

**Mu-An Chiou (@muan)**

**Previously GitHub (accessibility, JS, WC)**

**Now Head of Design Systems,**

**Ministry of Digital Affairs, Taiwan**

**Let developers set**

**Sequential Focus**

**Navigation Starting**

**Point (SENSP)**

[HTML Spec: 6.6.5 Sequential focus navigation](#)

# Removing Headaches from Focus Management

If the element which was the focus start point is removed from the DOM, its parent becomes the focus start point. No more focus whack-a-mole!

– **Rob Dodson** circa 2016

# whatwg/html#5326

Issue created by @Alice

- @domenic
- @bkardell
- @othermaciej
- @AutoSponge
- @robdodson
- @hidde
- @BoCupp-Microsoft
- @Emilio
- @Rich-Harris
- @nimahkh

# Issue supporters

@BoCupp-Microsoft:

I like the direction this issue is heading.

@robdodson:

Internally, browsers have the ability to move the focus start point and it would be useful to expose this to developers.

@hidde:

I see great benefit in having a specific, non-hacky feature built into HTML for this situation.

@Rich-Harris:

+1, In SPAs, fragment navigation would not set SFNSP and we resort to hacks.

@muan (when at GitHub):

I think this convenience is much needed considering that it is very rare for businesses to put development resource into designing a keyboard/screen reader specific experience.

# Why

Focus management is an important piece of web accessibility. Currently hacks exists (moving focus) as an alternative to setting SFNSP, but they are far from ideal.

# Current solution

## Quoting myself:

Using the current pattern, developers have to consider the following steps:

Marking an element to be the start point

1. Setting `tabindex="-1"` on the element
2. Call `focus()`
3. Ensuring focus outline does not apply to this element
4. Install a one time blur handler to remove the `tabindex`

Making this pattern re-usable is very difficult for a large scale website. In these steps, I'd say step 1 alone is already a big task and quite a burden to maintain.

Code, supplied by @Rich-Harris, not dissimilar to what GitHub has.


```
function getFocusStartPoint() {
  return
  window.getSelection()?.focusNode.parentElement;
}


function setFocusStartPoint(element) {
  const tabindex =
  element.getAttribute('tabindex');





  element.setAttribute('tabindex', '-1');
  element.focus();
  element.blur();

  if (tabindex) {
    element.setAttribute('tabindex', tabindex);
  } else {
    element.removeAttribute('tabindex');
  }
}
```

# Real life example

jsdom / .github /  Add file ▾ ⋮

 **domenic** Include Node.js v20 in CI 💬 ✖ e134f68 · 4 months ago 🕒 History

Name	Last commit message	Last commit date
 ..		
 workflows	 Include Node.js v20 in CI	4 months ago
 ISSUE_TEMPLATE.md	Update Node.js minimum and dependencies	5 months ago



# IDL method

```
document.setSequentialFocusStartingPoint(element);
```

# Global attribute

Suggested by Keith Cirkel.

```
<div sfnbsp></div>
```

Bikeshedding welcomed/needed.

# Known issues

1. SFNSP is poorly supported in AT.
2. SFNSP implementation method seems to differ wildly across engines.

## SFNSP is poorly supported in AT

<https://sarahmhigley.com/writing/focus-navigation-start-point/>

However, if browsers save focus-navigation-starts-here-point where the removed element used to be, **keyboard navigation seems to work fine even if focus is not handled.**

...

**Overall, this is a win for accessibility.** For someone relying on keyboard navigation (or any other assistive tech that navigates via focus, like switch devices), getting sent back to the top of the page every single time a developer forgot to handle focus is an enormous pain.

...

**screen readers do not pay attention to the internal browser concept of focus-starts-here-point.** While testing with a keyboard alone may make everything seem fine, trying the same interaction while running a screen reader exposes some rough edges.

- Sarah Higley (Accessibility @ Microsoft)

# Closing

1. Assistive Technology not supporting things in spec is not new.
2. Exposing this method to developers would up the visibility of SFNSP, and make focus management not a hacky script you paste from StackOverflow, which should encourage adoption, and thus help make SFNSP a priority for ATs.
3. The fact that this feature is highly polyfillable (backed by current hack) would actually help with web accessibility in general, given that the hack is not commonly implemented due to the complexity and the lack of understanding in web accessibility.

**Good idea? Bad idea?**  
**(browsers friends wdyt?) →**

**Name suggestions?**

**(async to not take up valuable time?) →**

**Next step**