The Gatsby team is passionate about helping you create websites that work for everyone, with helpful defaults that bake in web accessibility as well as performance optimizations. By making your website accessible to people with disabilities, you can make more inclusive sites that reach and remove barriers for more people on the Internet.

What is web accessibility?

Web accessibility means that websites, tools, and technologies are designed and developed so that people with disabilities can use them. But not only people with permanent disabilities benefit from it. Accessibility also benefits people with temporary disabilities. For example, imagine being in an environment where you cannot listen to audio or you can't use a computer because of a broken arm.

Back in the early days of the Web, Tim Berners-Lee, inventor of the World Wide Web, said:

> *"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect."*

The web of today is an important resource in many aspects of life such as health care, education, or commerce. Accessibility is an important consideration when building for the web.

Accessibility supports social inclusion for everyone, and has a strong business case.

Gatsby makes accessibility easier

While ultimately it's up to you to develop your site with accessibility in mind, Gatsby aims to provide as much out-of-the-box support as possible.

Accessible routing

One of the most common features of every site is navigation. People should be able to navigate across your pages and content in an intuitive and accessible way.

That's why every Gatsby site aims to have an accessible navigation experience by default. Thanks to @reach/router, a routing library for React, Gatsby handles page announcements for screen readers on page change. We're actively making improvements to this experience, and we welcome your feedback.

Since the second major release, your Gatsby sites use `@reach/router` under the hood. While additional accessibility testing is always a good idea, the Gatsby Link Component wraps @reach/router's Link component to improve accessibility without you having to think about it. `@reach/router` also supports scroll restoration.

Gatsby builds HTML pages by default

For websites, rendering static HTML pages means that JavaScript isn't required to access and navigate through content. Gatsby compiles HTML pages by default from React components using Node.js, meaning you don't have to worry about setting up server-rendering yourself to support progressive enhancement. With Gatsby's static support out of the box, you can build dynamic sites that still enable user access without requiring client-side scripting.

Linting with eslint-plugin-jsx-a11y

Gatsby ships with the `eslint-plugin-jsx-a11y` package and warnings for all of its rules enabled by default. eslint-plugin-jsx-a11y is an accessibility linting tool for your code, helping you develop more inclusive Gatsby projects by reducing the time to find accessibility errors. This plugin encourages you to include alternative text for image tags, validates ARIA props, and eliminates redundant role properties, among other things.

For more on supported rules, check out the docs for `eslint-plugin-jsx-a11y`. You can customize those rules in your `.eslintrc`.

```
{
  "extends": ["react-app", "plugin:jsx-a11y/recommended"],
  "plugins": ["jsx-a11y"],
  "rules": {
    "jsx-a11y/rule-name": "warn"
  }
}
```

Note: Including a local `.eslintrc` file will override all of Gatsby's default linting and disable the built-in `eslint-loader`, meaning your tweaked rules won't make it to your browser's developer console or your terminal window but will still be displayed if you have ESLint plugins enabled in your IDE. If you would like to change this behavior and make sure the `eslint-loader` pulls in your customizations, you'll need to enable the loader yourself. One way to do this is by using the Community plugin `gatsby-plugin-eslint`. Additionally, if you would still like to take advantage of some subset of the default ESLint config Gatsby ships with, you'll need to copy them manually to your local `.eslintrc` file.

This is a start to testing for accessibility: further recommendations can be found below.

Tips for improving web accessibility

Accessibility by default is a win for everyone. Here's a starting point for accessibility testing when making a Gatsby site or theme:

- Use your keyboard to tab through the pages. Can you reach and operate every interactive control (links, buttons, form inputs, etc.) and see a focus indicator on the screen?
- Use Lighthouse, axe or Accessibility Insights to find and fix common accessibility issues in development
- Test for adequate color contrast with the accessibility color picker in Chrome Developer Tools
- Create inclusive and accessible forms
- Employ accessible headings, landmarks, and semantic structure
- Include image, video, and audio text alternatives
- Test for screen magnification and zoom
- Ensure accessibility of interactive menus, modals, and custom widgets
- Create safe animations and motion
- Write Cypress accessibility tests for your site or application

Accessibility resources

- React accessibility
- Gatsby's commitment to accessibility
- How to do an accessibility review from Google Web Fundamentals
- A11y Project's Quick Tests
- The importance of manual accessibility testing from Smashing Magazine
- Writing Automated Tests for Accessibility
- Free web accessibility course by Google and Udacity
- WebAIM introduction to web accessibility
- Deque University, with free online accessibility training for people with disabilities
- Web.dev accessibility docs
- All Gatsby accessibility blog posts