Skip Navigation

Many people navigate the web primarily using the keyboard, often with a screen reader. If main content is not the first thing on a page (e.g. if there is a top site banner or navigation menu before the main content starts), these users have to tab through every single link before they can get to the main content. As this is cumbersome and not an ideal user experience, it is recommended that a "skip link" be added at the very top of every page that links to the main content, allowing users to skip right to the most relevant information on the page. It is idiomatic to position this link off of the page so it can only be reached with a keyboard and doesn't interrupt visual flow of the page, and to display the link if it is focused.

Additionally, many users (e.g. low vision users who view pages at a high magnification) have difficulty orienting themselves on a new page if focus is set on too large of an element or an inoperable element, like the wrapper div that Gatsby focuses by default. Sending focus directly to a smaller, interactive control like the skip link is ideal, as laid out in research conducted by @marcysutton and Fable Tech Labs.

This example will show you how to leverage the @reach/skip-nav component to implement a skip link and also how to focus that link on navigation to a new page in your `gatsby-browser.js` file.

Using @reach/skip-nav

Add `SkipNavLink` and `SkipNavContent` to your Layout

```
// src/components/layout.js

import { SkipNavLink, SkipNavContent } from "@reach/skip-nav"
import "@reach/skip-nav/styles.css" //this will show/hide the link on focus

const Layout = ({ children }) => (
  <>
    <SkipNavLink /> //put the link at the very top of your page
    <Navigation />
    <Header />
    <SkipNavContent /> //the link will go to here, so put it right before main content
    <main>{children}</main>
    <Footer />
  </>
)
export default Layout
```

Focus your link on page navigation

Hooking into Gatsby's `onRouteUpdate` API method will allow you to focus automatically on a skip link on page change, putting a user in a more appropriate spot to take action. More information about this method can be found the browser API docs.

```
// gatsby-browser.js

export const onRouteUpdate = ({ location, prevLocation }) => {
  if (prevLocation !== null) {
    const skipLink = document.querySelector("[data-reach-skip-link]") //this is the query selector that comes with the <SkipNavLink> component
    if (skipLink) {
      skipLink.focus()
    }
  }
}
```

🚀 Using this example

🧐 What's inside?

A quick look at the relevant files and directories you'll see in this example:

```
    .
    ├── cypress/
    |   ├── integration/
    |   |   ├── skip-nav.test.js
    ├── src/
    |   ├── components/
    |   |   ├── header.css
    |   |   ├── header.js
    |   |   ├── layout.js
    |   |   └── seo.js
    |   └── pages/
    ├── gatsby-browser.js
```

1. `/src` : This directory will contain all of the code related to what you will see on the front-end of your site. It has pages and components to be used in those pages.

2. `/components` : This directory will contain all of the code related to what you will see on the front-end of your site. It has pages and components to be used in those pages.

    1. `/header.css` : This ensures that our navigation links have focus rings, which are important visual cues when navigating by keyboard.
    2. `/header.js` : This has the navigation links for your pages.
    3. `/layout.js` : This component is used in all of your pages to make sure they all use the same navigation. This is also where you will implement your skip link.
    4. `/seo.js` : This information is necessary to give the site a 100% in Lighthouse for accessibility since screen readers need a title and language.

3. `/pages` : This directory contains pages that will be automatically built and served by Gatsby. This example includes three pages to demonstrate navigation between them and how your skip nav link behaves. All of these pages use the `Layout` component.

4. `gatsby-browser.js` : This file is where you tell the Gatsby to focus the skip navigation when users navigate to a new page.

5. `/cypress` : This directory is where tests and Cypress configuration live. You're going to focus on the test. If you want to learn more about using Cypress, check out the example.

6. **`/integrations/skip-nav.test.js`** : runs two tests to ensure that you have a skip link and that the skip link is focused on page navigation.

Running the example

1. Use the CLI to create run this example

   ```
   gatsby develop
   ```

2. Your site is now running at `http://localhost:8000` !

3. Press `tab` and you'll see the skip link!

4. If you press `enter` and then `tab` again you'll see that you skipped over the rest of the navigation and are in the main content!

Running tests

1. Use the CLI to run Cypress tests

   ```
   npm run test:e2e
   ```

2. this will pop open a window with your tests
3. click "Run all specs" in that window