Accessibility   UX   CSS   JavaScript   Performance   Design   Figma

John C Barstow *(↦ /author/john-barstow/)* / NOV 23, 2021 /   **18 (↦ #comments-dyslexia-fri**

Adding A Dyslexia-Friendly Mode To A Website

QUICK SUMMARY ↪ *With a little CSS, we can adapt our web designs to be more accommodating for people with dyslexia. In this article, we'll explore those techniques by adding a dyslexia-friendly mode to an existing design.*

Dyslexia is perhaps the most common learning disorder in the world, affecting somewhere between 10–20% of the world's population. It can cause difficulties with reading, writing, and spelling, though the degree of impairment varies widely — some people are barely affected while others require a great deal of extra support.

Existing best practices and guidance, such as the Web Content Accessibility Guidelines (WCAG), give us a solid foundation for inclusive design and already incorporate many details that affect dyslexic readers. For example, WCAG **guidance around line length and spacing** match the recommendations I found doing my research. In fact, some of those resources are linked in the Understanding WCAG 2.1 *(↦ https://www.w3.org/WAI/WCAG21/Understanding/)* document which provides extended commentary on the guidelines.

We can build upon those foundations to offer more focused support for different communities, making it easier to engage with our websites on their own terms. In this article, we'll look at ways to make an existing design dyslexia-friendly.

This article builds on English-language research and can be generalized to cover most European languages that use Latin and Cyrillic scripts. For other languages and scripts, you will find you need to tailor or even ignore these guidelines.

Font Selection # (↦ #Font-Selection)

> *"The font for the body copy should be chosen for its on-screen readability, before any concern for style."*
>
> — *"[How To Apply Macrotypography For A More Readable Web Page](↦ https://www.smashingmagazine.com/2012/05/applying-macrotypography-for-readable-web-page/)*," Nathan Ford*

When I first started researching this topic, I incorrectly believed that I would have to limit my font selection. Luckily, research shows that standard fonts like Helvetica and Times New Roman are just as readable as purpose-built fonts like [Dyslexie](↦ https://www.dyslexiefont.com/) or [Open Dyslexic](↦ https://opendyslexic.org/).

What this means for your font selection is that you merely have to select fonts with [legibility in mind](↦ https://www.smashingmagazine.com/2020/07/css-techniques-legibility/).

All right, problem solved, let's go home!

Well, not really. It turns out there is something special about those purpose-built fonts.

Whitespace # (↦ #Whitespace)

> *"It seems that at least for some people with dyslexia, they are vulnerable to a phenomenon called 'visual crowding' when they read."*
>
> — [Dr Jenny Thomson](↦ https://www.sheffield.ac.uk/health-sciences/people/human-communication-sciences/jenny-thomson)

While study after study shows little benefit from the choice of font, they also consistently show **spacing between letters and words** as the most important factor

in supporting a dyslexic reader. Jon Severs has written [a very good overview of these studies](https://www.tes.com/news/does-comic-sans-really-help-dyslexic-learners) (↪ *https://www.tes.com/news/does-comic-sans-really-help-dyslexic-learners)* with quotes from many of the leading researchers.

The popularity of Comic Sans in the dyslexic community seems to be driven by the wider spacing found in that font, spacing that has been built into additional fonts intended for their community.

As designers, we have the power to extend this spacing to any font, letting us support our readers without a major redesign. While we're at it, we can further **improve things by reducing distractions** and design choices that can produce the visual crowding that affects dyslexic readers.

More after jump! Continue reading below ↓

An Existing Design [# (↪ #An-Existing-Design)](#An-Existing-Design)

The following CodePen example shows a fun little design with semantic and accessible markup that received 100% from a Lighthouse audit. It follows best practices, tries to present a strong visual identity, has good contrast levels, and uses [Overpass](https://overpassfont.org/) (↪ *https://overpassfont.org/)* for headings and body, which provides a unified and legible sans serif family of typefaces:

See the Pen [Dyslexia-unfriendly design](https://codepen.io/smashingmag/pen/jOLXpgg) *(↪ https://codepen.io/smashingmag/pen/jOLXpgg) by* [John C Barstow](https://codepen.io/jbowtie) *(↪ https://codepen.io/jbowtie).*

This will be our starting point, which we will extend to build our dyslexia-friendly version.

Initial Changes [# (↪ #Initial-Changes)](#Initial-Changes)

We want the entire document to work together to support our dyslexic readers, so we will begin by adding a class to the `body` element.

```
<body class="dyslexia-mode">
```

This will allow us to easily toggle our new changes on and off via JavaScript and makes it easy to locate the relevant CSS rules.

The *British Dyslexia Association* published a [style guide (↪ https://www.bdadyslexia.org.uk/advice/employers/creating-a-dyslexia-friendly-workplace/dyslexia-friendly-style-guide)](https://www.bdadyslexia.org.uk/advice/employers/creating-a-dyslexia-friendly-workplace/dyslexia-friendly-style-guide) in 2018 which we can use as a starting point:

> *"Larger inter-letter / character spacing (sometimes called tracking) improves readability, ideally around 35% of the average letter width."*
>
> *"Inter-word spacing should be at least 3.5 times the inter-letter spacing."*

The `ch` unit in CSS is based on the advance of the `0` glyph, but in practice for proportional fonts can often be used as an approximation of the average character width. If you're using a font with a particularly narrow or wide zero, you may find you need to adjust the numbers below.

We're using Overpass in our example, which has a fairly standard zero, so we can express the recommended numbers directly:

```css
.dyslexia-mode {

    letter-spacing: 0.35ch;

    word-spacing: 1.225ch; /* 3.5x letter-spacing */

}
```

Modern browsers default to enabling a font's common ligatures, and older browsers will do so if you use the unofficial `text-rendering: optimizeLegibility` property. For most of us, this improves legibility as it merges close-set characters into a single glyph. For example, 'f' and 'i' are often merged to create 'fi'.

Dyslexic readers, on the other hand, **may struggle to recognize the ligature** as two letters, especially as we have increased the spacing, making ligatures stand out even more than usual. While some browsers may automatically disable ligatures as a result

of the increased letter spacing, for consistent behavior we should explicitly disable them ourselves via CSS:

```css
.dyslexia-mode {

    letter-spacing: 0.35ch;

    word-spacing: 1.225ch; /* 3.5x letter-spacing */

    font-variant-ligatures: none; /* explicitly disable ligatures */

}
```

Line Spacing # (↪ #Line-Spacing)

The WCAG guidelines (↪ https://www.w3.org/TR/WCAG21/#visual-presentation) suggest a minimum line height of 1.5, with a paragraph setting at least 1.5 times larger than the line spacing.

Following this guidance is already quite helpful for your dyslexic readers, but that minimum value is based on the standard word spacing. Since we're increasing the word spacing, we should increase the line height proportionally.

I find a **line-height of 2.0** works quite well. It's a little more than the BDA guidance of 1.5x the word spacing, unitless as suggested by MDN documentation (↪ https://developer.mozilla.org/en-US/docs/Web/CSS/line-height#prefer_unitless_numbers_for_line-height_values), and easy to sync up to a design's vertical rhythm.

To achieve the recommended amount of paragraph spacing, in this example we apply a top margin on our `p` elements. In a larger project, you might want to use Heydon Pickering's famous owl selector (↪ https://alistapart.com/article/axiomatic-css-and-lobotomized-owls/), especially if you have nested content.

Following the WCAG suggestion, that top margin should be **a minimum of 3em** to get the desired paragraph spacing. After feedback from my dyslexic reader, I **increased this to 3.5em** which was more comfortable for them.

> **As with any inclusive design, feedback from real users is critical to ensuring the best results.** (↪
> https://twitter.com/share?
> text=%0a%20As%20with%20any%20inclusive%20design,%20feedback%20from%20real%20users%20is%20critical%20to%20ensuring%20the%20best%20results.%0a&url=https://smashingmagazine.com%2f2021%2f11%2fdyslexia-friendly-mode-website%2f)

While we could apply these settings to our entire page, I prefer to target them to the main content area, especially when modifying an existing design. Site headers, footers, and navigation tend not to have paragraph content and can be particularly sensitive to vertical whitespace changes.

```css
.dyslexia-mode main {
    line-height: 2.0;
}


.dyslexia-mode main p {
    margin-top: 3.5em;
}
```

Other Typographic Changes # (↪ #Other-Typographic-Changes)

At this point, we've made the large-scale changes that will have the biggest impact on a dyslexic reader. Now we can turn our thoughts to the smaller touches that help refine a design.

The extra whitespace we've introduced will make many fonts appear lighter, thinner, or lower contrast, so we can increase the font weight or adjust the color to compensate.

```css
.dyslexia-mode {

  font-weight: 600; /* demi-bold */

}
```

This in turn may make bold (at a font-weight of `700`) harder to distinguish. You could make it a heavier bold by increasing the font-weight or distinguishing it in some other way like changing the size or color. For my design, I chose to leave it at the same weight, but make it darker than the regular text.

```css
.dyslexia-mode strong {

  color: #000;

}
```

Now is a good time to use your developer tools to **check your contrast**. For dyslexic readers, you should aim for a contrast ratio of **at least 4.5:1**, which corresponds to the WCAG 2.1 minimum contrast guidelines (↪ https://www.w3.org/TR/WCAG21/#contrast-minimum).

Why the minimum guidelines? Well, there are two issues to consider. One is that at very high contrast ratios some dyslexic readers will see their text blurring or swirling. This is known as the "blur effect". This is one of the reasons that the BDA style guide we referenced earlier recommends avoiding pure black text or pure white backgrounds.

The second consideration is that many dyslexic readers find larger font sizes more readable (↪ https://pielot.org/2013/06/use-18pt-font-size-for-readers-with-dyslexia/). Research suggests a base size of 18pt, which meets the WCAG definition of large-scale text (↪

[https://www.w3.org/TR/WCAG21/#dfn-large-scale)](https://www.w3.org/TR/WCAG21/#dfn-large-scale) and therefore a contrast ratio of 4.5:1 will still meet the enhanced contrast guidelines.

Which reminds us, we should **bump up that base font size**!

```
.dyslexia-mode {

  font-size: 150%; /* assuming 16px base size, convert to 18pt */

}
```

Responsive designs tend to scale well with browser zoom settings, so a different strategy here could be to leave your font size untouched and suggest that your readers increase the page zoom in their browser.

Following the WCAG guidelines means that our design does not use justified text, so we don't have to make an adjustment. Because **justification can alter the spacing** between letters and words, if you have used it, you should ensure you disable it in a dyslexia-friendly mode.

Reduce Clutter [# (↪ #Reduce-Clutter)](#Reduce-Clutter)

The extra whitespace we've been adding makes it easier to focus on letters and words. That implies that we can be even more helpful by reducing the amount of confusing, cluttered, or potentially distracting things in our design.

Best practices in web design tend to emphasize progressive enhancement and mobile-first design, which **helps keep page weights down and makes web pages resilient**. These practices naturally lead to a minimal default state with fewer decorations and distractions (because these would overwhelm a small screen). We can preserve this minimal state in our dyslexia-friendly mode.

For the background, this means defaulting to a solid color and using the `:not` pseudo-class in our enhancements to avoid applying them to our new mode.

We can use similar constructs to avoid the creation of decorative borders and shadows, leaving only those that are functionally necessary.

```
@media(min-width:700px) { /* only apply on wider screens... */

  body:not(.dyslexia-mode) main { /* ...if not in our friendly mode! */

    background-image:
url(https://res.cloudinary.com/jbowtie/image/upload/v1631662164/exclusive_paper_dyit

  }

}
```

In the existing design, we deliberately make the heading look like an imperfectly applied printed label by rotating it slightly. This is meant to evoke a playful or humanistic touch, and we often see designs adopt little touches like these for similar reasons.

However, this label-like appearance is a prime example of a decorative element that produces **visual crowding**. So even though it works well in a mobile context, we are going to need to remove this touch to provide a better experience for our dyslexic readers.

```
.dyslexia-mode h2 {

  border: none; border-bottom: thin grey solid;  /* just keeping the bottom
border for this element, to retain some separation */

  max-width: 100%; /* standard width */

  transform: none; /* do not rotate */

  background-color: inherit; /* We no longer look like a label, so we don't
require our own background */

  margin-bottom: 1em; padding-left:0; /* some spacing adjustments */

}
```

Zebra striping has long been used when displaying tabular data, but research by Jessica Enders (↪ https://alistapart.com/article/zebrastripingmoredataforthecase/) shows that the benefits

are not necessarily as clear as I thought, and I didn't find any dyslexia-specific research on the subject.

What I did find was a request from my dyslexic reader to implement **zebra striping for tables and lists**! Once again, real user feedback is invaluable.

I chose to restrict this to the main content, to avoid having to revisit the design of the site navigation. We don't actually have any tables in our example, but the CSS changes would be quite similar.

```css
.dyslexia-mode main li:nth-of-type(odd) {

    background-color: palegoldenrod;

}
```

Toggling Our New Mode [# (↪ #Toggling-Our-New-Mode)](#Toggling-Our-New-Mode)

Now that we have a dyslexia-friendly design, we need to decide whether to make it the default, or something that is chosen by the user.

When retrofitting an existing site, as in this example, we'll probably opt for a mode, to reduce the impact of changes on existing users.

In building a new site or refreshing a design, we should consider which changes we can make the default, for the benefit of all users. As with any other design work, you're **balancing the needs of multiple audiences**, branding constraints, and tensions with other design goals such as evoking specific moods or keeping certain information "above the fold".

Switching between modes is accomplished by toggling the class on the body element. Here we do it with a toggle button and some JavaScript, using `localStorage` [(↪ https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage)](https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage) to persist the change across visits and pages. This could be set and stored as part of a user profile.

```javascript
// toggle dyslexia support

const isPressed = window.localStorage.getItem('dyslexic') === 'true';

if(isPressed) {

    document.body.classList.add('dyslexia-mode');

}

// set the button to pressed if appropriate

const toggle = document.getElementById('dyslexia-toggle');

if(isPressed) {

    toggle.setAttribute('aria-pressed', 'true');

}

// toggle dyslexia support

toggle.addEventListener('click', (e) => {

    let pressed = e.target.getAttribute('aria-pressed') === 'true';

    e.target.setAttribute('aria-pressed', String(!pressed));

    document.body.classList.toggle('dyslexia-mode');

    window.localStorage.setItem('dyslexic', String(!pressed));

});
```

📷 *See the Pen [Dyslexia-friendly mode added](https://codepen.io/smashingmag/pen/dyzwqXm) (↪ https://codepen.io/smashingmag/pen/dyzwqXm) by*
[John C Barstow](https://codepen.io/jbowtie) *(↪ https://codepen.io/jbowtie).*

## Conclusion [# (↪ #Conclusion)](#Conclusion)

The separation of content and presentation that CSS gives us always comes in handy when we need to adapt designs to better serve different communities.

Building on the solid foundations of a design that embraces accessibility guidelines, we've learned to extend our design to **improve the experience for dyslexic readers**. There are other audiences that could benefit from this kind of focused design work, and I hope this inspires you to seek them out and share your experience.

This design was tested with a small and possibly unrepresentative sample size. If you or someone you know has dyslexia, your feedback in the comments below about what does or doesn't work would be very welcome and helpful!

ADDITIONAL REFERENCES # (↦ #ADDITIONAL-REFERENCES)

- "[Shorter Lines Facilitate Reading in Those Who Struggle](#) (↦ https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0071161)," Matthew H. Schneps, Jenny M. Thomson, Gerhard Sonnert, Marc Pomplun, Chen Chen, Amanda Heffner-Wong

- "[A Comparative Study Of Dyslexia Style Guides In Improving Readability For People With Dyslexia](#) (↦ https://www.atlantis-press.com/article/125947156.pdf)" (PDF)

- "[Extra-Large Letter Spacing Improves Reading In Dyslexia](#) (↦ https://www.pnas.org/content/109/28/11455)," Marco Zorzi, Chiara Barbiero, Andrea Facoetti, Isabella Lonciari, Marco Carrozzi, Marcella Montico, Laura Bravar, Florence George, Catherine Pech-Georgel, and Johannes C. Ziegler

(vf, yk, il)
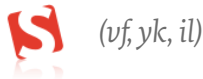
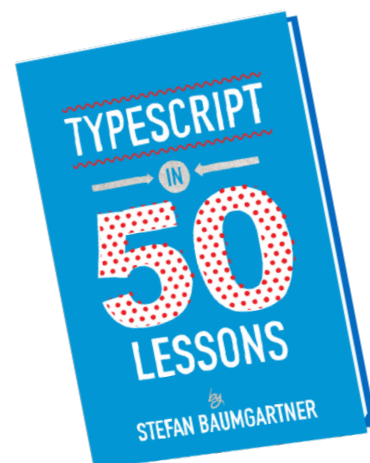*Explore more on*

CSS(↦ /category/css)   **Usability**(↦ /category/usability)

Accessibility(↦ /category/accessibility)   **Techniques**(↦ /category/techniques)

User Experience(↦ /category/user-experience)

[Smashing Newsletter](#)

Tips on front-end & UX, delivered weekly in your inbox. Just the things you can actually use.

[Front-End & UX Workshops, Online](#)

With practical takeaways, live sessions, video recordings and a friendly Q&A.

[TypeScript in 50 Lessons](#)

Everything TypeScript, with code walkthroughs and examples. And other printed books.

With a commitment to quality content for the design community.

Founded by [Vitaly Friedman](#) and [Sven Lennartz](#). 2006–2023.

Smashing is proudly running on [Netlify](#).

Fonts by [Latinotype](#).

Cats can be forgetful, but we are not.
Thanks for being truly smashing — yet again.
www.smashingmagazine.com