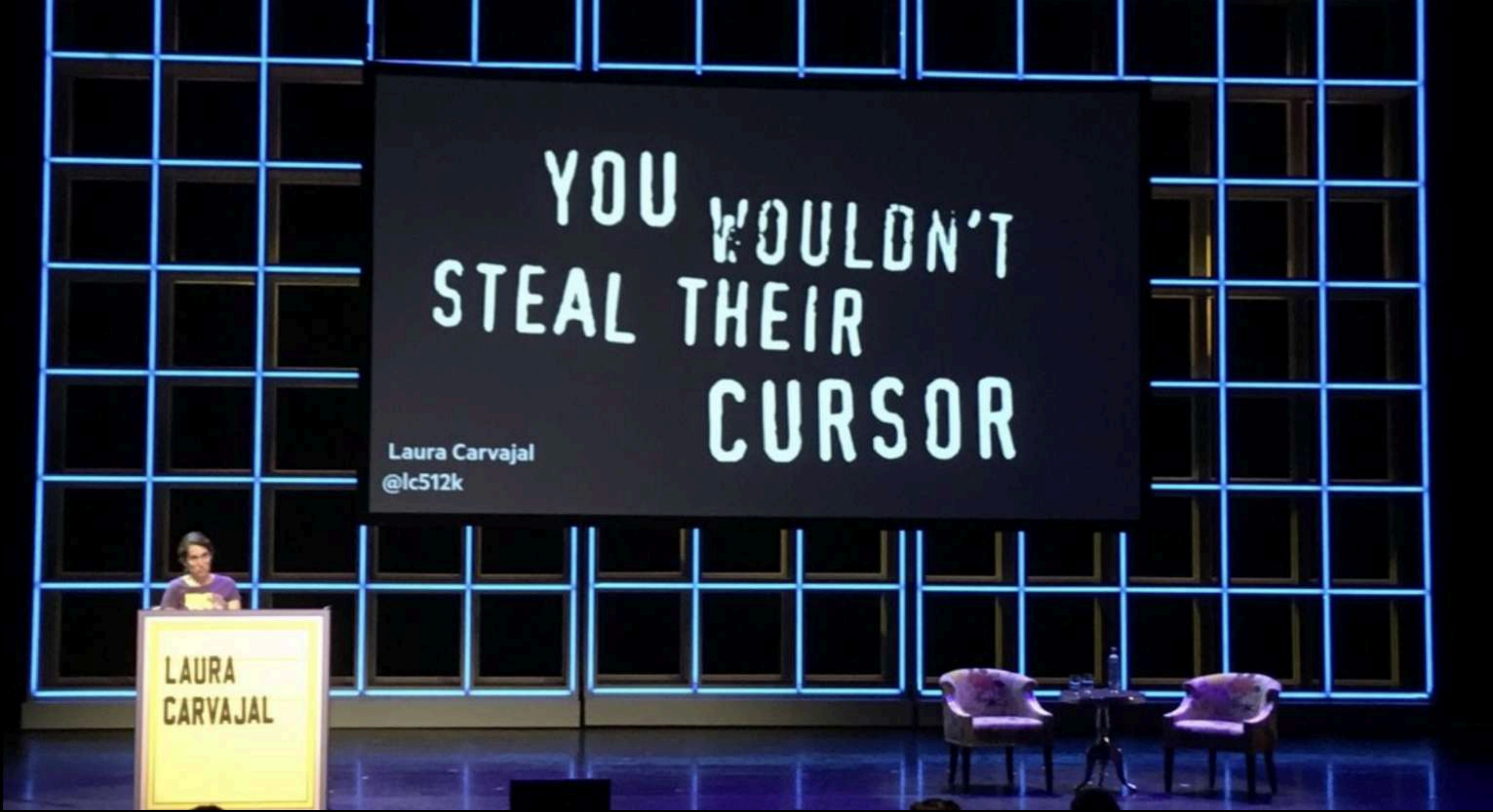
Submit

Ask your self: is the tab order sensible?

(not to many tab stops)

tv
NETFLIX
max
Disney+
prime video
player
CANAL+
arte
movies
iTunes
Filmy
YouTube
vimeo
TVP
VOD

Never remove or hide the focus ring!



Laura Carvajal on stage at Fronteers conference 2018.

Basic keyboard navigation

How to navigate a website with the keyboard, keyboard navigation

Keyboard	Action
`Tab`	move to next interactive element
`Shift` + `Tab`	move to previous interactive element
`Return` / `Enter`	activate elements (links, buttons, etc)
`Spacebar`	activates buttons, checkbox, selectbox (such as to pause/play videos, submit forms, etc)
`Esc`	close opened content (modals, navigation menus, etc) or cancel current action

Good focus styles

- Visible
- 3:1 contrast against adjacent colors
- Consistent

Oreo focus style

This is a dark text on a white background. It contains a link on a white background and serves as an excellent example.

This is a dark text on a gray background. It contains a link on a gray background and serves as an excellent example.

This is a light text on a black background. It contains a link on a black background and serves as an excellent example.

```
:focus {
  box-shadow: 0 0 0 .25rem white;
  outline: .375rem double black;
  border-radius: .125rem;
}
```

https://www.erikkroes.nl/blog/the-universal-focus-state/

Only one thing on a given page can have focus at a time

Interactive elements

`<button>`, `<a href>`, `<input>`, `<details>`, `<select>`

Let’s try!

`<button>` and `<a href>`

button link

Color `<input type="color">`

■

`<radio>` and `<checkbox>`

Do you like coffee?

○ Nope ○ Yes ○ No, I LOVE it!

Please send me

□ Pizza □ Cake □ Ice Cream

Debugging focused element

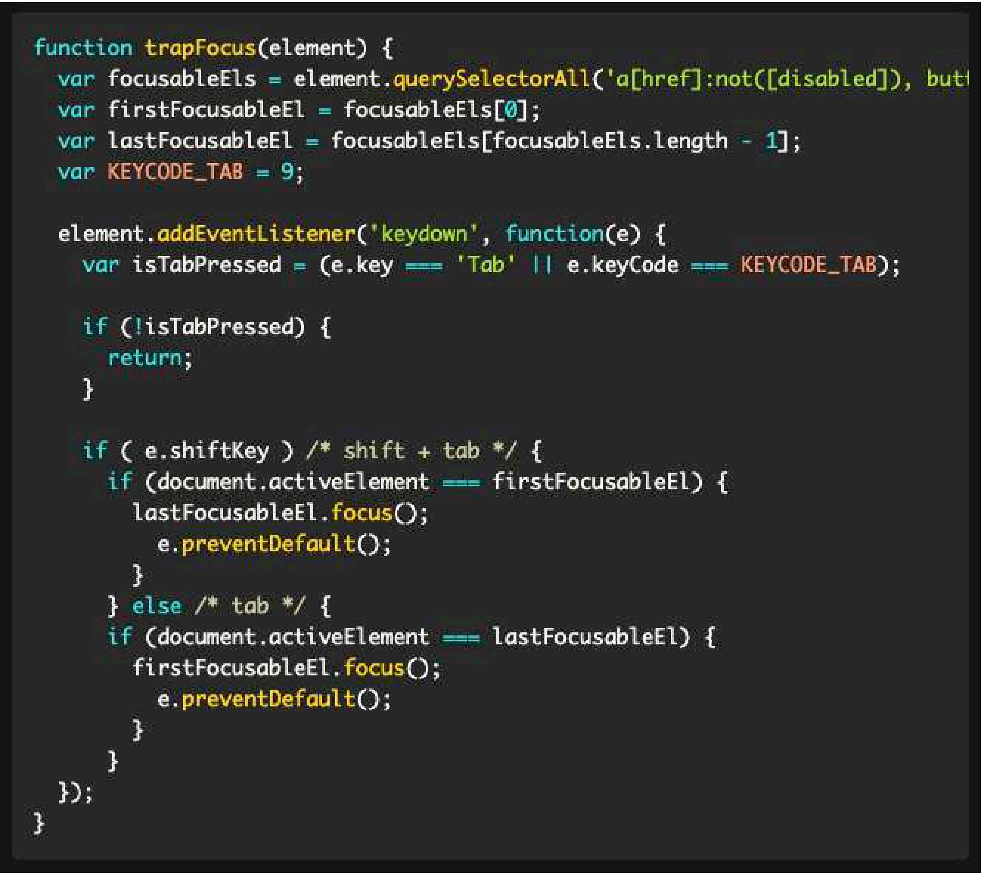
```
document.addEventListener('focus', function() {
  console.log('focused: ', document.activeElement)
}, true);
```

Focus with JavaScript

```
element.focus()
```

When do we need to trap focus?

- Within a modal window
- Within modal navigation
- In a full screen alert



```
function trapFocus(element) {
  var focusableEls = element.querySelectorAll('a[href]:not([disabled]), but
  var firstFocusableEl = focusableEls[0];
  var lastFocusableEl = focusableEls[focusableEls.length - 1];
  var KEYCODE_TAB = 9;

  element.addEventListener('keydown', function(e) {
    var isTabPressed = (e.key === 'Tab' || e.keyCode === KEYCODE_TAB);

    if (!isTabPressed) {
      return;
    }

    if ( e.shiftKey ) /* shift + tab */ {
      if (document.activeElement === firstFocusableEl) {
        lastFocusableEl.focus();
          e.preventDefault();
        }
      } else /* tab */ {
      if (document.activeElement === lastFocusableEl) {
        firstFocusableEl.focus();
          e.preventDefault();
        }
      }
  });
}
```

How to trap focus within an element

```
modal.show();
modal.trapFocus();
```

```
<div
  id="dialog" role="dialog"
  tabindex="-1" hidden
  aria-labelledby="dialog-title"
>
  <form class="dialog-content">
    <h1 id="dialog-title">Name Entry</h1>

    <label for="within-dialog">Name</label>
    <input id="within-dialog">

    <button type="button" id="close-dialog">Close</
    <button type="submit" id="save-dialog">Save</bu
  </form>
</div>
```

https://hidde.blog/using-javascript-to-trap-focus-in-an-element/

How to trap focus within an element (2)

```
// method of the HTMLDialogElement
modal.showModal();
```

Demo `modal.show()` vs `modal.showModal()`

- The modal should trap focus
- First focusable element should be focused
- Close with `Esc`
- When the modal is closed, focus should return to the element that opened the modal

show() ShowModal()

HTML (`tabindex`) allows us to customise focus order

use this feature with caution!

What is the `tabindex` attribute?

The `tabindex` attribute specifies the tab order of an element. It can have three different types of values:

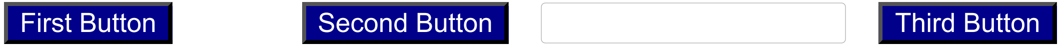
`tabindex="0"`

`tabindex="-1"`

`tabindex="1+"` **(positive values)**

Natural Tab Order

These elements are in their natural tab order (no `tabindex`):



Whenever possible, structure your HTML so that the natural tab order matches the logical reading order. This is the most accessible approach.

Making Non-Interactive Elements Focusable

Regular divs are not focusable by default, but `tabindex="0"` makes them part of the tab sequence:

This div has `tabindex="0"` - It is now keyboard focusable

Another div with `tabindex="0"`

Regular button (naturally focusable)

Programmatic Focus with `tabindex="-1"`

Elements with `tabindex="-1"` can receive focus programmatically but are not in the tab order:

This div has `tabindex="-1"` - Not in tab order

Click to focus the element above

Focus with with javascript

```
element.focus()
```

`tabindex="-1"` is useful for:

- Custom widgets where you manage focus manually
- Off-screen content that needs to be focused programmatically
- Managing focus for elements that should only be focused under specific conditions

Positive Tabindex Values (Not Recommended)

Positive tabindex values create a custom tab order that overrides the document structure:

Button with tabindex="3" (third) Button with tabindex="1" (first)

Button with tabindex="2" (second)

Button with no tabindex (fourth)

❌ Not Recommended

Using positive tabindex values is generally discouraged because:

- It creates a disconnect between visual and keyboard navigation orders
- It's difficult to maintain as your HTML changes
- It can create a confusing experience for keyboard and screen reader users

Summary

- People using the web with a keyboard
- Shortcuts
- Interactive elements
- Focus rings
- Focus traps
- Tabindex

focustrap podcast

I’m hosting a podcast called "focustrap" where we discuss all things digital accessibility.

Check it out! focustrap ❤️

focus trap
Digital accessibility podcast
Hosted by Tim Damen
Episode 4
With Giamir Buoncristiani, Staff Software Engineer at Stack Overflow
stackoverflow

Thank you! 🙏

Let’s stay in touch!

slides: webdevcon25.timdamen.io