

REACT Component Documentation Guidelines

Versions

Version	Date	Who
1.0	February 2023	Charlene Jaszewski

Who is this style guide for?

This style guide is for U.S. Bank content strategists and writers who are creating/editing REACT Shield 2.0 component documentation.

What’s in this style guide?

This style guide will detail our standards for structure, formatting, content, and language for REACT Shield 2.0 component documentation.

Other references

This style guide supplements the [Shield Writing Guidelines](#), which provides general U.S. Bank writing, terminology, and usage guidance.

Component documentation process in brief

Our component documentation process consists of four phases:



1. Content creation & review (Word template)
2. Illustration co-creation (Figma template)
3. Page staging & review (Zeroheight template)
4. Prep for launch

Questions?

If you have any questions or suggestions, contact [people].

Table of contents

- GENERAL 4**
 - Pre-addressing questions on missing or unclear info 4
 - REACT shield component style guides in Zeroheight..... 4
 - Headings and hierarchies..... 4
 - Text/images in running text (not do’s and don’ts) 4
 - Link anchor text 4
 - Formatting 5
 - Table formatting 5
 - Talking about components 6
 - Notes/Callouts 7
- OVERVIEW TAB 8**
 - H1: Component name..... 8
 - Component names and sidebar navigation 8
 - Component description 9
 - Tabs..... 9
 - Shield features and status..... 10
 - H2: Live demo 11
 - H2: Usage 12
 - H3: Use a [component] 12
 - H3: Don’t use a [component] 15
 - We recommend* table 15
 - Don’t use [component]: don’t rules 17
 - Related components..... 18
- GUIDELINES TAB.....19**
 - H2: Anatomy 19
 - CS notes 20
 - H2: Customization options..... 20
 - Example: Borders 22
 - H2: Behaviors 23
 - H3: [Custom Behavior heading, etc.] 23
 - H3: Focus order..... 23
 - H3: Keyboard interaction..... 23
 - H3: Screen reader 23
 - H3: Responsive 24
 - H3: Validation and errors..... 24
 - H2: Content..... 25
 - H2: Best practices/More best practices 26
- CODE TAB.....28**
 - Technical overview 28
 - Accessibility..... 28
 - Anatomy..... 29
 - Adding emphasis..... 30

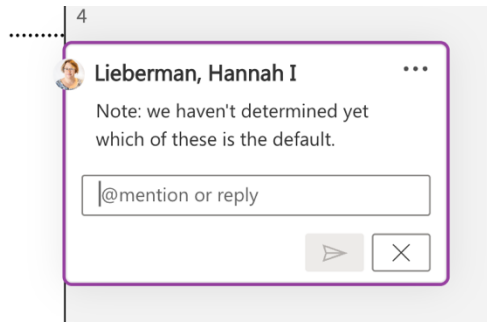
REACT Component Documentation Guidelines

Exception: Quotes are used in rules text in Zeroheight.....	30
Terminology (words not addressed in U.S. Bank Brand Guide)	31
How to say it (preferred words/phrases).....	31
Current.....	31
Preferred.....	31
GENERAL WRITING TIPS	33
List order/Image order sync.....	33
Use contractions	33
Don't need to write "for the user" or "to the user"	34
To save space, use numerals instead of spelling out numbers less than 10.....	34
Don't use indefinite language UNLESS you provide an explanation	34
Anchor link text when linking to component.....	34
Example language, short.....	34
Example language, longer.....	35
When referring to other sections	36
States	36
Ensure prop name and prop description are the same	36
What if part of the functionality/option isn't available yet?	36
When should you use a table for information?	37
Options tables.....	38
When to use bullets vs. paragraph text.....	39
Bullets are great for chunking out information to make it more scannable	39
Consistency	40
Give parallel examples	40
REFERENCE.....	41
Component descriptions (as of Jan 2023).....	41
Language for referring to components/user experience.....	42
Overview > Do's Example Language	43
Anatomy item descriptions.....	44
Do's and Don'ts vs. Best Practices Language	44

GENERAL

Pre-addressing questions on missing or unclear info

If you're sending a doc to peer review that has missing or not-yet-decided information, leave a note on that section for reviewers, like this:



REACT shield component style guides in Zeroheight

Though we've detailed all the standards we've set for REACT documentation in this guide, it's helpful to [view the components in Zeroheight](#) yourself.

Headings and hierarchies

The content hierarchy (order and H2 & H3 heading text) repeats across all component pages, which creates a consistent experience for our users.

- Don't change or customize content order or heading text.
- When necessary, it's okay to remove a section due to lack of content (check with the lead content strategist first).
- All headings/sections are required unless noted as optional/recommended.

Text/images in running text (not do's and don'ts)

When an illustration is provided along with text (mostly on *Guidelines* tab):

- Text explaining the image goes BEFORE the image
- Don't include image captions for full-size images in running text

Link anchor text

Whenever possible, have the anchor text match the name of the destination of the link, whether that's a Yammer, a Figma library, a Gitlab repo, or other destination. Use the specific name on first mention for sure. If, however, you are linking to the same thing over and over on a page, there could be a case for using more generic anchor text on latter mentions.

No	Yes
For more guidance, ask a question in our Yammer .	For more guidance, ask a question in the Shield DS-Yammer .
Add the code to our Gitlab repo .	Add the code to the Shield Components repo .

Formatting

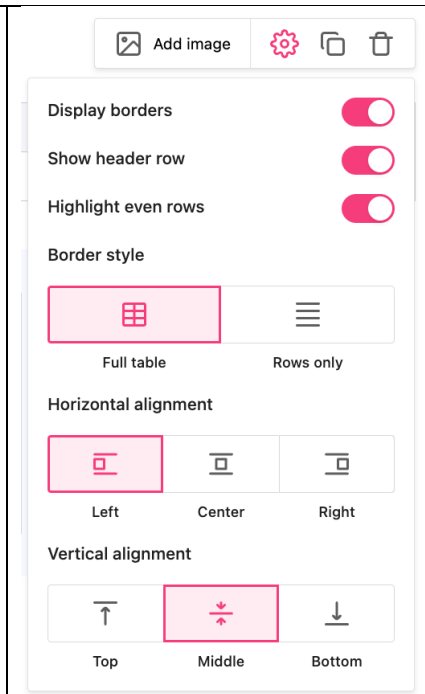
Table formatting

All tables should have the following settings in **Table settings**:

- Display borders: on
- Show header row: on
- Highlight even rows: on
- Border style: Full table
- Horizontal alignment: left
- Vertical alignment: middle

Header row formatting: **BOLD**.

Note: Ideally the header row would be automatically formatted in **BOLD H4** when the user selects *Show header row*. Accessibility needs table/column header rows to be formatted in heading styles, not just bold. We have submitted this feature request to ZH.



Key	Function
space or enter	Toggles the accordion panel open and closed
tab	Moves focus from one accordion panel to the next and follows DOM order for interactive elements within panels
arrows	Moves focus from one accordion panel to the next

Talking about components

Don't use the word "component" when discussing a component

No	Yes
Use a progress indicator component for multi-step flows.	Use a progress indicator for multi-step flows.

Don't capitalize a component name in running text

We don't capitalize component names.

Exception: A component name in the H1 on the ZeroHeight page has the first word capitalized.

No	Yes
Use a Progress Indicator for multi-step flows.	Use a progress indicator for multi-step flows.
	<p>Components</p> <p>Progress indicator Ⓢ Status tag</p> <p>A progress indicator orients users within a specific task flow.</p>

Don't use platform name within component's documentation

If you're documenting React components, you don't need to use "React" before the component name.

No	Yes
The React dropdown can be configured to have either heavy or minimal emphasis.	The dropdown can be configured to have either heavy or minimal emphasis.

Refer to component in the singular in component name and description (top of page)

- Refer to component in the singular in the component name (h1 title) and component description, except for components that are by nature plural.
- You can be flexible (singular or plural) in following text.

No	Yes
Accordions expand... Buttons let...	An accordion expands... A button lets users...
A chip is... A breadcrumb is... A tab organizes...	Removable chips are... Breadcrumbs are... Tabs organize...

A few components are kind of weird sounding. For example, select. It sounds kind of weird to say "use a select." Though we normally don't use the word "component" with a component, in this case you could ("use the select *component*...").

Refer to component with its proper component name in anchor text

If referring to a **checkbox**, don't call it "single checkbox" in other mentions. (Also, a checkbox IS single, a checkbox group is multiples.)

No	Yes
When using filtering chips...	When using removable chips...
When using chips that filter...	When using removable chips...

Notes/Callouts

You can format notes in two ways:

- **Asides:** for notes that are more asides, add a paragraph introduced by the word "Note."
- Don't italicize the word "note."

Note: More guidance on Design Tokens is coming in 2023.

- **Important:** If the Note is important (and not just an aside), use the **Callout** element with the blue color (color 2).

Use U.S. Bank themes.

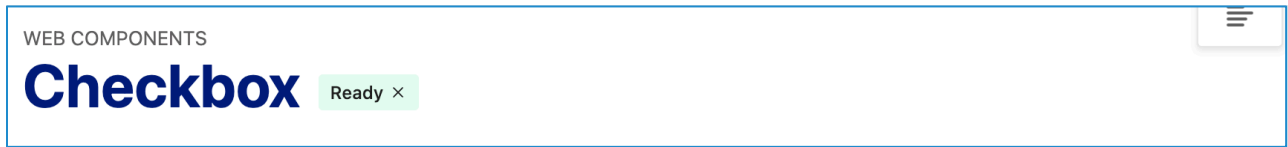
The Shield React 2.0 library supports multiple themes to help customize the look and feel of your experience. Themes are driven by tokens that support color, typography and other miscellaneous values.

If your team has moved to the new U.S. Bank brand, use the *USB Light* theme. If not, use the *Consumer* theme.

If your business line needs a new theme, reach out to the Shield team in the [Shield DS-UX Yammer](#).

OVERVIEW TAB

H1: Component name



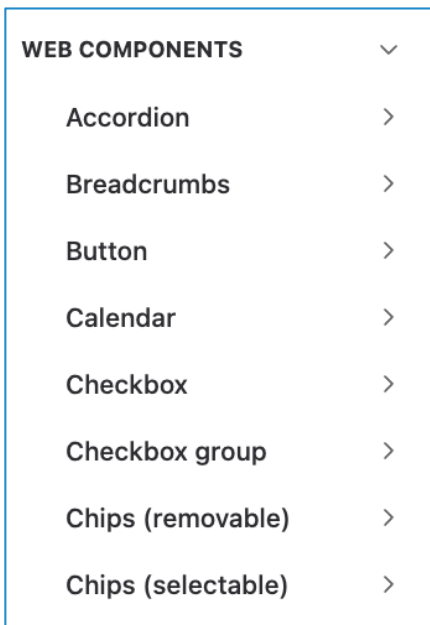
Content guidelines

- Only capitalize the first word (Checkbox group, Date picker). *Note:* The capitalization is only for use for this heading. In running text, do not capitalize component names.
- Don't add the word 'component' at the end of the name (e.g., "Button" and not "Button component").
- Make component name singular (unless it's a component that has multiples, like removable chips).

Component names and sidebar navigation

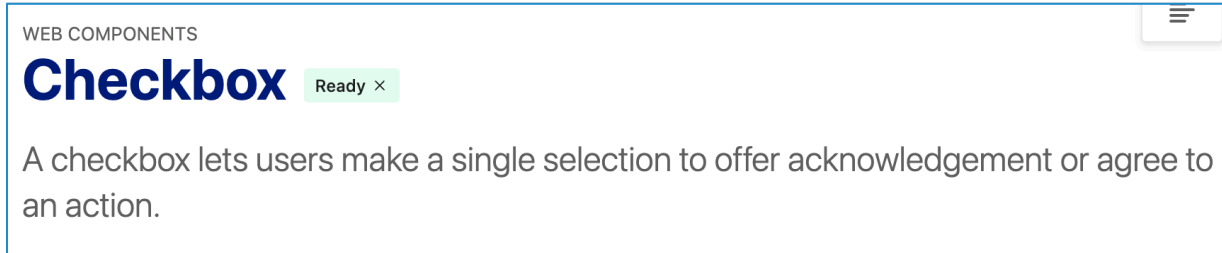
Currently we list components in the sidebar in alphabetical order. (We're considering possibly grouping them by function (i.e., input, messaging, etc.)).

However, related components might not land next to each other in alphabetical order. A current example is Removable chips and Selectable chips. For this situation, we name the component with the common part first, followed by the specifier in parentheses.



Component description

A component description is a brief and broad summary of the component’s function and use case.



Strategy notes

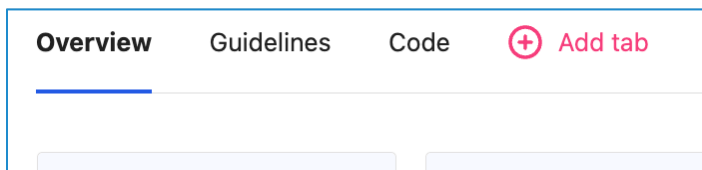
- Write from the perspective of how a component enables the end user (e.g., A checkbox lets a user to make a selection, either from a list of options or a single yes/no question or statement.).
- Keep to 1-2 sentences. One-sentence description of what the component does/is.
- Don’t add commentary or component benefits. That information belongs on the *Guidelines* tab.

Content guidelines

- Include the name of the component in the description (e.g., “A checkbox lets users...”)
- Use the standardized language you wrote for the description elsewhere when referring to the component, for example, when referring to the component from another component.
- Make component name in description singular (unless it’s a component that has multiples, like removable chips).
- Review current [Component Descriptions](#) for reference.

Tabs

These tabs (Overview, Guidelines, Code) repeat across all component pages, creating a consistent experience for our users across the library.



- Overview: Summarizes what the component is, when to use/when to consider something else and available resources (e.g., Auditor test results, Figma UI kit).
- Guidelines: Breaks down component customization options, behavior and proper usage.
- Code: Includes component ReadMe (copied from GitLab package) and developer-specific A11Y guidance.

Content guidelines

- Don’t change or customize tab names.
- Don’t change the order of tabs.
- If there’s a need to remove or add a tab to this template, discuss with the lead content strategist.

Shield features and status

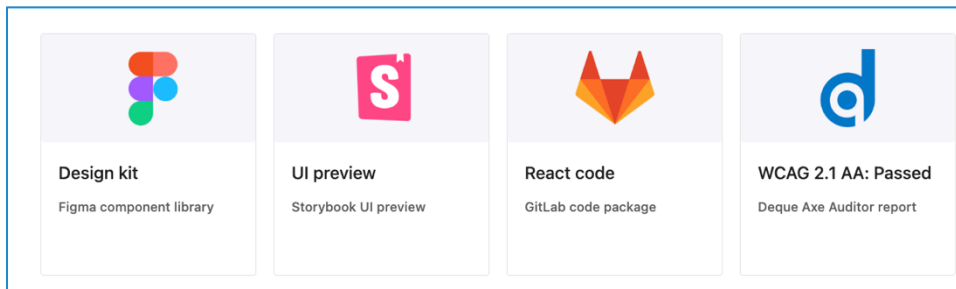
This section summarizes component features & the status of those features. Each tile/shortcut is repeated across all component pages.

Content guidelines

- Don't change order of the tiles.
- If resource isn't yet available, don't include a tile for it.

The tiles/links must be customized to reflect the component:

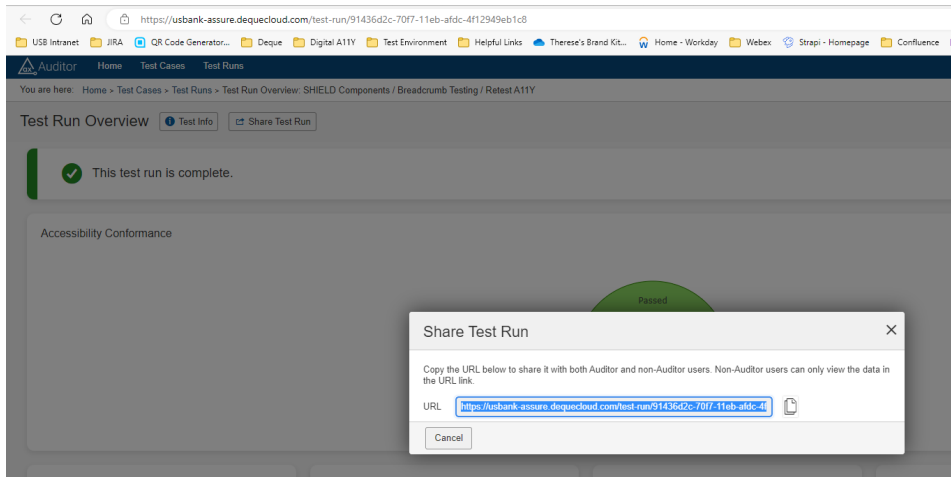
- Design kit/Figma component library: Link to component Figma library
- UI preview/Storybook UI preview: Link to component in Storybook
- React code/GitLab code package: Link to component GitLab package & Storybook instance
- WCAG 2.1 AA [status]: Link to Deque Auditor report, specifically the



Adding the Deque Axe Auditor report link

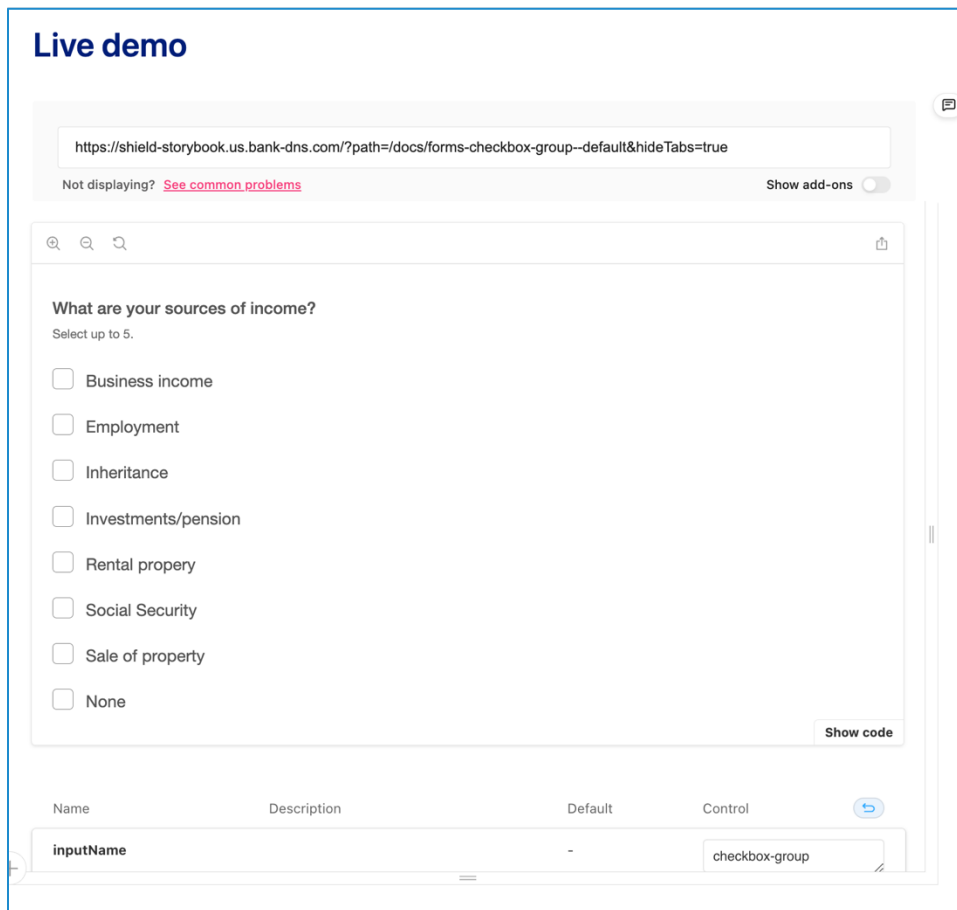
For components that have passed WCAG criteria, we need to link to the *Test Run Overview* page in Axe Auditor with a view-only link, so people without Deque access can view it. To get the link:

1. Log into Deque.
2. On the *Test Run Overview* page for the component, click the *Share Test Run* button.
3. Copy the link, then paste into the shortcut tile.



H2: Live demo

Live demo uses a Storybook URL to show the component in action.



Content guidelines

- Use an actual example, not generic text in the live demo.
- Example may need to be reviewed by A11Y team.

H2: Usage

This is a new H2 heading above the **use/don't use** sections.

- The **Usage** section helps designers determine if the component is the right solution for their situation. The **Use [component]** content summarizes the most common use cases for this component.
- The **Don't use [component]** content lists situations where designers might be tempted to use the component and refers them to a component better suited to that situation.

H3: Use a [component]

Use a checkbox

- To offer acknowledgment (like agreeing to terms and conditions)
- To agree to an action (such as opting in to receive emails)
- To supplement another input (like asking to remember a username when logging in to an account)

I've read the documents, and I agree to the terms and conditions.

Agreeing to terms and conditions

I would like to receive news and product announcement email updates.

Opting in to receive email updates

✓ Do

Use a checkbox to offer acknowledgment.

✓ Do

Use a checkbox to agree to an action.

Strategy notes

- You can use a combination of bullets and Do's "rules" to describe common use cases for the component.
- Since we don't have an actual "overview" section for each component, this section can contain "overview" content, as well as "where to use" advice (like "use in a form").
- This is high-level information and shouldn't contain implementation details (e.g., "Place breadcrumbs near the top of the page..."). That type of guidance goes in the *Guidelines* tab.
- Ideally all the "Do's" would be in the form of DO Rules (images + text) vs. bullets. But if you don't have the bandwidth to generate illustrations, bullets to communicate "do's" are acceptable.
- Try not to duplicate information between the bullets and the Do's rules.

Do use [component]: bullets

- For the H3 heading, refer to component in the singular, unless the component is used in multiples. So, "Use a checkbox," and "Use removable chips."
- The bullets should complete the sentence "Use [component]..." e.g. "To send users somewhere new."
- Exception: if there's only one bullet of content, make the phrase into a full sentence, and remove the bullet.
- No periods needed

- It's okay to have example language in the bullets (set off with parentheses).
 - "To reveal critical information or consequences of an action (like exiting without saving work)"
- If you use examples in multiple bullets, mix up the language ("like" or "such as" are preferred, and definitely don't use "e.g.")

Usage

Use an accordion

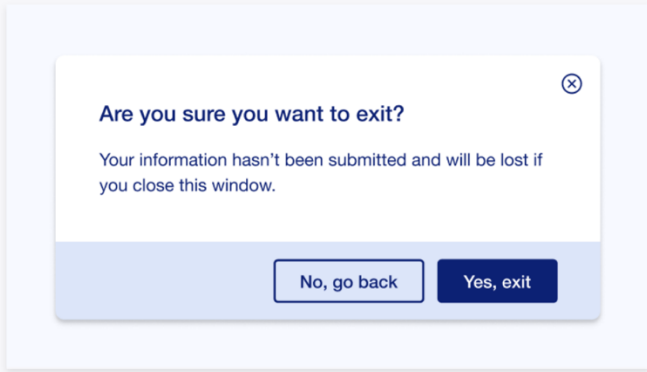
- On text-heavy pages that would otherwise require too much scrolling
- To let users control how much content they want to see on text-heavy screens (like FAQs)
- To let users compare content

Use [component]: Do's (optional but recommended)

Include do's rules only if showing an example better clarifies the guidance.

Each "do" rule (ZeroHeight rule component) needs:

- An illustration
- General guideline (rules text)
- Example text (image caption)



Confirming exit (and warning of consequences)

Example text

✓ Do

Use modals to confirm user actions that may have critical consequences.

General guideline

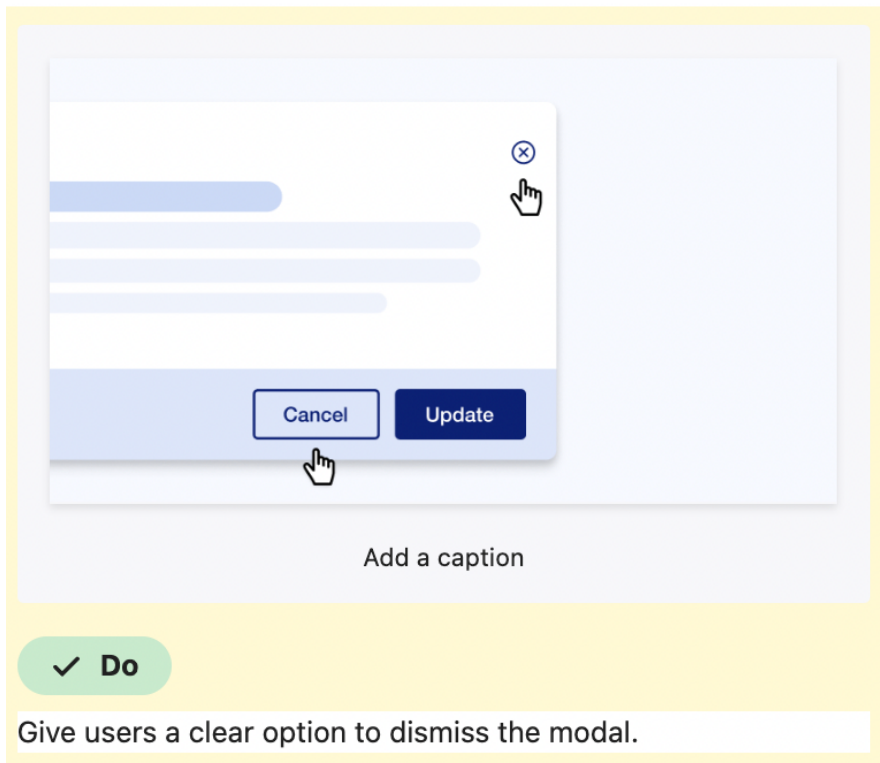
Content/formatting guidelines

- Write rules text/guidelines text that states the general usage guideline.
 - Use the format “use [component]...”
 - Keep it short, just stating the rule. Don’t use examples. If you do have some “extra” explanation, that should go in the bullets above the Do’s/Don’ts.
- Write example text that describes what’s happening in the example image.
 - Use a gerund verb form for the verb in example text (displaying, confirming, etc.)
 - No periods needed on the example text (it’s a phrase)

Don’ts for the do’s

- Don’t make a DO RULE of something that the system does by default.

The following DO makes it sound like it’s the designer’s responsibility to add a way to dismiss the modal. But a dismiss close button is included with the modal by default.



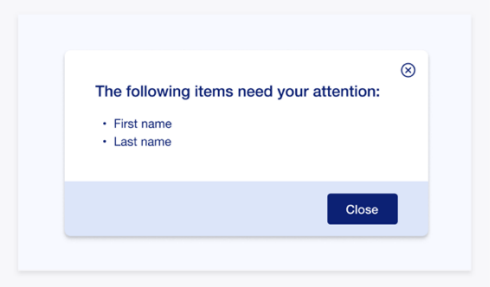
- Don’t make a DO rule that doesn’t need an image or example for understanding.

H3: Don't use a [component]

The goal of this section is to help users determine whether the component they're reading about is the right component, and if not, possibly refer them to a component or user experience better suited to their situation.

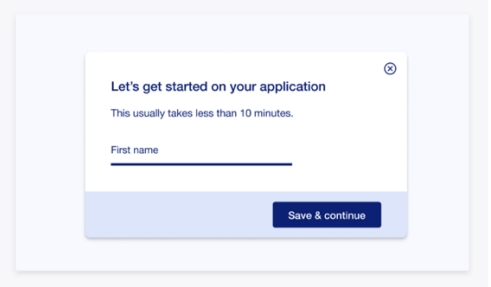
Don't use modals

In this situation	We recommend
Supplemental content that only appears when activated by user	Toggletip
Confirming success of a specific task or communicating issues or errors	Notification
Confirming success after completion of a complex task or multi-step flow	Confirmation page
Progressing through multi-step flows	Progress indicator



✗ Don't

Don't use modals to address inline form errors.



✗ Don't

Don't use modals to guide users through a multi-step process or flow.

We recommend table

The table summarizes situations in which the component wouldn't be the right choice and suggests a more appropriate component or user experience.

Don't use text area

In this situation...	We recommend...
Short, single-line entries	Text input
One selection allowed from a list	Select , combobox or radio group
Multiple selections allowed from a short list	Checkbox group

Content guidelines

In this situation column	We recommend column
<ul style="list-style-type: none"> Use the situation language from the Language to refer to components table in the appendix. If you need to add language for a new component/situation: <ul style="list-style-type: none"> Keep it brief (one short line). Start with the gerund verb form if possible (entering, taking, confirming, etc.). 	<ul style="list-style-type: none"> If linking to another component, list that component in the <i>We recommend</i> column and link to it on the name. Don't use a verb ("Use Toggletip"), just list the referred component (or other solution). If you're referring to guidance other than a component, link to that if possible.

- If there's a reason to not use the component, but it doesn't refer to another component or user experience, don't put the guidance in the table, put the guidance in a bullet before the *We recommend* table (this should be rare).

Don't use a tooltip

- For content that must be always visible (such as footer information)
- For communicating status information within a form (like successes or errors)

In this situation	We recommend
Confirming success of a specific task or communicating issues or errors	Notification
Long or complex content	Another disclosure pattern

- If two situations refer to the same component, make two separate rows:

Don't use notifications

In this situation...	We recommend...
Urgent decisions that require user confirmation	Modal
Alerting users to system or connectivity issues	Separate page
Confirming success after completion of a complex task or multi-step flow	Separate page

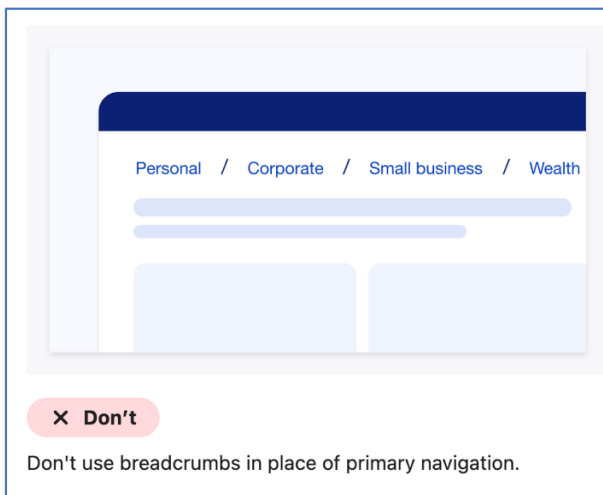
Don't use [component]: don't rules

(Optional but recommended)

Again, only include a “Don't” rule if it provides additional clarity to the situations detailed above. Each “don't” rule needs:

- An illustration
- General guideline (rules text)

Note: We decided not to use example text in an image caption as it sounded like it was endorsing what was happening in the image.



Strategy notes

- Only use a Do/Don't/Caution if using an image with an example will help clarify understanding.
- Ideally use a don't/do pair. Start with a don't and include a do explaining/showing the correct alternative.

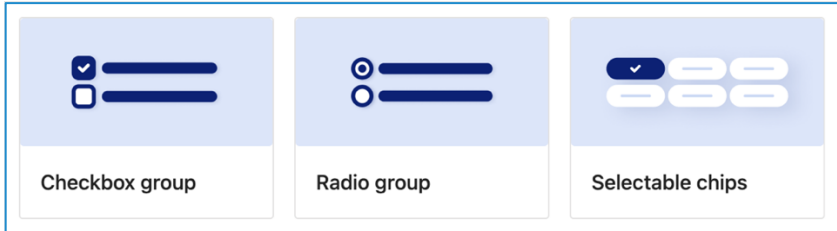
Content guidelines

- Use complete sentences. “Don't use [component name] to...” is often an easy starting point.
- It's okay to use example language for a Don't in the rules text (since we're not using captions).



Related components

This section includes links to related components in the Shield library. They may be alternatives (taken from the **We recommend** section above) or similar components.



The shortcut for each related component will include:

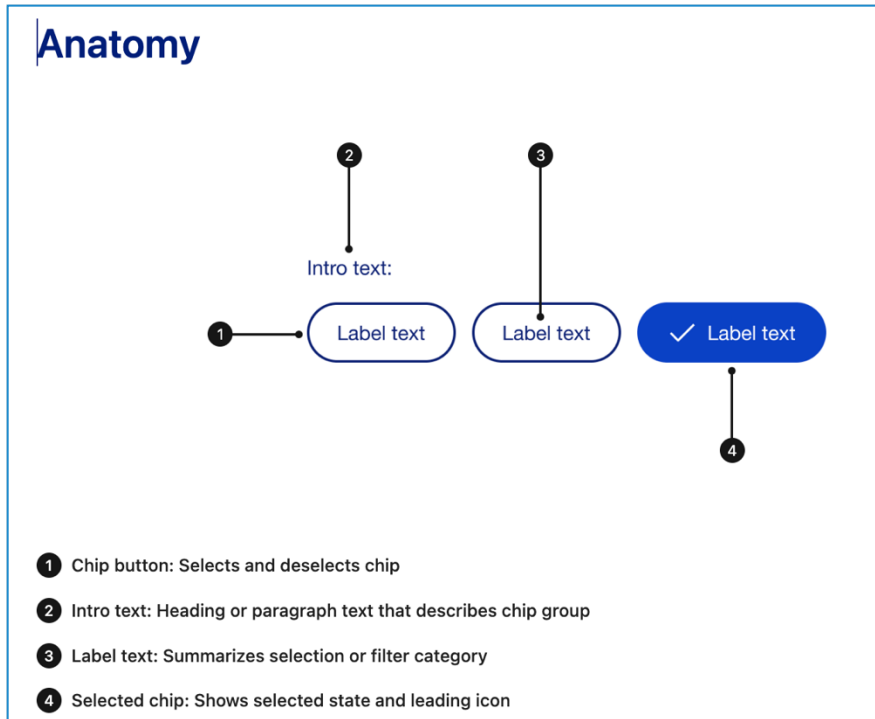
- The component name
- A lo-fi illustration (PNG) of the component
- A link to the component page

GUIDELINES TAB

H2: Anatomy

This section highlights and describes the purpose of key anatomy items.

Terms and definitions should be used consistently throughout page content and when applicable, across the library and ecosystem (i.e., a close button is called a close button on both the notifications component and the modal component and this is consistent across all documentation spaces).



Strategy notes

- Only annotate essential anatomy items.
- Use a Figma illustration.
- Use Figma annotation functionality for the legend, not tables.
- Focus on the purpose/function of the element (e.g., “Indicates whether accordion panel is collapsed or expanded”).

Content guidelines

- Use standardized language for anatomy items from the [Anatomy Item Descriptions](#) table in the appendix.
- If you must write a new anatomy item description:
 - Keep description short (one line).
 - Don’t use the name of the anatomy item in the description, start with the verb (e.g., indicates, displays, etc.).
 - Use a colon between the term and the description, capitalize first word of description, and don’t use a period (except in the rare case that the description has two complete sentences).
 - Don’t use articles in anatomy text.

- Put “if applicable” or the like in parentheses (“Displays the next page view (if available)”)

No	Yes
<ul style="list-style-type: none"> • Overlay – The overlay obscures page content behind the modal • Modal container - Contains the modal content • Heading text - Describes the message text. 	<ul style="list-style-type: none"> • Overlay: Obscures page content behind modal • Modal container: Contains modal content • Heading text: Describes message text

CS notes

- Ideally, the legend item would only be the anatomy item, and its definition and/or functionality. However, the anatomy legend items sometimes have content that doesn’t currently have a home. For example: “Action bar: Always contains 1-3 buttons.” This is more specifications information and technically shouldn’t be in a legend.
- It would be nice if we could remove definitions for obvious things (e.g., Heading text: Describes the message text) but leave for now

H2: Customization options

This section outlines out-of-the-box customization options for the component (e.g., sizes, icon/no icon, dismissible/non-dismissible) and provides guidance on the different use cases for each option.

Customization options

Nested checkboxes

- Checkboxes within the checkbox group can be nested to create a parent-child hierarchy.
- When nesting is enabled, users can check or uncheck the entire group using the parent checkbox, or they can check or uncheck individual child checkboxes below the parent.
- If some (but not all) child checkboxes in a nested checkbox group are checked, the parent checkbox will show an indeterminate state.

Parent checkbox

- Child checkbox
- Child checkbox
- Child checkbox
- Child checkbox

All unchecked

Parent checkbox

- Child checkbox
- Child checkbox
- Child checkbox
- Child checkbox

All checked

Strategy notes

- Customization options can be found in the *Properties* section of the component ReadMe file. Work with dev and design to determine which ones to highlight.

REACT Component Documentation Guidelines

- When describing an option, clearly state what it changes about the component and ideally include an illustration to demonstrate.
- If a customization option includes several choices (e.g., customizable size and you can choose small, medium or large), include a table below the illustration that breaks down usage for each choice. See [our style guide] for table language guidelines).

Content guidelines

- The H2 is always *Customization options*, followed by a custom H3 for each option highlighted.
- If there are two or more settings for an option, say the default first in the list, preferably with a (default) in parentheses.
- If one option references another option, be sure to put that option *after* the one referenced.
- If component has **required/optional** and/or **helper text** sections, put those at the end of this section.
- Sometimes content can be better parsed in bullets vs. paragraph text. Use bullets if the content seems like separate points, vs. telling a story in a paragraph.
- [INSERT IMAGES HERE]
- Use tables as necessary to break down properties into easy-to-parse chunks. In tables, keep row content as short as possible, and break up different types of information into separate columns.
 - Most tables have a “use for” column for each property/property type.
 - The text in the “Use for” column cells should not start with “use for,” as that gets repetitive. Jump right in to the usage. The text in the “use for” column should complete the sentence starting with “use for” (as seen in following screenshot).

No	Yes
Use for	
Use sparingly for the primary action on the screen, in limited in cases where the primary action does not need a label.	Primary action on the screen...
Use for high-priority actions that are understandable without a label, such as submitting or sending.	High priority actions...
Use for most icon buttons.	Most icon buttons
Use for helpful or optional actions that support the primary focus of the screen, such as zooming.	Helpful or optional actions...

- Try to come up with examples for each usage and put them in parentheses.

Type	Use for	Number per screen
Loud	Highest-value calls to action (such as <i>Apply</i> and <i>Log in</i>) or other key actions common across platforms	Usually only one
Primary	High-priority actions that guide users through multi-screen flows or indicate an important one-step action (such as <i>Submit</i>). Primary buttons command attention, but without the urgency of loud buttons.	Usually only one
Secondary	Actions less important than a primary button or those that deviate from the primary flow (such as saving an application)	Multiples OK
Text	Low-priority or less desirable actions (like canceling an application)	Use sparingly
Utility	Helpful or optional actions that support the primary focus of the screen	Multiples OK

-

Example: Borders

Here's an example describing border options:

H3: Borders

- Borders create visual distinction between table rows and columns.
- Borders can apply to only rows, or to both columns and rows.
- To increase distinction, use borders with zebra striping.

H4: Rows and columns

By default, the table displays borders around both rows and columns.

Table header	Table header	Table header
--------------	--------------	--------------

H4: Rows only

Displays borders under rows only.

Table header	Table header	Table header
--------------	--------------	--------------

H4: None

No borders are displayed.

H2: Behaviors

This section explains how the component behaves in different scenarios. It includes behaviors unique to the component (e.g., warning notifications and inline errors) and a set of behaviors that are repeated across all components.

Strategy notes

- A11Y consultants typically provide behavior guidance for the accessibility-focused sections (focus order, keyboard interaction, screen reader and responsive).

Content guidelines

- The H2 is always *Behaviors*
- H3 order should be:
 - Custom H3s for this component
 - Predefined H3s for repeated behaviors (Focus order, Keyboard interaction, Screen reader, Responsive).
- Whenever possible, include illustrations for unique behaviors (the illustration comes *after* the text).
- Sometimes content for the custom H3s can be better parsed in bullets vs. paragraph text. Use bullets if the content seems like separate points, vs. telling a story in a paragraph.
- [INSERT IMAGES HERE]

H3: [Custom Behavior heading, etc.]

H3: Focus order

This section uses a table to show focus order as a user tabs through the component.

- For very simple components, this section may be combined with the Keyboard interactions section/table below.
- The H3 is always *Focus order* (or *Focus order and keyboard interaction* if the two sections are combined)
- Ensure anatomy items mentioned here match those in the *Anatomy* section.

H3: Keyboard interaction

This section uses a table to show what happens within a component when different keys are pressed.

- In some cases, this section may be combined with Focus order above as *Focus order and keyboard interaction*.
- The H3 is always *Keyboard interaction* (or *Focus order and keyboard interaction* if the two sections are combined).
- Ensure anatomy items mentioned here match those in the *Anatomy* section.

H3: Screen reader

This section explains how screen readers announce component content.

REACT Component Documentation Guidelines

- Note: Don't edit text in this section as it's provided by A11Y group and may have specific language (e.g., don't change "label of button" to "button label").
- The H3 is always *Screen reader*.
- Ensure anatomy items mentioned here match those in the *Anatomy* section.
- Don't use articles to introduce the bullet text.

No	Yes
<ul style="list-style-type: none">• Legend• The checkbox label and image alt text (if an image is present)• The role of the checkbox input• The state of the checkbox	<ul style="list-style-type: none">• Legend• Checkbox label and image alt text (if an image is present)• Role of the checkbox input• State of the checkbox

H3: Responsive

This section explains what happens to the component in responsive views.

- The H3 is always *Responsive*.
- When possible, include an illustration.
- Ensure anatomy items mentioned here match those in the *Anatomy* section.

H3: Validation and errors

Some components have a *Validation and errors* section in the Behaviors section. Use the following format/order.

Order:

1. Bullets
2. Image (optional)
3. Validation rule/error message table
4. Yellow callout Note (copy and paste the following message verbatim)

Note: This component's default error messages should be customized. Work with a content designer and use the [Error Message Resource](#) to determine the best language for each error scenario.

**Exception: on components that have multiple validation error tables, put this note BEFORE the tables (after the bullets) and then it should say:

Note: This component's default error messages (shown below) should be customized. Work with a content designer and use the [Error Message Resource](#) to determine the best language for each error scenario.

Note location with one error table

Note location with multiple error tables

Validation and errors

- Field validation occurs after the user has exited the text input field (`onBlur`).
- If a validation error occurs, an error message appears below the input field but above any helper text.

Event type	Validation rule	Error code	Default message
<code>onBlur</code>	Field is required.	224	This information is required.

Note: This component's default error messages should be customized. Work with a content designer and use the [Error Message Resource](#) to determine the best language for each error scenario.

Validation and errors

- Validation occurs when the user is typing (`onChange`), after they've tabbed out of a date input (`onBlur`) and when they attempt to paste a value into a date input (`onPaste`).
- If a validation rule is broken, an error message appears below the input field.

Note: This component's default error messages (shown below) should be customized. Work with a content designer and use the [Error Message Resource](#) to determine the best language for each error scenario.

General

Event type	Validation rule	Error code	Default error message
<code>onChange</code>	Entry must be some combination of numbers, hyphens, periods, forward-slashes and spaces	118	Enter a valid <label> date.
<code>onPaste</code>	Pasted entries aren't allowed in confirmation fields	130	Please enter your <label> date again.

Date format: mm/dd/yyyy

Event Type	Validation rule	Error code	Default error message
<code>onBlur</code>	Must be a valid date	118	Enter a valid <billing> date.
<code>onChange</code>	Max of 8 characters is allowed	121	You've reached the character limit. Please try again.
<code>onBlur</code>	Field can't be empty	224	Billing date is required.

H2: Content

(optional)

This section provides guidance on content specific to the component (e.g., Writing warning notifications, Ampersands and tabs, Writing helper text, Alternative text, etc.) but does NOT repeat overarching guidance from the content style guide.

Content

Label text

- Label text should describe the action that will occur when the user checks the box.
- Punctuate full sentences.

Text alternatives for images

- When an image is present, screen readers announce the visible label text, followed by the text alternative. Always check the reading order to ensure this combination is clear and understandable.
- Text alternatives should be meaningful and not repetitive. For example, don't use the word *Spotify* as a text alternative for a Spotify logo when the label is also *Spotify*. This will cause the screen reader to announce "Spotify Spotify."
- If an image is purely decorative, don't provide a text alternative.

Content guidelines

- The H2 is always *Content*.
- H3 order should be (if these sections are present):
 - Any custom content sections
 - Intro text
 - Label text
 - Text alternatives for images
 - Off-screen text
- When relevant, include illustrations or tables.
- Ensure anatomy items mentioned match those in the *Anatomy* section.

H2: Best practices/More best practices

(optional but recommended)

The do’s and don’ts in this section provide guidance on how to properly use or configure the component and should pertain to design and/or code.

More best practices

3.875%

10 year adjustable-rate mortgage.

Prequalify

Centering button underneath centered content block

Lowest

Lower

Highest

Primary (far right), secondary (middle), text (far left)

✓ Do

✓ Do

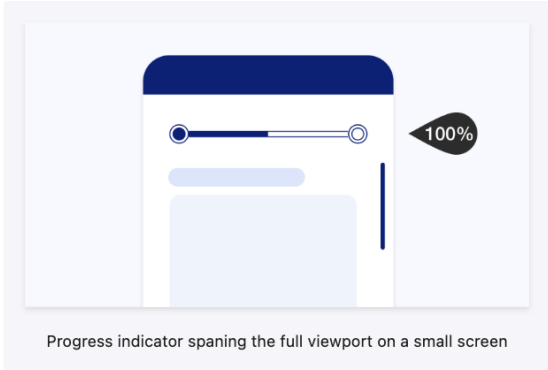
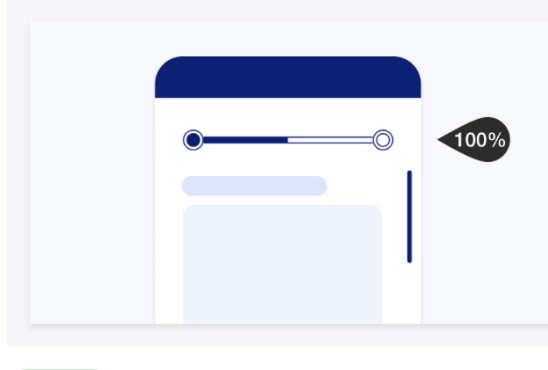
Allow buttons to inherit the alignment of the blocks when placing them within content blocks.

Arrange button groups in order of importance.

Strategy notes

- Any high-level dos and don’ts (such as when to use/not use) should be relocated to the *Overview* tab.
- In the past all do’s and don’ts lived in one section, but moving forward, place each do/don’t along with the relevant section. For example, place Do’s and Don’ts about button styles in the *Button type* section.
- This section may contain some “caution” rules.
- Use the same guidance for creating Do’s/Don’ts as in the previous *Use [component]/Don’t Use [component]* sections on the *Overview* tab.
- You may not need to use an image caption, as the guidance text usually describes the example.

No	Yes
----	-----

 <p>Progress indicator spanning the full viewport on a small screen</p> <p>✓ Do</p> <p>Span the full viewport width on small screen sizes below 768px.</p>	 <p>✓ Do</p> <p>Span the full viewport width on small screen sizes below 768px.</p>
---	---

No	Yes
<p>Customization option: compact, default, spacious</p> <p>Don't use a large row height...</p>	<p>Don't use spacious row height...</p>

Content guidelines

- We renamed this section *Best practices* (it was formerly known as *Do's and Don'ts*)
- If the guidance is for a customization option, use the name of the option in the rules text.
- Changing the H2 *Best practices* heading name:
 - If you've moved *all* the do's and don'ts to relevant sections, delete the H2 *Best practices* section.
 - If there are some do's and don'ts left over (that don't relate to any existing sections), rename the H2 *More best practices*.

CODE TAB

Repeated elements:

- Component name
- Component description
- Tabs

Technical overview

More guidance coming soon. For now, this is just a partial copy and paste directly from the ReadMe file which lives in the component package in GitLab.

Accessibility

This section includes A11Y guidance specific to developers. It typically includes information regarding HTML native, role and aria.

A11Y consultants always provide this content but may need your guidance on formatting it.

Content guidelines

- The H2 is always *Accessibility*, followed by a custom H3s (as needed).
- Use bullet points as needed to add additional organization to the content.

PATTERNS

[note official pattern template is in development, just capturing some decisions we've made so far]

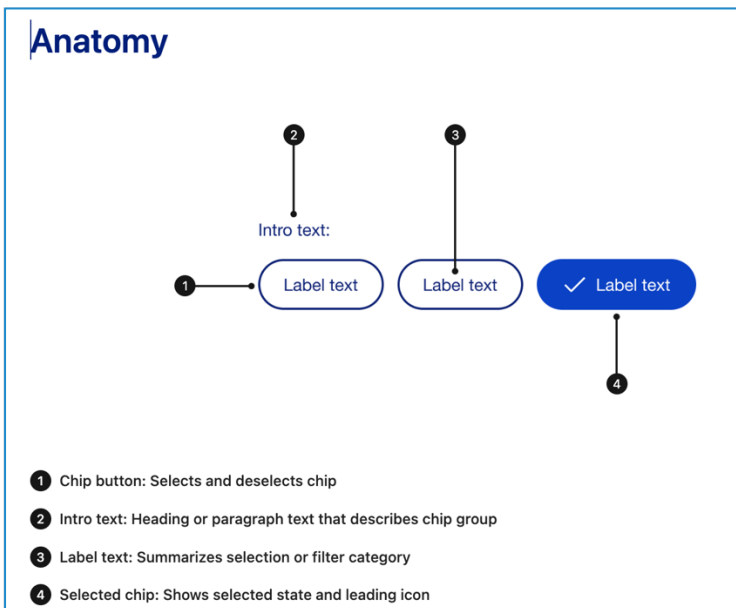
- Can include do's/don'ts throughout
- Try not to duplicate content found with each component; link out where possible.

Tabs:

- Overview
 - Short description
 - Anatomy
 - Usage
 - Bullets
 - [Component used]
 - Accessibility considerations
 - [component used]
 - Accessibility considerations
 - Related components
- Scenarios
 - [scenario 1]
 - Accessibility considerations
 - [scenario 2]
 - Accessibility considerations
 -

Anatomy

- In an anatomy for a pattern, only call out the components used in the pattern.
- Don't call out the individual pieces of each component (like icon, label text, etc.).
- Don't use a name/description pair for the legend.



FORMATTING

Adding emphasis

Per the COP guidance, only two types of emphasis can be used:

- Quotes (for screen reader announcements)
- Italics (for everything else)

Italics can be used when showing examples, for such things as:

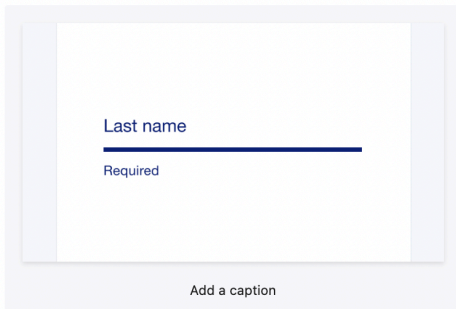
- Field names
- Button names
- Quoting helper text

Exception: Quotes are used in rules text in Zeroheight

The Zh Rules component, which we use for our Do's and Don'ts/Best Practices, doesn't allow *any* text formatting. So occasionally we've had to use quotes to set off a word for emphasis.

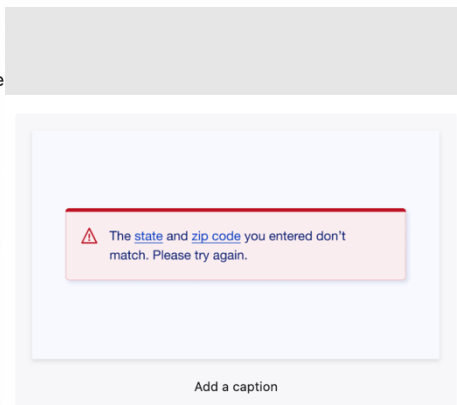
Helper text

Helper text can be enabled to provide additional guidance



✗ Don't

Don't use "Required" as helper text except for in rare cases where there are fewer required fields than optional fields.



✓ Do

Use "please" to soften two-part messages that would otherwise sound bossy or brusque.

WRITING GUIDANCE

Terminology (words not addressed in U.S. Bank Brand Guide)

- Opt in (verb) /Opt-in (adjective)
- Full date, partial date
- (if applicable) should be in parentheses
- Screen reader (two words)
- Autoformat
- styleguide (one word, a Zeroheight set of documents) vs. “style guide” (two words), a document that tells you how to write stuff
- Activate: used to describe what you do to a button (vs. clicking)

How to say it (preferred words/phrases)

Current	Preferred
<p>Image Picture</p> <p>When describing something non-text in the documentation.</p>	<p>Illustration</p> <p>E.g., “Use an illustration to clarify the guidance.” We prefer “illustration” to “image” or “picture” because most of the time it will be an illustration from Figma.</p>
<p>“Consider”</p> <p>Try not to use “consider” as it doesn’t carry any authority.</p> <p>“Consider using the dropbox in [situation]...”</p>	<p>“We recommend”</p> <p>“We recommend” has more weight and positions us as the experts.</p> <p>“We recommend using the drop box in [situation].”</p> <p>Related language:</p> <ul style="list-style-type: none"> • It’s best to.../best not to • [blah] is the best choice
<p>Rarely</p> <p>“Rarely” has a sort of time component to it.</p>	<p>Sparingly</p> <p>“sparingly” is better, because it connotes that you are using discretion in use of [the thing].</p>
<p>Small list</p>	<p>Short list</p> <p>Note: we don’t officially define what a “short list” is</p>
<p>Large list</p>	<p>Long list</p> <p>Note: we don’t officially define what a “long list” is</p>
<p>Preset</p>	<p>Predefined</p>
<p>Take user to a new location (links)</p>	<p>Send user to a new location</p>

REACT Component Documentation Guidelines

<ul style="list-style-type: none">• Multi-step process• Tasks that span multiple steps and screens• Complex task or flow	Multi-step flow
To avoid a disorienting experience <i>Avoid means to stay away from something already happening (avoid the car accident by changing your route).</i>	To prevent a disorienting experience <i>Prevent is to stop it from happening in the first place.</i>
When someone interacts with a button, an action occurs right away .	When someone interacts with a button, an action occurs immediately .
“when it” vs. “to [verb]” Use a button when it completes an on-page action...	Use a button to complete an on-page action.

GENERAL WRITING TIPS

See U.S. Bank style guide for general style guidance. But in short use:

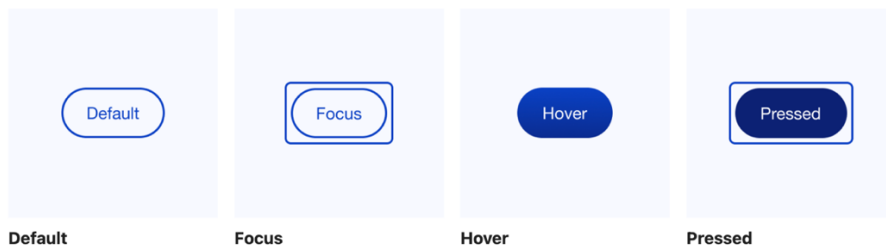
- Active language (vs. passive)
- first person vs. third
- Informal vs. formal language

List order/Image order sync

If text contains a list, and then images below, the image order should mirror the list order. In the example below, the list and image order don't match. It's a minor thing, but consistency is king!

States

Selectable chips inherit the states (default, hover, focus, and **pressed**) and source styles of the button component. When a chip is selected, a checkmark icon appears in front of the label text and the chip's container color changes.

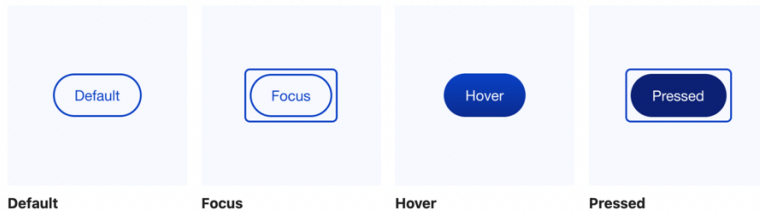


Here, they match.

Behaviors

States

Selectable chips inherit the states (default, focus, hover and pressed) and source styles of the button component. When a chip is selected, a checkmark icon appears in front of the label text and the chip's container color changes.



Use contractions

Contractions make content sound friendlier, less formal.

Formal	Informal
Do not change the default browser behavior.	Don't change the default browser behavior.

Don't need to write "for the user" or "to the user"

It's assumed everything is for the user.

No	Yes
Provides brief, supplementary information to the user	Provides brief, supplementary information
Opens the message container when activated by the user	Opens message container

To save space, use numerals instead of spelling out numbers less than 10

No	Yes
Options that vary in length or are longer than twelve characters	Options that vary in length or are longer than 12 characters

Don't use indefinite language UNLESS you provide an explanation

If you use indefinite language like "might," "possibly" or similar, explain circumstances in which [X] might be true.

No	Yes
<p>It's possible to apply button styling to the link component when necessary.</p> <p><i>(The reader wonders, "When would it be necessary?")</i></p>	<p>You can apply button styling to a link component in [X], [Y] or [Z] situation.</p>

Anchor link text when linking to component

Anchor link text should use correct component name.

No	Yes
When more than one selection is allowed, use checkboxes .	When more than one selection is allowed, use checkbox group .

Example language, short

In short text, or where example is just a few words, try not to say "for example." Use "like" and "such as." Don't ever use "e.g."

No	Yes
----	-----

The loud button style is used for the highest-value calls to action (e.g., Apply and Log in) or other key actions common across platforms.	The loud button style is used for the highest-value calls to action (such as Apply and Log in) or other key actions common across platforms.

Example language, longer

If you are using an example in longer text, say the “for example” at the beginning, not the end.

No	Yes
“The continue button is always in the same place, for example.”	“For example, the continue button is always in the same place.”

When referring to other sections

When referring to other sections in proximity, use “previous” and “following” instead of “above” or “below.”

States

Something is “in” a state, it doesn’t “have” a state.

No	Yes
<ul style="list-style-type: none"> • A checkbox is either unchecked or checked. • A checkbox has either an unchecked or checked state. 	A checkbox is in either an unchecked or checked state.

Ensure prop name and prop description are the same

In this example we’re writing about the “expand cell” functionality.

Expand table

The **expand cell** functionality allows users to view more data related to a row/record. **Expand cell** is useful when the additional details need to be available to users but may not be frequently accessed.

User reading thinks, *Hey the heading says “Expand table” but then they’re calling it “expand cell.” Which is it? Is this a mistake?* And once they start wondering if something is incorrect, you’ve lost their trust. (In this case, the Expand table functionality is an umbrella term over two other types of functionality: expand cell and expand row.)

What if part of the functionality/option isn’t available yet?

If a property has several parts, but one is in development, but each can be understood without the other, leave the “under development” part out. But if the second part is needed to properly explain the option, put a mention of it in, with the note, “[property] coming soon” or “[property] in development.”

In the following example, the property has two strings, Expand Cell and Expand Row. Expand row is still in development. To make this section make more sense, it would help to add the second string in just a short sentence like this.

Before	After
<p>Expand table</p> <p>The expand cell functionality allows users to view more data related to a row/record. Expand cell is useful when the additional details need to be available to users but may not be frequently accessed.</p>	<p>H3: Expand table</p> <p>The Expand table option allows the table to hold additional information (revealed through user action) related to a row/record. There are two types: <i>Expand cell</i> and <i>Expand Row</i> (currently in development).</p> <p>H4: Expand cell</p> <p>The expand cell functionality allows users to view more data related to a row/record. Expand cell is useful when the additional details need to be available to users but may not be frequently accessed.</p>

Now the “expand cell” type has the needed context.

When should you use a table for information?

- When information is chunked up to be easily read at a glance.
- When information is short (one or two lines)

Good use of a table

Key	Function
tab	Moves focus to and from a button
enter or space	Selects the button

Not great use of a table

Difficult to see	Disabled buttons are not required to meet color contrast ratios which can make them hard to see. If contrast requirements are met, users may find it hard to distinguish differences between disabled and enabled buttons.
No user feedback	If a user tries to engage a disabled button, they receive no feedback. This can cause confusion because it's often unclear why the button is disabled.
Inconsistent user experiences	Disabled buttons are not keyboard accessible and screen readers handle them in various ways, which makes it difficult to create a predictable user experience.

Why? Users expect table content to be short and easy to parse. The information in this table has points of discussion/persuasion for someone to not use disabled buttons. This information is better in bullets, like the following.

Use **bold** for main point, then regular weight text for explanation. Can use tables with “rows only” lines to separate.

Disabled buttons are not recommended because they cause usability and accessibility issues.

- **They're difficult to see.** Disabled buttons are not required to meet color contrast ratios which can make them hard to see. If contrast requirements are met, users may find it hard to distinguish differences between disabled and enabled buttons.
- **No user feedback.** If a user tries to engage a disabled button, they receive no feedback. This can cause confusion because it's often unclear why the button is disabled.
- **They don't provide consistent user experience** for keyboard and screen readers. Disabled buttons are not keyboard accessible and screen readers handle them in various ways, which makes it difficult to create a predictable user experience.

Options tables

Content in tables should be written differently than running text. Tables should break up information into easy-to-parse chunks.

- In tables that describe multiple options/styles that have the name of the option/variant in the first column, you don't need to have an introductory phrase in the description. It slows down the reader.
- For example, in a table describing Button Styles, don't need to repeat the name of the style in the "use for" column.

Before		
Style	When to use	
Loud	Use loud buttons sparingly. They're reserved for the highest-value calls to action, such as Apply and Log in or other key actions that are common across platforms.	
	Avoid using more than one loud button on a screen.	
Primary	Use primary buttons as the default for high-priority actions. Primary buttons command attention, but without the urgency of loud buttons. They help guide users through multi-screen flows or indicate an important one-step action.	
	Generally, each screen has only one primary button.	
Secondary	Use secondary buttons for actions less important than primary button or that deviate from the primary flow, such as Cancel or Save application.	
	It's fine to use multiple secondary actions per screen.	
Utility	Use utility buttons for helpful or optional actions that support the primary focus of the screen.	
Text	Use text buttons for lower-priority actions or those we would prefer a user not take (such as canceling an application).	
After		
Type	Use for	Number per screen
Loud	Highest-value calls to action (such as <i>Apply</i> and <i>Log in</i>) or other key actions common across platforms	Usually only one
Primary	High-priority actions that guide users through multi-screen flows or indicate an important one-step action (such as <i>Submit</i>). Primary buttons command attention, but without the urgency of loud buttons.	Usually only one
Secondary	Actions less important than a primary button or those that deviate from the primary flow (such as saving an application)	Multiples OK
Text	Low-priority or less desirable actions (like canceling an application)	Use sparingly
Utility	Helpful or optional actions that support the primary focus of the screen	Multiples OK

When to use bullets vs. paragraph text

Bullets are great for chunking out information to make it more scannable.

Use bullets if the points being made in the paragraph can be separated. For example, this paragraph is giving advice about radio button and radio button group legend text. It works better as bullet points because there are several separate pieces of advice.

Before	After
<p>Keep radio button label text as short as possible. The legend for the group should introduce the choices. This legend can be a full sentence but keep it concise. Legend and label text must be unique.</p>	<ul style="list-style-type: none"> • Keep radio button label text as short as possible. • The legend text for the group should introduce the choices. • Legend text can be a full sentence but keep it concise. • Legend and label text must be unique.
<p>Buttons have three size options: large, medium (default) and small. Each button’s size has a fixed container height. Button width varies, depending on the length of the button text.</p>	<ul style="list-style-type: none"> • Buttons have three size options: large, medium (default) and small. • Each button’s size has a fixed container height. • Button width varies, depending on the length of the button text.

Consistency

Give parallel examples

If an example is given for one item in a list or a table, should also given examples for others (if possible).

In the following screenshot, examples are given for three out of five styles. Should have examples for all.

Style	When to use
Loud	<p>Use loud buttons sparingly. They're reserved for the highest-value calls to action, such as Apply and Log in or other key actions that are common across platforms.</p> <p>Avoid using more than one loud button on a screen.</p>
Primary	<p>Use primary buttons as the default for high-priority actions. Primary buttons command attention, but without the urgency of loud buttons. They help guide users through multi-screen flows or indicate an important one-step action.</p> <p>Generally, each screen has only one primary button.</p>
Secondary	<p>Use secondary buttons for actions less important than primary button or that deviate from the primary flow, such as Cancel or Save application.</p> <p>It's fine to use multiple secondary actions per screen.</p>
Utility	<p>Use utility buttons for helpful or optional actions that support the primary focus of the screen.</p>
Text	<p>Use text buttons for lower-priority actions or those we would prefer a user not take (such as canceling an application).</p>

REFERENCE

Component descriptions (as of Jan 2023)

We tried to make all these consistent in the structure of a [component] [verbs] because it keeps the focus on the component and keeps the description short. Unfortunately, it made some sound strange. So we allow a few different structures. The current options, in order of desirability, are:

Structure	Example
A [component] [verbs]	Accordions expand and collapse sections of related content.
A [component] is/are	Selectable chips are buttons that filter items from a list.
A [component] is + lets users	Selectable chips are buttons that let users filter content...
A [component] lets users [do something]	A button lets users perform a task.

Component	verb	Description
Accordion	active	An accordion expands and collapses sections of related content.
Breadcrumbs	active	Breadcrumbs provide secondary navigation to orient users in a navigational path.
Button	lets	A button lets users make a choice or perform a task. When someone interacts with a button, an action occurs immediately.
Calendar	is/lets	A calendar is an interactive grid that lets users quickly navigate days, months and years in the future or past.
Checkbox	lets	A checkbox lets users make a single selection to offer acknowledgement or agree to an action.
Checkbox group	lets	A checkbox group lets users make multiple selections from a short list of options.
Chips (R)	is/lets	Removable chips are buttons that let users filter data by removing options from a group.
Chips (S)	is/lets	Selectable chips are buttons that let users filter content by choosing from a predetermined set of options.
Combobox	active	A combobox filters a list of predefined options based on user input, then allows a single selection.
Date input	lets	A date input lets users enter a full or partial date through a text input.
Date picker	lets	A date picker lets users enter a date through a text input or calendar dialog.
Divider line	is	Divider lines are thin, horizontal lines that visually group or separate content.
Dropdown	lets	A dropdown lets users select one navigation or filtering option from a list.
Link	active	A link sends users to another page, screen or location. It can also be used to download files.
Modal	active	A modal is a dialog that disables current page content and focuses the user's attention on a single task or message.
Notification	active	A notification provides essential, immediate feedback about an action or flow.
Pagination	lets	Pagination lets users navigate back and forth through a large collection of related content.
Progress Indicator	active	A progress indicator orients users within a specific task flow.
Radio group	lets	A radio group lets users make a single selection from a short list of predefined options.

Select	lets	Select lets users choose one item from a short list of options.
Tabs	active	Tabs organize content at the same level of hierarchy into a compact horizontal display.
Text area	lets	A text area lets users input multiple lines of text, usually within a form.
Text input	lets	A text input lets users enter a single line of information, usually within a form.
Toggletip	is	A toggletip is a user-activated pop-up message that defines or enhances surrounding content.

Language for referring to components/user experience

This language ideally can/should be used in multiple areas: component descriptions, the

We recommend table, and other places that refer to component functionality.

Component	Language
Button	Taking action or submitting input
Checkbox	User must acknowledge something or agree to an action
Checkbox group	Multiple selections allowed from a short list <ul style="list-style-type: none"> Except text area/text input: Capturing multiple selections from a short list (the field--used incorrectly--captures user's choices from a list)
Combobox	One selection allowed from a long list where users will likely know their input
Confirmation page	Confirming success after completion of a complex task or multi-step flow
Date Picker	When users need context of a calendar to pick a date OR Entering date range selection
Date input	Entering memorable dates that don't require a calendar OR Entering partial dates in month/year (mm/yy) or year (yyyy) formats
Link	Sending user to a new location
Notifications	Confirming success of a specific task or communicating issues or errors
Radio group	One selection allowed from a short list Except text area/text input: Capturing user's selection from a short list (the field--used incorrectly--captures user's choice from a list)
Separate page	Confirming success after completion of a complex task or multi-step flow OR

	Alerting users to system or connectivity issues
Modal	Focus the user’s attention on a single task or message with a disruptive dialog
Toggletip	Supplemental content that only appears when activated by user

For example, here is the language used in multiple places.

Component	Component description	We recommend table situation
Checkbox	A checkbox lets users make a single selection to offer acknowledgement or agree to an action.	User must acknowledge something or agree to an action
Checkbox group	A checkbox group lets users make multiple selections from a short list of options.	Multiple selections allowed from a short list

Overview > Do’s Example Language

Component	Guideline text	Example text
Accordion	Use accordions to condense content on text-heavy screens.	Using an accordion for FAQs
Button	Use buttons for on-page actions.	Using a button to open a modal
Button	Use buttons to progress within a flow.	Using a button to continue with an application
Button	Use buttons to submit user input.	Using a button to submit an online payment
Checkbox	Use a checkbox to offer acknowledgment.	Agreeing to terms and conditions
Checkbox	Use a checkbox to agree to an action.	Opting in to receive email updates
Checkbox group	Use a checkbox group when multiple selections are allowed.	Asking users for sources of income
Removable chips	Use removable chips to apply or remove custom filters.	Choosing specific dates from a calendar to filter transactions
Removable chips	Use removable chips to display data.	Displaying a group of files poised for upload
Selectable chips	Use selectable chips to filter by predefined categories.	Choosing from a list of cash-back categories
Selectable chips	Use selectable chips in compact horizontal areas where radio groups or checkbox groups would compromise usability.	Placing categories below a search field in a narrow area
Combobox	Use combobox to select from long lists.	Selecting from a list of countries
Combobox	Use combobox to support complex forms that have limited space.	Filling out an application
Date input	Use the date input to enter a memorable date.	Entering a birth date
Date input	Use the date input to enter a partial date.	Entering a card expiration date (mm/yy)

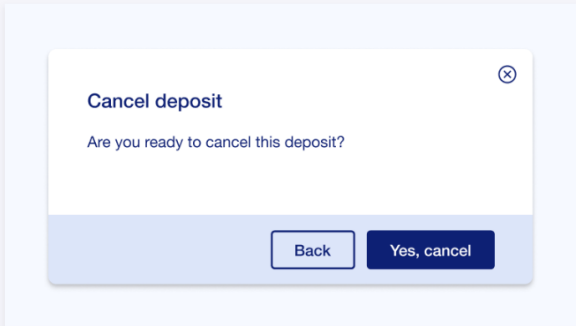
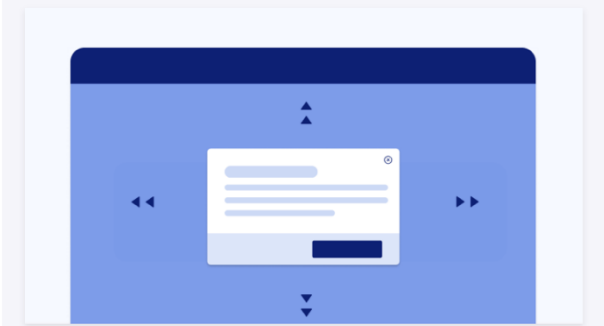
Date picker	Use a date picker to let users pick dates for a date range.	Selecting a date range to filter transactions
Dropdown	Use dropdowns to update on-page content.	Choosing number of items displayed on a page
Dropdown	Use dropdowns to help filter results.	Filtering account activity by type

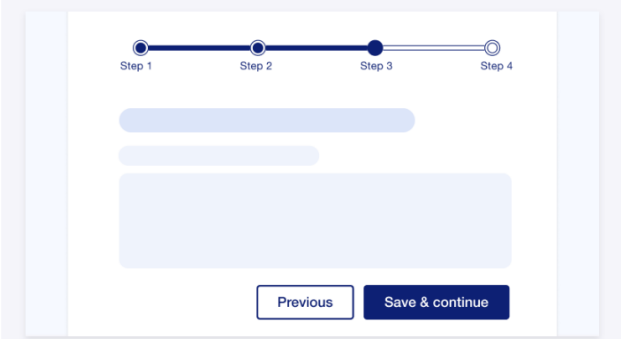

Anatomy item descriptions

Common Anatomy Item	Description Structure	Examples
Heading text	Describes [thing]	Describes content block (accordion) Describes message text (modal)
Intro text	Heading or paragraph that describes [thing]	Heading or paragraph text that describes chip group
Label text	Describes [thing] Summarizes [thing]	Describes checkbox value (checkbox) Describes expected user input (date input) Summarizes selection or filter category (chips)
Legend text	Names [thing]	Names checkbox group Names radio button group
Helper text (optional)	Offers field guidance	Offers field guidance (same for all)

Do’s and Don’ts vs. Best Practices Language

Do’s and Don’ts on the Overview page cover high-level usage, and help the user determine whether the component is right for their situation. In contrast, Best practices on the Guidelines tab provide guidance on how to design and configure the component.

Do’s (Overview)	Best Practices (Guidelines)
 <p>Confirming exit (and warning of consequences)</p> <p>✓ Do</p> <p>Use modals to confirm user actions that may have critical consequences.</p>	 <p>Centered modal in desktop view</p> <p>✓ Do</p> <p>Place modals in the center of the screen.</p>

 <p>Showing progress in multi-step application</p> <p>✓ Do</p> <p>Use the progress indicator for tasks that span multiple steps and screens.</p>	 <p>Add a caption</p> <p>✗ Don't</p> <p>Don't exceed a max viewport width of over 75% on larger screen sizes of 768px and above.</p>