

# Key Findings - WDS Comparative Analysis

---

Table of Contents:

[Key Findings - WDS Comparative Analysis](#)

[Aesthetics](#)

[Ease of Use](#)

[Code Snippets](#)

[Navigation](#)

[Help and Documentation](#)

[Getting Started](#)

[Staying up to Date](#)

## Aesthetics

- Using a bold color palette that aligns with branding gives contrast to different sections
  - Dark nav bar next to light colored background along with headers of new sections with a bold background fill color make it clear where a new topic begins
- Sites that are “playful” and interactive are more inviting
  - e.g. Pages where you can test how the sliders work, dropdown menu, checkboxes
  - The examples with recognizable aspects are more interesting
    - e.g. MDL uses Breaking Bad characters for a list example
- It’s good to be thorough, but if the content is broken down into bite-sized chunks, it’s easier to follow
  - e.g. Bootstrap has a single scroll page, but the content is close together and blends
- Minimalist design is good, to an extent - extreme minimalism does not work well with style guides
  - e.g. Bourbon had just the demo and code with no explanation or tips on usage and it left more to be desired

## Recommendations:

- Break the content into smaller “bite-sized” chunks where the extra context is there for those who want it, but those who don’t want it, won’t have to read through a paragraph to find just a single thing
- Use bullet points and highlight key words to help with scanning rather than large paragraphs

- Use colors with high enough contrast for headers of collapsed sections so it's obvious there is more to see there rather than low contrast (light gray on white). This also helps with differentiating where a new section begins
- Use interactive examples where users can test out functionality for themselves such as sliders, checkboxes, dropdown menus, etc.

## Code Snippets

- Using a background color other than white allows these sections of code to pop
  - Using familiar code editor theme colors is a commonality
    - Angular Material does this especially well
- Using syntax highlighting is familiar and easy to follow along with
- Link for copying makes for quicker workflow
  - Most common is to have a button for copying in top right (bottom left works just as well though - which is what MDL does)
  - Copy popup on hover works well
  - Have to be careful not to overlap the copy button with the actual code that is being copied
- Link to CodePen allows for quick editing and testing out the code real time before committing to using it
  - Angular Material and MDL were the two sites to link to CodePen

### Recommendations:

- Add a copy link in the code block areas to allow for quicker workflow
- Option to add the copy link to appear when you mouse over the code area (MDL does this cleanly)
- Be careful not to cover up sections of the code with the copy link
- For more complex code, add a CodePen link where users can test out code real time before committed to implementing it
- Consider using a darker background for the code section (something other than white) so the code area is better differentiated from the rest of the content on the page (MDL and Angular are stellar examples)
- Option to use familiar code editor theme colors such as Sublime Text for the code snippets
- Place code samples close in proximity to the relevant visual example
- Add CSS and JS code samples where applicable

## Navigation

- Left nav is the most common

- Having two forms of navigation is useful
  - e.g. MDL had the main items (About, Getting Started, Components, FAQ, etc) at the top and individual components on the left side
- Having three or more forms of navigation can be confusing
  - e.g. Foundation uses top (main Zurb elements), left (Foundation for sites-specific content) and right nav (what's on the current page)
- Nesting nav items is useful for organization if there are multiple options for a component
  - e.g. Lightning had Utilities (alignment, float, etc) and Touch (forms, menus) nested

### Recommendations:

- Should continue using the sticky left nav for the components and using the top nav for the main, more general items
- Should continue using the nesting in the left nav where necessary for organization and ease of access

## Help and Documentation

- It's good to anticipate that users will have questions come up - FAQ page is helpful for people unfamiliar with the framework
  - Also likely saves time in the long term with answering support emails about common questions
- Search bar is key for quickly finding something within extensive documentation
- Showcase demo of the working site is a good way to build confidence and give potential users an idea what the final result could look like
  - Foundation goes all out with proving their success: showcase, case studies, buzz (essentially customer reviews), brands
  - They have a Foundation by the numbers page that shows how many forks, commits, contributors, etc
- Accessibility is a common topic to include information about in the supplemental docs
- Separating help by Developers and Designers could help with faster searching
  - Designers are invited to explore the guidelines and principles and also to review the components section so they know what can be incorporated into their designs
  - Developers are invited to read about the markup and style guidelines, naming conventions, components and where to get started
- Customization - Bootstrap has dedicated page where you can pick and choose just about everything - CSS and JS components, which JQuery plugins to add, gray and base colors, etc.
  - MDL has an interactive custom theme builder - more tailored toward less technical folks. It's fun to test and can't help but try it out

## Recommendations:

- Add an FAQ page that proactively addresses the top questions that come up (or could potentially come up)
- The FAQs should address topics such as customization, browser support, government-specific information users might need to know, how to report a problem, etc
- Having FAQs could potentially save time in the long term with answering support emails about common questions
- Add a showcase section where users can see what the final product could look like before committing
- Consider adding real users' testimonials in order to build confidence for potential users
- Consider adding customization option with a theme builder for users who want to try out other color schemes

## Getting Started

- All of the well-established sites have an extensive getting started guide - no question about whether or not to have one
- The guide should have details regarding
  - You should use this framework IF...
  - Overview of what it is
    - e.g. "The Salesforce Lightning Design System includes the resources to create user interfaces consistent with the Salesforce Lightning principles, design language, and best practices. Rather than focusing on pixels, developers can focus on application logic, while designers can focus on user experience, interactions, and flows."
    - e.g. "The Lightning Design System reflects the patterns and components that underpin the Salesforce product. These patterns and components provide a unified language and consistent look and feel when designing apps and products within the Salesforce ecosystem."
  - Installation - should give options
    - Through command line, manual set-up, etc.
  - Downloads
  - Directory structure
  - Supported browsers
    - Helpful to talk about what's not supported before someone new starts using it in case they need it for a specific browser
    - A chart with obvious red X and green checkmarks for what browsers and devices are supported is a quick way for someone to scan the support
  - Links to other necessary documentation
  - Details for devs/designers - separated out

- Separating out by developer and designer is a simple way for people to read what's relevant to them
- The path to "Getting Started" is always obvious
  - e.g. first link in nav at top or on left
- "Getting Started" seems to be the standard page name for this content
- "How do I start?"
  - Links to CodePen, tutorials and GitHub are good ways for new people to quickly find the essentials and get up and running quickly
- Tips for where to go if you have problems or questions - GitHub or the forum
  - e.g. Angular Material has this at the end of their Getting Started Guide
- Basic HTML starter templates are an easy way to get started quickly
  - Angular Material has a blank shell Material application
  - MDL has 6 starter templates that come with placeholder images and color examples
  - Foundation has 8 HTML only templates and they require you to give your email address before downloading
- This page is a good place to add link to sign up for mailing list
- Good page to add licensing info, legal stuff
  - e.g. Bootstrap

### Recommendations:

- Add starter templates that includes HTML, CSS and JS with examples showing what the layout with color could look like
- Consider adding a "How do I start?" section for beginners with links to CodePen, tutorials and a starter project
- Add the supported browsers/devices for the framework. Using a table is a clear way to display what's supported and what's not so users can scan it quickly

### Staying up to Date

- Common to show the latest release/version number at the top of the page
  - Link to older versions in case someone is looking for a previous rev
- It's good to do a big callout at the top for major new releases
  - e.g. Bootstrap has a banner that calls out Bootstrap 4 is coming and they let you know ahead of time so you can plan for it
    - Banner links to blog post
    - They call out that it is not backwards compatible
- Bootstrap has dedicated page for keeping track of new versions
  - What you need to do if you update
  - Is it backwards compatible?
  - What's new?
  - Other common questions?

- Common to have a link in the footer for joining a mailing list to stay up to date
- Links to Twitter are popular ways for devs to keep track of updates

### **Recommendations:**

- Show the latest version number on the top of the page in the nav or on bottom of the page and link to where previous revisions live in GitHub
- Encourage users to sign up for mailing lists to receive updates and use Twitter for sharing news relating updates
- From the nav, link to relevant blog posts about updates to the WDS informing users about what's new, how they get it, and any potential issues that they might have if they update