DOM Order Matters

By Meggin Kearney
Meggin is a Tech Writer

By Dave Gash
Dave is a Tech Writer

By Rob Dodson
Rob is a contributor to Web**Fundamentals**

Working with native elements is a great way to learn about focus behavior because they are automatically inserted into the tab order based on their position in the DOM.

For example, you might have three button elements, one after the other in the DOM. Pressing `Tab` focuses each button in order. Try clicking the code block below to move the focus navigation start point, then press `Tab` to move focus through the buttons.

```
<button>I Should</button>
<button>Be Focused</button>
<button>Last!</button>
```

However, it's important to note that, using CSS, it's possible to have things exist in one order in the DOM but appear in a different order on screen. For example, if you use a CSS property like `float` to move one button to the right, the buttons appear in a different order on screen. But, because their order in the DOM remains the same, so does their tab order. When the user tabs through the page, the buttons gain focus in a non-intuitive order. Try clicking on the code block below to move the focus navigation start point, then press `Tab` to move focus through the buttons.

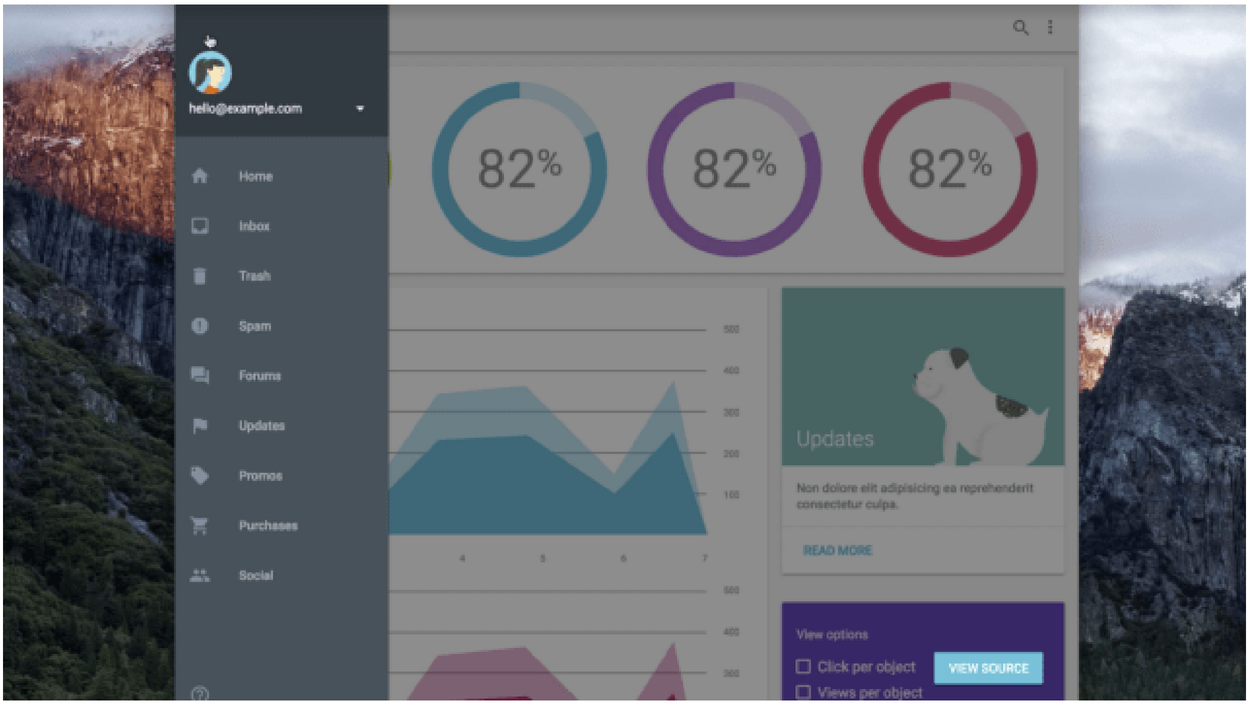
```
<button style="float: right">I Should</button>
<button>Be Focused</button>
<button>Last!</button>
```

Be careful when changing the visual position of elements on screen using CSS. This can cause the tab order to jump around, seemingly at random, confusing users who rely on the keyboard. For this reason, the Web AIM checklist states in section 1.3.2 that the reading and navigation order, as determined by code order, should be logical and intuitive.

As a rule, try tabbing through your pages every so often just to make sure you haven't accidentally messed up the tab order. It's a good habit to adopt, and one that doesn't require much effort.

Offscreen content

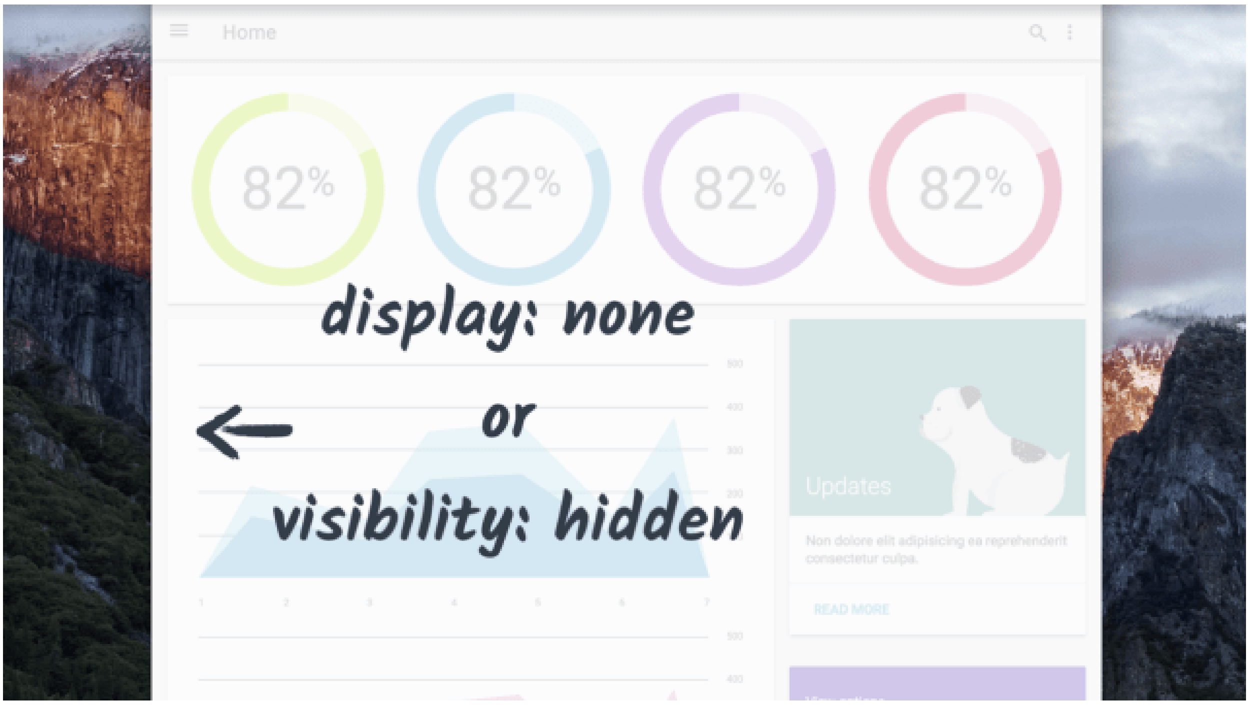
What if you have content that isn't currently displayed, yet still needs to be in the DOM, such as a responsive side-nav? When you have elements like this that receive focus when they're off screen, it can seem as if the focus is disappearing and reappearing as the user tabs through the page — clearly an undesirable effect. Ideally, we should prevent the panel from gaining focus when it's off screen, and only allow it to be focused when the user can interact with it.

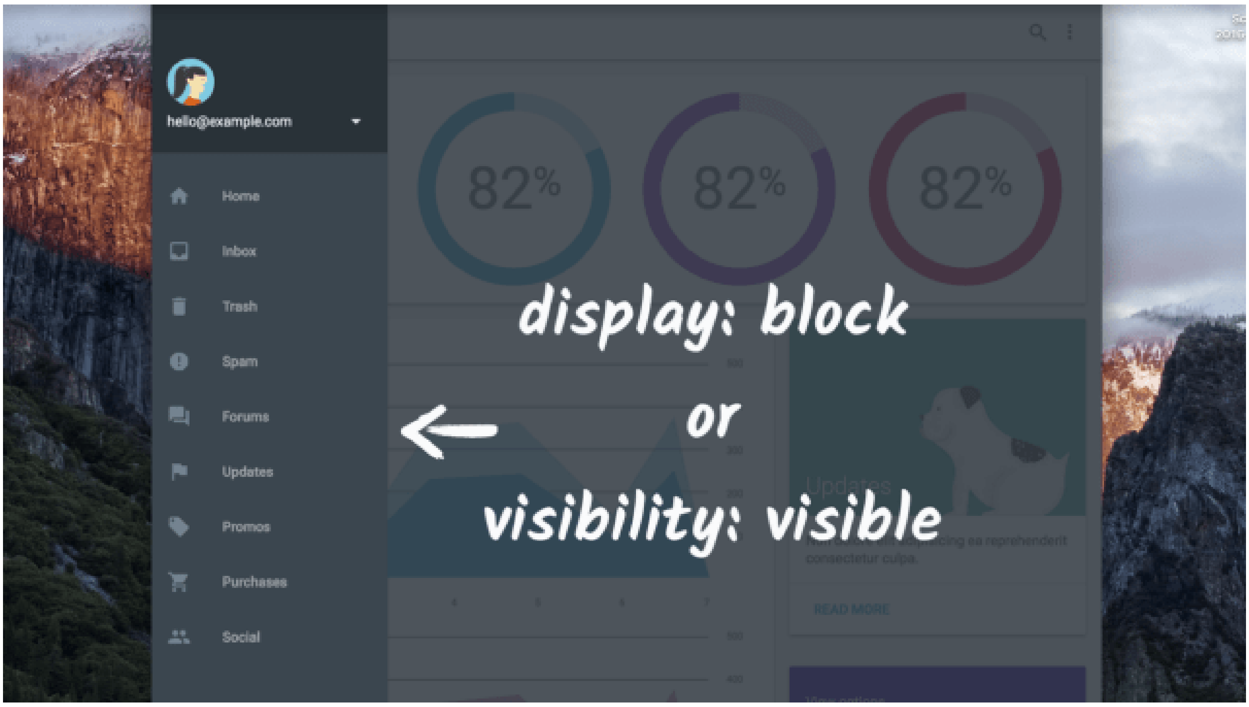


Non dolore elit adipisicing ea reprehenderit consectetur culpa.

Sometimes you need to do a bit of detective work to figure out where focus has gone. You can use `document.activeElement` from the console to figure out which element is currently focused.

Once you know which off screen element is being focused, you can set it to `display: none` or `visibility: hidden`, and then set it back to `display: block` or `visibility: visible` before showing it to the user.





In general, we encourage developers to tab through their sites before each publish to see that the tab order doesn't disappear or jump out of a logical sequence. If it does, you should make sure you are appropriately hiding offscreen content with `display: none` or `visibility: hidden`, or that you rearrange elements' physical positions in the DOM so they are in a logical order.

Last updated July 2, 2018.