zelis

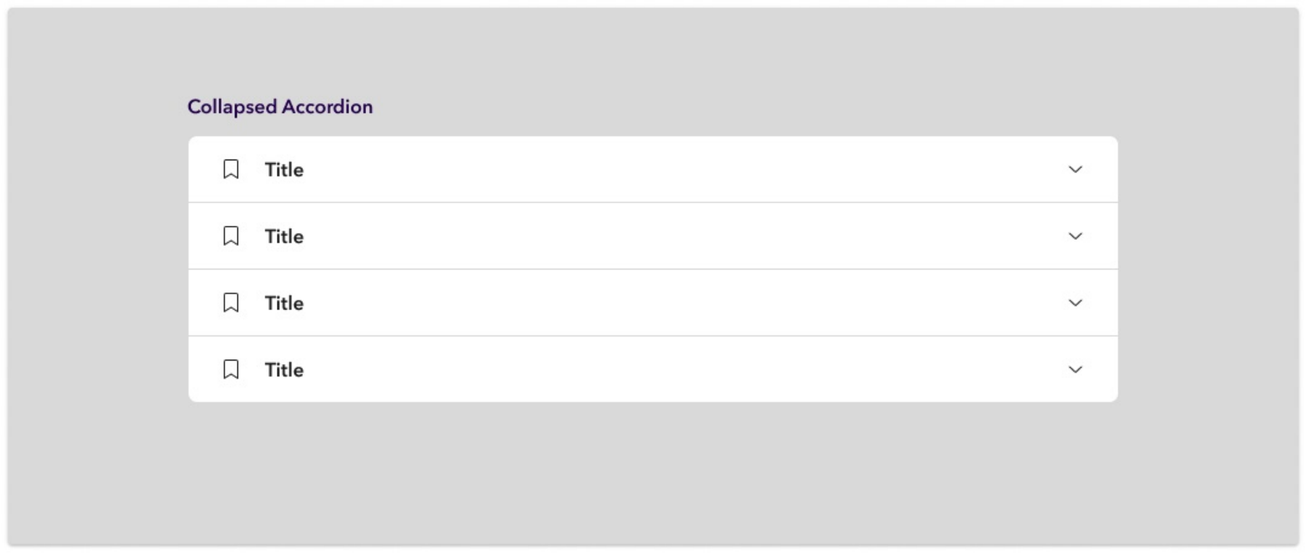
Accordion

Design Kit Figma

An accordion is a component with one or more expandable sections. The accordion component delivers large amounts of content in a small space through progressive disclosure.

Accordion

Multiple expansion-panels can be combined into an accordion, therefore inheriting styles and atoms of the expansion panel component. The multi="true" input allows the expansions state to be set independently of each other. When multi="false" (default) just one panel can be expanded at a given time. Meaning a user can then independently expand each section of the accordion allowing for multiple sections to be open at once.



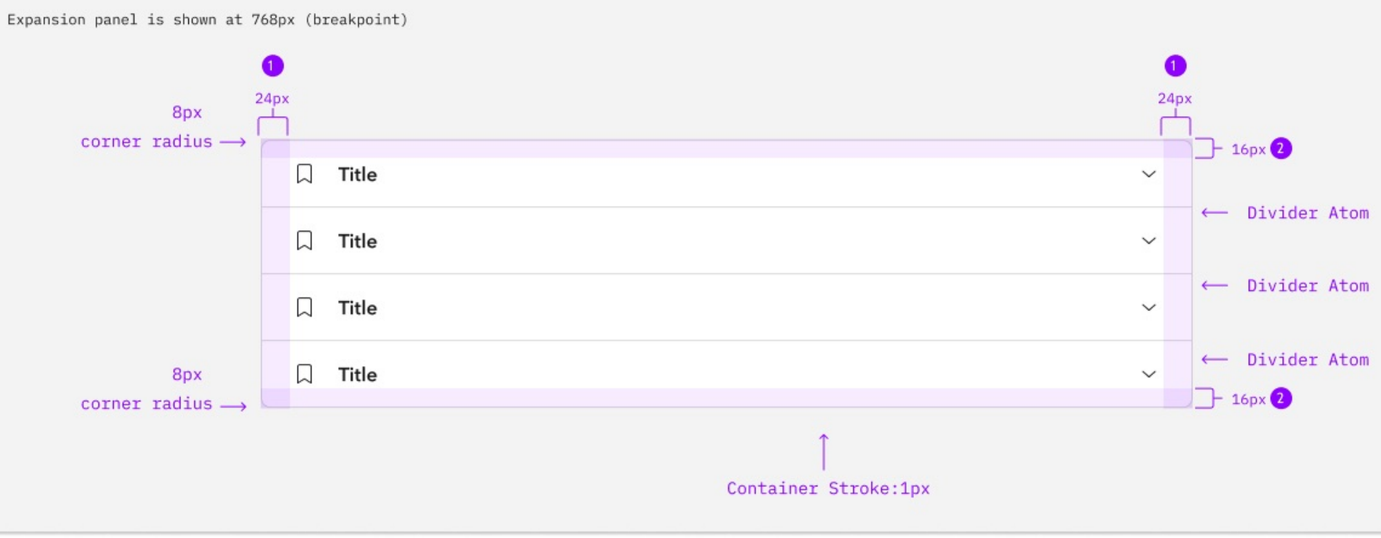
Measurements + interactions

Accordion (multiple expansion panels collapsed):

Padding/Spacing:
1. **Left/right padding** = 24px
2. **Top & bottom padding** = 16px

Container Style:
- **Container stroke =** 1px outer stroke; grey 300 and can be toggled on/off
- **The Divider atom =** is used to separate the expansion panel header. It is optional, can be toggled on/off

Corner Radius:
8. **Corner Radius all 4 corners** = 8px when collapsed



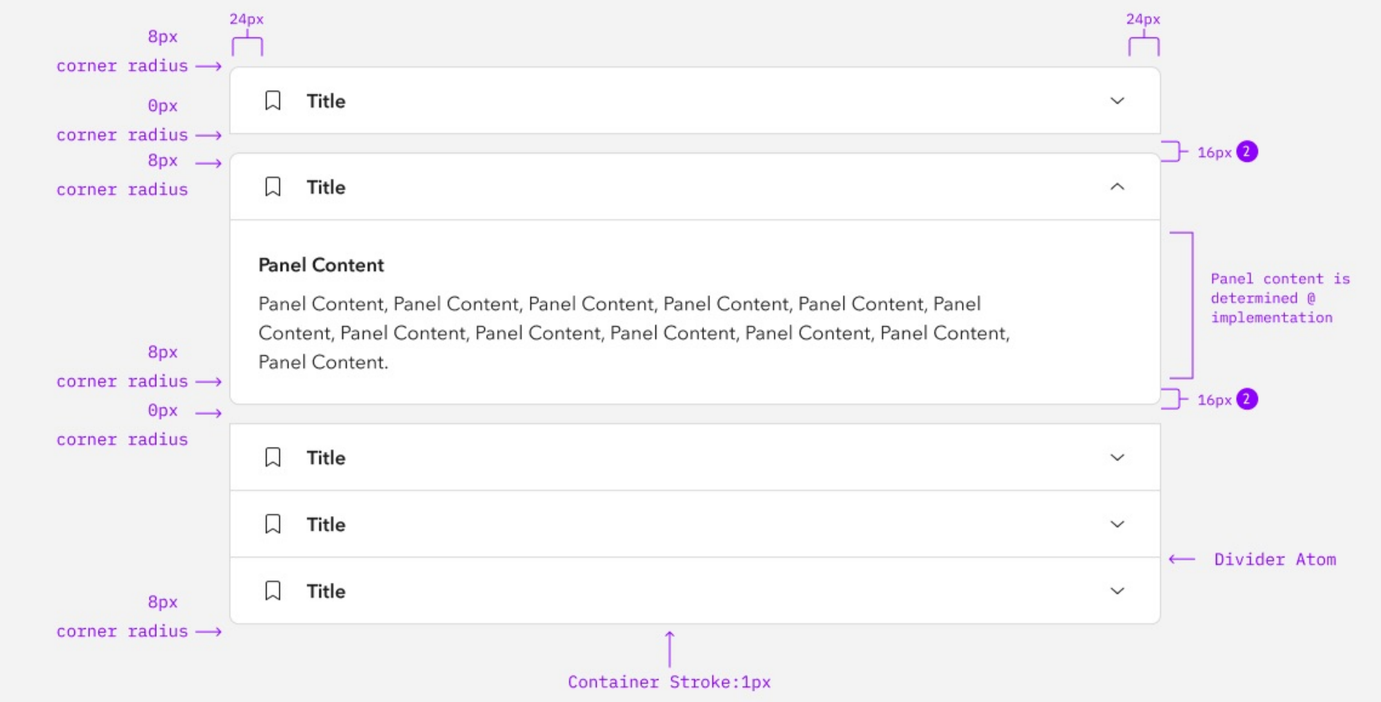
Middle expansion panel expanded:

Padding/Spacing:
1. **Left/right padding** = 24px
2. **Top & bottom padding** = 16px

Container Style:
- **Container stroke =** 1px outer stroke; grey 300 and can be toggled on/off
- **The Divider atom =** is used to separate the expansion panel header. It is optional, can be toggled on/off

Corner Radius When a middle expansion panel is expanded:
- **All 4 corners of the middle expansion panel are 8px.**
- **The expansion panel above, the top corners are 8px and the bottom corners are 0px.**
- **The expansion panel below, the top corners are 0px and the bottom corners 8px.**

Interaction:
- **Upon expansion of a middle panel a spacing of 16px occurs between top and bottom expansion panels**



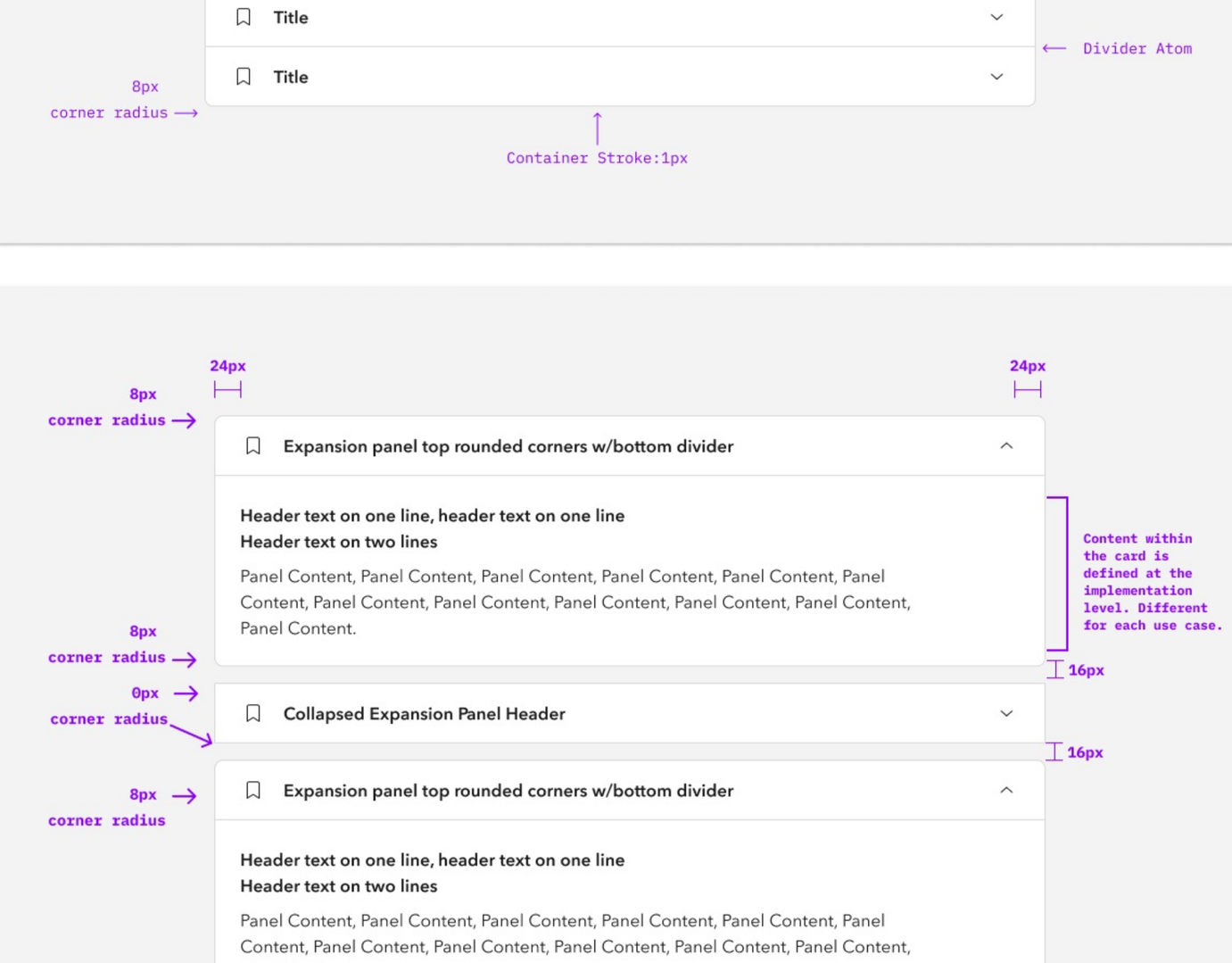
Middle expansion panel collapsed:

Padding:
- **Left/right padding** = 24px
- **Padding inbetween atoms** = 8px or 16px

Container Style:
- **Container stroke** = **is a stroke optional** 1px grey 300 (inherits card atom styles)
- **The Divider atom** = Used to seperate the expansion panel headers

Corner Radius:
- **When a middle expansion panel is collapsed =**
  - All 4 corners of the middle expansion panel are 0px.
  - The expanded panel above all 4 corners are = 8px
  - The collapsed panel below, all 4 corners are = 8px

Interaction:
- When a middle expansion panel is collapsed, and the top and bottom panels are expanded a spacing of 16px occurs between the panels.



zelis

Avatars

Design Kit Figma

An avatar is an object that represents the user's identity on screen. The user avatar creates a connection between a user and a product. Avatars can also form connections between users. Well-designed avatars humanize product experience and simplify communication.
Saving screen space - with less visual clutter to distract a user, designers use visual images instead of text characters to convey information.
Improving comprehension - Users rely on images instead of text labels to identify other users.
While generic avatars serve the functional purpose of representing objects, they lack one important component of effective UI – visual identity – something that makes the product stand out from the crowd.

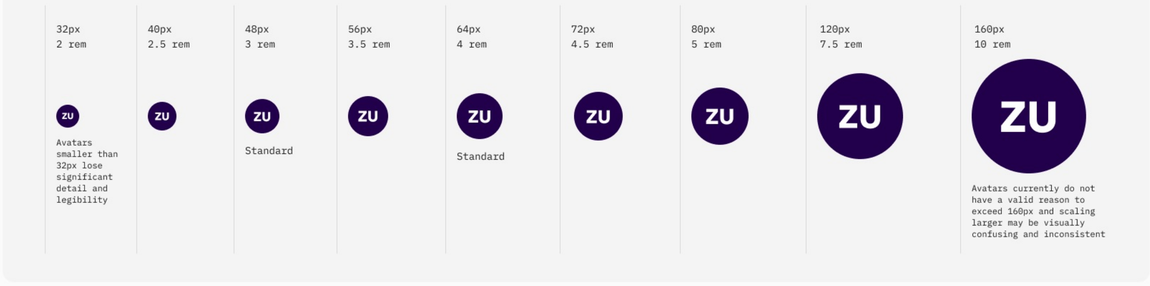
Usage

TYPE	AVATAR	USE
Initials	ZU	• text is the user's initials • at 64px avatar, text is H3 bold and centered within the circle • if scaling is needed, make sure text is proportional starting from the 64px point • 120px in member profile

Size

Standard avatar sizes are 64x64 for desktop and 48x48 on mobile. With the introduction of a member profile avatar, a sliding scale will be implemented that continues to observe the 8-pixel base rule. If straying from standard desktop or mobile sizes, include justification for those deviations.

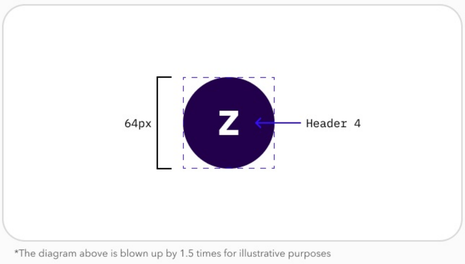
While scaling avatars, maintain proper proportions (which can be done by pressing "k" on the keyboard to transform). The minimum size for an avatar is 32px and the maximum size is 160px. The following example uses a provider avatar:



Initial Avatar Specs

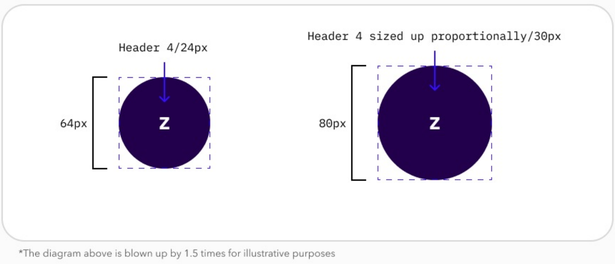
Initial Avatars are used as a simple way to identify names (a user or brand) without using illustrations or images. To keep avatars consistent across platforms and different sizes, the text should be scaled proportionally with its background.

At 64 x 64px, the text should use Header 4 type centered both horizontally and vertically.



*The diagram above is blown up by 1.5 times for illustrative purposes

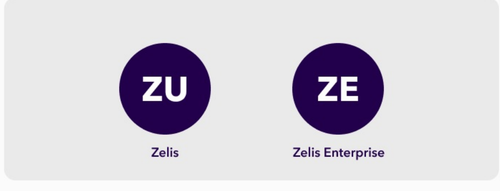
To size up or down proportionally, press "k" in Figma first, then resize appropriately.



*The diagram above is blown up by 1.5 times for illustrative purposes

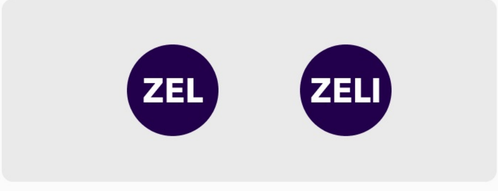
Guidance

Do



Do use 1-2 letters in initial avatars to denote names.

Don't

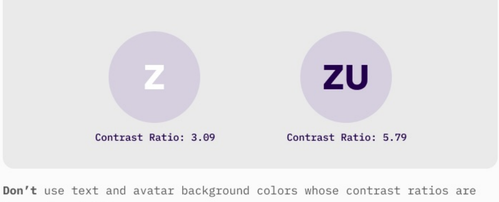


Don't use more than 2 letters in initial avatars. There is not enough negative space within the avatar boundary and the initials are meant to be read/identified quickly.

Do

Do ensure that the contrast ratio between text and avatar background exceeds 4.5:1 to meet AA WCAG standards.

Don't



Don't use text and avatar background colors whose contrast ratios are lower than 4.5:1.

zelis

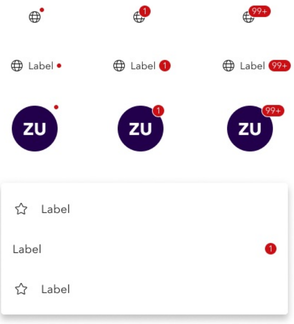
Badge

Design Kit Figma

Badges convey dynamic information, such as counts, status, or notifications. A badge can be empty or include number indicators.

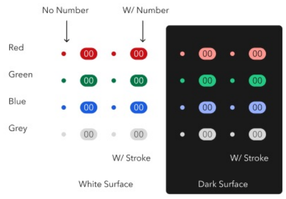
Usage

Badges are placed after the label of the thing they're quantifying, such as the number of votes for an issue. They should only be used to represent a number, however, the following letters and special characters can be used to represent number values: +, -, K.



Variations

Badges vary by two sizes differentiated by the larger size containing numbers. Badges can consume **four** different colors to indicate different statuses, while also ranging in value to support accessibility across various surfaces. Strokes may be applies to badges when a badge is overlapping content.



Specs

Badges always consume a system color, (never brand colors) to communicate statuses and notifications accurately and consistently. Strokes are used to separate the badge from the content it's laid on top of and should employ the same color as the surface the elements are on.

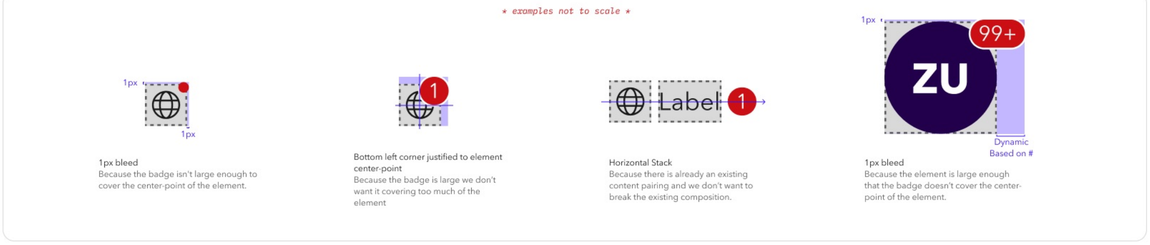
Badges are fixed sizes, do not modify the size of a badge (with the exception of dynamic width due to numeric entry).

Position

All badges should be placed either in the top right corner of an element with a 1px bleed, or center justified on the left of a horizontal element (like a list item).

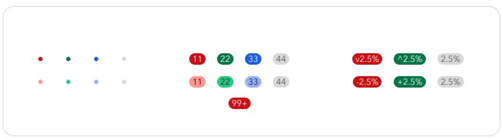
If the element is small, the badges bottom left corner should be fixed to the center point of the element it's on, allowing the badge to extend beyond the elements initial frame.

All badges should extend to the right when more numbers are included



Guidance

- All badges can be used with navigation items to convey dynamic information associated with that destination.
- Small badges use only shapes to indicate a status change or new notification.
- Large badges display numbers within a container to indicate a quantifiable status change.
- Badges with a stroke may be used to visually separate the badge from the element it's laid over.
- Letters and special characters including (+,-,%,$,v,^,k,b) may be used to give slightly more context to a badge number or used to truncate a number beyond two characters.



Color may be used to indicate the kind of status change or notification, and values may be changed depending on the given surfaces color.

- 00 Red for most status/notifications, specifically ones that're unread or to indicate a negative change.
- 00 Green for active status' or to indicate a positive change.
- 00 Blue for non-time sensitive notifications or used for informative/educational content indicators.
- 00 Grey for indicating inactive status', notification metadata for read messages or to indicate no change.

zelis

Banner: Guidance

Design Kit Figma

Banners display essential information in an easily digestible manner and supply additional context for that information.

Description

A successful banner communicates relevant information to users. Remember this principle when selecting banners to avoid the often client-driven traps of word-heavy, non-dismissible stacked banners. Follow best practices, word-count enforcement and review the notification guidelines to craft an effective banner.

The notification guidelines can help with this assessment.

Banner Quick-Facts

- Banners are part of the notification framework
- Banners can have different status levels and exist in different locations on the page
- Banners should be used sparingly, as "banner blindness" is a legitimate concern

Banner Types

The four types of banners are informational, warning, error/critical, and success/positive.
Choose a relevant banner type to match with the content.

Banner: **Informational**
Use informational banners to provide additional information that is not tied to a user's current action or task, nor is dependent on a user taking action.

This is the title of the banner. This is body copy of the banner, the core message that we are trying to get across, even in a link.

Banner: **Warning**
Warning banners are more noticeable and should only be used to communicate information that is relevant to content on the screen or could potentially impact the user's direct experience with the application. Use sparingly.

This is the title of the banner. This is body copy of the banner, the core message that we are trying to get across, even in a link.

Banner: **Error / Critical**
Error or Critical banners should only be used if there is a system-level problem that directly impacts the user's experience or to communicate important content that would severely impact a user's experience on or off the site. Excessive error/critical banners within the UI creates potentially negative perception of the application, so only use these when absolutely necessary.

This is the title of the banner. This is body copy of the banner, the core message that we are trying to get across, even in a link.

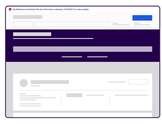
Banner: **Success / Positive**
Success or Positive banners are used to communicate that some sort of task or action was completed as intended (typically driven by a user's direct interaction with the UI).

This is the title of the banner. This is body copy of the banner, the core message that we are trying to get across, even in a link.

Banner Levels

GLOBAL BANNER

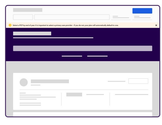
Use global banners for application or system-level issues. Global banners should never be contextual and should be used sparingly.



PAGE BANNER

Page banners communicate information relevant to all content on the page. Page banners can contain different contextual information, such as error messages or general information.

Always place page banners below the global navigation.

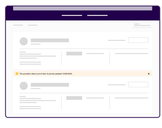


SECTIONAL BANNER

Sectional banners can be placed above any grouping of content/components on a page and are typically contextual.

INLINE BANNER

Inline banners relate to a specific piece of content or component. They should provide additional, always contextual, information to a user.



OPTIONAL: Banner Title and Close Button

Banners support having a title section (in bold), which must adhere to a reasonable length and be a high-level summary of the overall content. Banners can be dismissed and display an "x" icon.

Stacked Banners

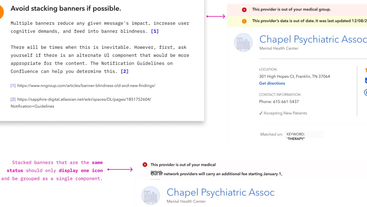
Avoid stacking banners if possible.

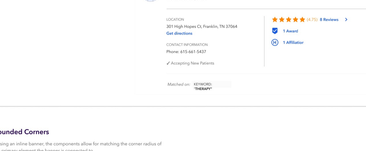
Multiple banners reduce any given message's impact, increase user cognitive demands, and feed into banner blindness. [5]

There will be times when this is inevitable. However, first, ask yourself if there is an alternate UI component that would be more appropriate for the content. The Notification Guidelines on Confluence can help you determine this. [2]

[1] https://www.nngroup.com/articles/banner-blindness-old-and-new-findings/

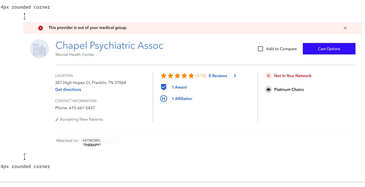
[2] https://sapphire-digital.atlassian.net/wiki/spaces/DL/pages/1851752604/Notification+Guidelines





Rounded Corners

If using an inline banner, the components allow for matching the corner radius of the primary element the banner is connected to.



Anatomy

1. **Icon:** Changes based on banner status – these are preset with defaults, but can be overridden only by Design and Product.
2. **Title:** Provides member summary of notification – can support text links.
3. **Body (optional):** Core message of the notification – can support text links.
4. **Close button (optional):** ability to close the banner.

Status Specifications

INFORMATIONAL:
Background color: Blue-50
Corner radius: 4px
Default icon: info-circle-solid
Icon color: Blue-500
Close button icon: times
Close button color: Gray-900

For the latest information on the coronavirus (COVID-19), please visit bluecross.com/coronavirus.

WARNING:
background color: **Yellow-50**
corner radius: **4px**
default icon: **exclamation-circle-solid**
icon color: **Yellow-500**
close button icon: **times**
close button color: **Gray-900**

This provider's data is out of date. It was last updated 12/08/2020.

ERROR/CRITICAL:
background color: **Red-200**
corner radius: **4px**
default icon: **times-circle-solid**
icon color: **Red-500**
close button icon: **times**
close button color: **Gray-900**

This provider is out of your medical group.

SUCCESS:
background color: **Green-50**
corner radius: **4px**
default icon: **check-circle-solid**
icon color: **Green-50**
close button icon: **times**
close button color: **Gray-900**

Successfully added as PCP.

zelis

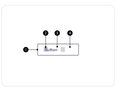
Buttons: Specs

Specs | Flat Buttons

1. Container
2. Leading Icon
3. Text Label
4. Trailing Icon

Element	Design Attribute	Token
PRIMARY		
Icon	**White** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & White** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**Primary 500**	For future
Container Fill - Hover	**Primary 900**	For future
ACCENT		
Icon	**White** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & White** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**Accent 500**	For future
Container Fill - Hover	**Accent 900**	For future
SYSTEM COLOR		
Icon	**White** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & System Blue 500** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**System Blue 500**	For future
Container Fill - Hover	**System Blue 900**	For future

Specs | Stroked Buttons



1. Container
2. Leading Icon
3. Text Label
4. Trailing Icon

Element	Design Attribute	Token
PRIMARY		
Icon	**Primary 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & Primary 500** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Stroke	**Primary 500**	For future
Container Fill	**None**	For future
Container Fill - Hover	**Primary 50**	For future
ACCENT		
Icon	**Accent 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & Accent 500** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Stroke	**Accent 500**	For future
Container Fill	**None**	For future
Container Fill - Hover	**Accent 50**	For future
SYSTEM COLOR		
Icon	**System Blue 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & System Blue 500** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Stroke	**System Blue 500**	For future
Container Fill	**None**	For future
Container Fill - Hover	**System Blue 50**	For future

Specs | Text Buttons

1. Container
2. Leading Icon
3. Text Label
4. Trailing Icon

Element	Design Attribute	Token
PRIMARY		
Icon	**Primary 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & Primary 500** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**None**	For future
Container Fill - Hover	**Primary 50**	For future
ACCENT		
Icon	**Accent 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & Accent 500** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**None**	For future
Container Fill - Hover	**Accent 50**	For future
SYSTEM COLOR		
Icon	**System Blue 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Text Label Not applicable for the icon button	**Body Small Med & System Blue 500** The text label s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**None**	For future
Container Fill - Hover	**System Blue 50**	For future

Specs | Disabled

1. Container
2. Leading Icon
3. Text Label
4. Trailing Icon

Element	Design Attribute	Token
DISABLED		
Icon	**Grey 700**	For future
Text Label	**Grey 700**	For future
Container Stroke For stroked buttons only	**Grey 700**	For future
Container Fill	**Grey 300**	For future

Specs | Elevated Icon Button

1. Container
2. Icon

Element	Design Attribute	Token
PRIMARY		
Icon	**Primary 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**White**	For future
Container Fill - Hover	**Primary 50**	For future
Elevation	**4dp**	For future
ACCENT		
Icon	**Accent 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**White**	For future
Container Fill - Hover	**Accent 50**	For future
Elevation	**4dp**	For future
SYSTEM		
Icon	**System Blue 500** The icon s/b inverse of container color to meet color contrast standards.	For future
Container Fill	**White**	For future
Container Fill - Hover	**System Blue 50**	For future
Elevation	**4dp**	For future
DISABLED		
Icon	**Grey 700**	For future
Text Label	**Grey 700**	For future
Container Fill	**Grey 300**	For future
Elevation	**None**	For future

zelis

Button

Design Kit Figma

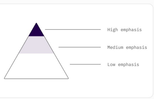
Buttons allow users to take actions, and make choices, with a single click.

Anatomy

1. Icon Label
2. Leading Icon (optional)
3. Trailing Icon (optional)
4. Container

Overview & When to Use

A button is a clickable component that triggers an action. Depending on the hierarchy and placement of the button, there is different variations of the button. Use buttons to communicate actions users can take and what will happen next. Button labels express what action will occur when the user interacts with it.



Hierarchy & emphasis

Each button variant has a particular function and its design signals that function to the user. Be mindful of the function each button variant produces and only use variants when specific user actions are required by the design.

Button Variants

Each button variant has a particular function and its design signals that function to the user. It is therefore very important that the different variants are implemented consistently across products, so that they message the correct actions.

Flat Buttons

Flat buttons are high-emphasis buttons. They have the most visual impact, and should be used to call attention to an important action. They are characterized by a fill, and can inherit client primary, accent, and system colors. Text labels can be paired with either a left or right icon (never both left and right).

1. Flat buttons
2. Flat button w/left icon
3. Flat button w/right icon

Stroked Buttons

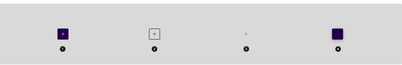
Stroked buttons are medium-emphasis buttons. They reflect actions that are important, but aren't the primary action in an app. Stroked buttons can be used in isolation but also pair well with flat buttons to indicate an alternative secondary action. They are characterized by a stroke and can inherit a client primary, accent and system colors. Text labels can be paired with either a left or right icon (never both left and right).

1. Stroked buttons
2. Stroked button w/left icon
3. Stroked button w/right icon

Text Buttons

Text buttons are low-emphasis buttons. They are typically used for less important actions. Text buttons can be paired with flat and or stroked buttons. They are characterized by a text label that can be paired with either a left or right icon (never both left and right).

1. Text buttons
2. Text button w/left icon
3. Text button w/right icon



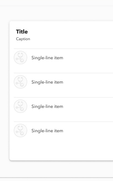
Contained Icon Buttons

Icon buttons come in flat, stroke, and text variants, in addition to offering varying degrees of elevation.

1. Flat icon button
2. Stroked icon button
3. Text icon button
4. Elevated icon button

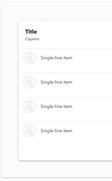
Hierarchy and placement - Do's & Don'ts

Do



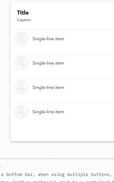
Do:
In a bottom bar, when using multiple buttons, indicate the more important action by placing it in a contained button (next to a text button).
Note this is not a real component it was created for illustrative purposes

Don't



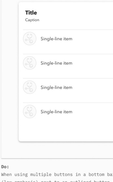
Don't:
Avoid using two contained buttons next to one another if they don't have the same fill color.
Note this is not a real component it was created for illustrative purposes

Do



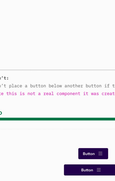
Do:
In a bottom bar, when using multiple buttons, you can place an outlined button (medium emphasis) next to a contained button (high emphasis).
Note this is not a real component it was created for illustrative purposes

Do



Do:
When using multiple buttons in a bottom bar, you can place a text button (low emphasis) next to an outlined button (medium emphasis).
Note this is not a real component it was created for illustrative purposes

Don't



Don't:
Don't place a button below another button if there is space to place them side by side.
Note this is not a real component it was created for illustrative purposes

Do

Do:
Ensure internal elements so that they scale appropriately. Icons and text labels remain centered when the button width increases.

Don't

Don't:
Don't ungroup elements inside of a button container. This can lead to unwanted movement, such as the icon and label anchoring to opposite sides of the button.

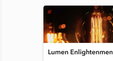
Do

Do:
Use icons that clearly communicate their meaning

Don't

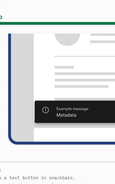
Don't:
Don't vertically align an icon and text in the center of a contained button.

Don't



Don't:
Don't use two icons on the same button.

Do



Do:
In cards text buttons help maintain an emphasis on card content.
Note this is not a real component it was created for illustrative purposes

Don't

Don't:
Avoid text labels that are too long. They should be concise.
Note this is not a real component it was created for illustrative purposes

Do

Do:
Use a text button in snackbars.
Note this is not a real component it was created for illustrative purposes

zelis

Card

Design Kit Figma

Cards are versatile containers, holding anything from images to headlines, supporting text, buttons, lists, and other components.

- Cards contain content and actions about a single subject
- The card container is the only required element in a card. All other elements shown here are optional.
- Card layouts can vary to support the types of content they contain. The following elements are commonly found among that variety.

Card Variants

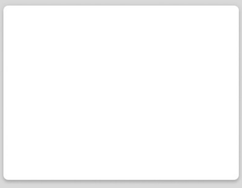
OUTLINED
(light backgrounds)

0 Elevation • 1px stroke

4dp Elevation • 0 Stroke

SIMPLE
(dark backgrounds)

0 Elevation • 0 stroke



4dp Elevation • 0 Stroke

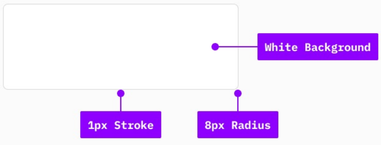
ELEVATED
(add'l emphasis)

1dp Elevation • 0 Stroke

4 dp Elevation • 0 Stroke

Specs

8px Rounded Corners
Background Color = White
Stroke (if any) = 1px, MatGray 300



Elevation

On mobile, a card's default elevation is most often 1dp, but can be 0dp to have fewer shadows on denser screens. In either case, the card's raised dragged elevation is 8dp.

On desktop, cards have a default resting elevation of 0dp and elevate to 4dp on hover.

zeis

Carousel

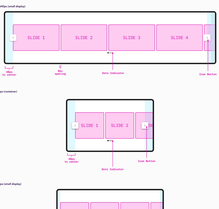
Design Kit Name

A carousel organizes a certain space by displaying only a subset of images, texts, videos, or a combination of types in a cyclic view. Give a carousel when it saves users time and clicks, aids in users' ability to compare things, and helps the users keep their place.

Anatomy

The navigational controls on a carousel suggest additional content that is not immediately visible. Carousels can be navigated manually or automatically. The content can be an extra incentive for users to browse all items at once.

Greater than 600px (small display)



Less than 600px (container)

Less than 600px (small display)

Navigation

Outer Buttons

Present when there is more content. Entering and displaying the next set of cards. Disabled or hyperlinked when the first or last card is displayed. Can be accessed contained inside the carousel.

Dots Indicators

Always present and indicate the number of pages/ sets of cards in a carousel. The element is not clickable. Can be accessed contained inside the carousel.

Content

Carousels typically use card style. A card's width and height remain consistent until hitting the specified breakpoint. All cards in a carousel display the same dimensions to ensure a smooth scrolling.

Card Sizing

Do
Make sure the dimensions are consistent for all cards in a carousel.

Don't
Allow cards to vary in their sizing.

Number of Cards

Do
Have an easily navigatable # cards within a carousel.

Don't
Add so many cards that users lose their place in the carousel.

Number of Carousels

Do
Use consistent layerings

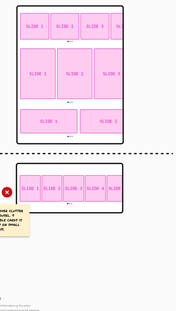
Don't
Develop carousels, that'll lose their benefits of wearability, navigation, etc

Never Cards (Center Justified)

Occasionally, carousels may have a few cards that the component can no longer display more cards. In this case, the navigational buttons and dots indicators should vanish, and you should justify the cookies cards to the screen's center.

Never Cards (Left Justified)

Occasionally, carousels may have a few cards that the component can no longer display more cards. In this case, the navigational buttons and dots indicators should vanish, and you should justify the cookies cards to the screen's left.



Never Cards (Expanding Cards)

Occasionally, carousels may have a few cards that the component can no longer display more cards. In this case, the navigational buttons and dots indicators should vanish, and you should justify the cookies cards should when expanding the screen.

Interactivity/Behavior

Users can navigate (carousels) due to the element's looping nature. Auto forwarding (carousels) serves as a function by supporting the site's purpose.

Resting Default

A static still carousel is the resting state when scrolling through the product. This state should be displayed when the user is not interacting with the carousel by right-click/tapping on the next card in the right margin.

Page Over

When a user utilizes a hinge to the next set of cards, the hinge should display the next set of cards and not the next card in the carousel. The dots indicator should also show the new corresponding dot.

Remainder Cards

Displaying the remainder is a perfect number of cards in a carousel. It's beneficial for the UX to display. The carousel will display any remaining cards while navigating from the previous set to view the remainder cards.

Navigating to the left will bring the user back to the second set of cards, and navigating to the right will get the user back to the beginning set of cards.

Possible Locks

Because of the simplest nature of the carousel, it can take on various locks. When a carousel is implemented, the benefits of these locks improve the overall user experience of the product.

Greater than 600px (small display)

SLIDE 1
SLIDE 1
SLIDE 1

zelis

Checkbox

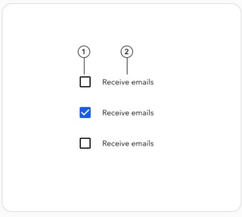
Design Kit Figma

Cards are versatile containers, holding anything from images to headlines, supporting text, buttons, lists, and other components.

Usage

Use checkboxes to:
- Select one or more options from a list
- Present a list containing sub-selections
- Turn an item on or off in a desktop environment

Anatomy



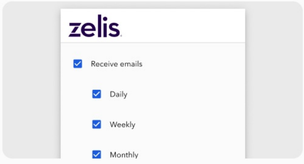
Content guidelines

- Keep checkbox labels short and descriptive.
- Start all checkbox labels with a capital letter.
- Don't include punctuation after checkbox labels.

1. Checkbox
2. Checkbox label

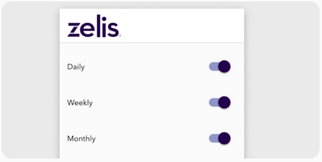
Guidance

Do



Do Checkboxes let users select one or more options from a list. A parent checkbox allows for easy selection or deselection of all items.

Don't



Don't If a list consists of multiple options, don't use switches. Instead, use checkboxes. Checkboxes imply the items are related, and take up less visual space.

Radio Button - Specs

Enabled Selected
- **Box Fill=** primary 500

Enabled UnSelected
- **Box Fill=** Grey 900

Disabled Selected
- **Box Fill=** 38% opacity of primary 500

Disabled UnSelected
- **Box Fill=** 38% opacity of Grey 900

Hover Selected
- **Box Fill=** primary 500
- **Hover Circle=** 4% opacity of primary 500
- **Cursor=** Hand

Hover UnSelected
- **Box fill=** Grey 900
- **Hover Circle=** 4% opacity of Grey 900
- **Cursor=** Hand

Focused Selected
- **Box Fill=** primary 500
- **Hover Circle=** 12% opacity of primary 500

Focused UnSelected
- **Outer Circle Fill=** Grey 900
- **Hover Circle=** 12% opacity of Grey 900

Pressed Selected
- **Box Fill=** primary 500
- **Hover Circle=** 8% opacity of primary 500

Pressed UnSelected
- **Outer Circle Fill=** Grey 900
- **Hover Circle=** 8% opacity of Grey 900

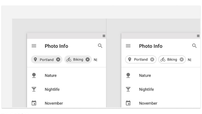
zelis

Chips

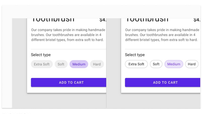
Design Kit Figma

Chips help users enter information, make selections, filter content, or trigger actions on a design. Chips are compact elements that can represent an input, attribute, or action. Use chips to show multiple interactive design elements together in the same area, like lists of items, or a series of business contacts.

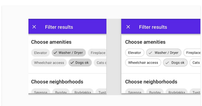
Types



Input chips
An input chip depicts complex information in compact form, such as an entity (person, place, or thing) or text. Chips enable user input and verify that input by converting text into chips.



Choice Chips
In sets that contain at least two options, choice chips represent a single selection.



Filter Chips
Filter chips represent filters for a collection of options.

Anatomy

1. Container: Chip containers hold all chip elements, and those elements determine their size. A stroke can also define a container.
2. Thumbnail (optional)
Thumbnails identify entities (like individuals) by displaying an avatar, logo, or icon.
3. Text
Chip text can be an entity name, description, tag, action, or conversational.
4. Remove icon (optional)
Input chips can include a Remove icon.

Input Chips

Input chips represent complex information in compact form, such as an entity (person, place, or thing) or text. They enable user input and verify that input by converting text into chips.

Input Chips are:
- Transformative
- Editable
- Informational
- Movable
- Multiple
- Expandable

Transformative: Input chips transform text based on user input.

1 Text transforming into an input chip. 2 Text transforming into an input chip. 3 Text transforming into an input chip.

Editable: An input chip's text is editable until the user takes an action with the chip, such as sending an email. To edit an input chip's text, tap the chip.

1 Input chips become editable when tapped. 2 Input chips become editable when tapped. 3 Input chips become editable when tapped.

Multiple: A single field can contain multiple input chips.

Movable: Input chips can be reordered or moved into other fields.

Expandable: Input chips can expand to show more information or options. A container transform transition pattern is used to reveal additional content.

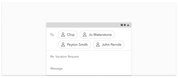
Multiple entry chips Multiple entry chips Multiple entry chips Multiple entry chips

Placement

Input chips can be integrated with other components. They can appear:

Inline with the text input cursor in a field
In a stacked list
In a horizontally scrollable list

Stacked List



If all the input chips require visibility they can wrap to a new row.

Horizontal Scroll

1 If all the input chips require visibility they can wrap to a new row. 2 If all the input chips require visibility they can wrap to a new row.

Choice Chips

Choice chips delineate and display options in a compact area. They are an excellent alternative to elements like toggle buttons, radio buttons and single-select menus.

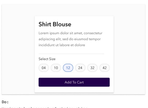
Behavior

1 Selecting a single choice chip automatically deselects all other chips in the set.

Placement

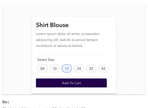
Choice chips appear as part of a series of other choice chips. They are typically displayed horizontally and sequentially.

Do



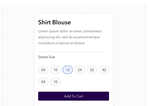
Do:
Horizontal placement of choice chips

Do



Do:
Choice chips can scroll horizontally.

Caution



Caution:
Choice chips can wrap to a new row. However, using two or more rows of choice chips can make each chip harder to scan.

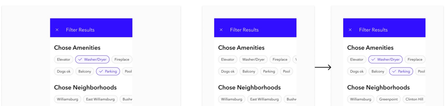
Don't

Don't:
Use choice chips to present only a single option.

Filter Chips

Filter chips use tags or descriptive words to filter content.
Filter chips delineate and display options in a compact area. They are an excellent alternative to toggle buttons or checkboxes.

Behavior

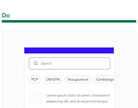


Tap a chip to select it. Multiple chips can be selected or unselected. An icon can be added to indicate when a filter chip is selected.

Filter chip suggestions can dynamically change as users start to select filters.

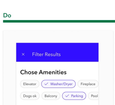
Placement

Do

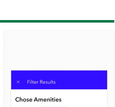


Do: Filter chips can show underneath a search field.

Do

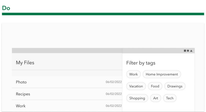


Do: Filter chips can wrap to a new row. If there are more than two rows, consider using horizontal scrolling to access all rows.



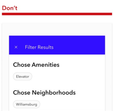
Do: Filter chips can horizontally scroll to show unlimited options.

Do



Do: Display multiple sets of filters on a side sheet.

Don't



Don't:
Filter chips should not present only a single option.

zelis

Zelis Brand Colors

Design Kit Figma

Colors can reflect the personality of our brand. They can attract users and they can better communicate your message. Any UI Design guidelines are likely to mention colors prominently. Color can set the basic mood, tone, concept and connotation for a brand or product.

See brand central

Primary Colors

NOT CREATED WITH VALUE-BASED PALETTES. Zelis is primarily a shade-based brand, leveraging a variety of shades of our Zelis Ink (and sometimes Gray/White) to achieve a tech-forward, modern look and clean aesthetic. The colors are designed to work individually, but also together to create captivating gradients and levels of depth.

Ink Blue

900	700	500	400	200	100	50
#23004B	#442666	#654D81	#9180A5	#BDB3C9	#D6CFDF	#E5E0E9
17.90:1	12.25:1	7.16:1	5.83:1	10.46:1	13.85:1	16.17:1

• Used for marketing materials in place of black in type, icons, and dark backgrounds.

Bright Blue

900	700	500	400	200	100	50
#320FFF	#5F44FB	#7057FF	#9987FF	#C2B7FF	#E6E2FF	#F7F6FE
7.92:1	5.67:1	4.67:1	7.26:1	11.51:1	16.69:1	19.57:1

• Marketing buttons
• Marking H2, H3
• Marketing graphic elements.

^ Because the secondary and neutral colors are published. Only publish the primary colors on this page ^

Secondary Colors

Pulling from the system red and yellow palettes, these secondary colors amplify the range, energy, warmth and humanity of our brand. Use them as pops in equal quantities, but not all together or adjacent in a single impression. Accent colors should be used in full saturation.

Red

900	700	500	400	200	100
#DF232A	#F0393A	#FA4F4D	#FF817A	#FFBBB7	#FFDFDA
4.76:1	5.35:1	6.29:1	8.67:1	13.04:1	16.82:1

Yellow

900	700	500	400	200	100
#F3B600	#FFC314	#FFD53A	#FFE768	#FFF6A1	#FEFDDB
11.50:1	13.07:1	14.83:1	16.88:1	18.95:1	20.29:1

Neutrals

Pulling from the system grey palette the grays, black, and white are the neutral tones in our palette and are intended to balance the bright accent colors.

900	700	500	400	200	100
#000000	#1A1A1A	#5A5A5A	#D9D9D9	#F3F3F3	#FFFFFF
21.00:1	17.40:1	6.90:1	14.88:1	18.93:1	21.00:1

zelis

System Color

Design Kit Figma

Like brand colors, system colors communicate a message. In practice, system colors offer additional context when conveying a message. Also referred to as semantic colors, these system colors act as another messaging layer available to the designer. For example, green could indicate a successful action, while yellow could serve as a callout for critical information. In all instances, follow appropriate brand and accessibility guidelines before using system colors.

See Internal Branding colors here.

Gray

The palette below is a "true gray" from which all subsequent palettes' values are based off.

• Text	• N/A	• Light dividers • Disabled layer states • Gray badge on dark background • Disabled layer states	• Hover states for list items, Expansion panel, chips	• Light backgrounds
900 #1A1A1A 17.90:1	500 #5A5A5A 7.16:1	300 #D9D9D9 13.85:1	100 #F3F3F3 18.92:1	50 #FBFBFB 16.17:1

Black & White

Apply opacity cautiously in products. Text accessibility can be easily lost when placed on surfaces in front of detailed content. Check contrast using a text color surface with the inverse opacity in front (eg. a white surface with a black opacity on it with white text on top).

White

[PREVIOUSLY White] • Card color • Background color • Text on dark backgrounds	N/A	N/A	N/A	N/A
100% #FFFFFF	24% #FFFFFF	16% #FFFFFF	12% #FFFFFF	8% #FFFFFF

[PREVIOUSLY Black] • Text on light background???	• Scrim	• Dragged layer states	• Dividers • Pressed layer state • Focused layer state	• Hover states
100% #000000	24% #000000	16% #000000	12% #000000	8% #000000

System Colors

Static colors are used for non-branded interface components. These colors should be used to communicate system-level messaging. (ex. Blue for interactivity or information, green for money or success, yellow for warnings, and red for errors or critical issues).

Blue

• Button hover	• Button • Hyperlinks • Info message icon	Snackbar CTA (aka Informational text on dark background) • Blue badge on dark background	• Banner background • Button hover
900 #0042B7 17.90:1	500 #1A5CE2 7.16:1	200 #98B1FF 13.85:1	50 #F0F3FF 16.17:1

Green

• Cost lozenge	• SmartShopper incentive badges • Cost-related text • Success message icon (currently sys 500 tho)	• Green badge on dark background	• Banner background
900 #005933 17.90:1	500 #047646 7.16:1	200 #23CC85 13.85:1	50 #C7FFE8 16.17:1

Red

• Not in use	• Error message icon	• Red badge on dark background	• Banner background
900 #9F0105 17.90:1	500 #CC0F14 7.16:1	200 #FF9690 13.85:1	50 #FFF0ED 16.17:1

Orange

• Star ratings outline + fill	• Star ratings	• Dark warning accents	• Light warning surfaces • Warning banners	• Light warning surfaces • Warning banners
900 #F15E00 17.90:1	700 #FF8600 17.90:1	500 #FFC314 7.16:1	200 #FFD559 13.85:1	50 #FEFDDB 16.17:1

zelis

Layout

Design Kit Figma

Product layouts must adapt to fit different screen sizes and form factors including mobile, tablet, foldable, and large-screen devices. When creating a layout, consider how the spacing and the parts of layout work together.

Breakpoint Scale

The responsive layout grid is primarily used to organize content and components in a layout's body region. When scaling a layout's body region to different sizes on smartphone, the responsive grid adjusts margin and fluidly widths as well as the number of columns in the layout.
Breakpoints only need to be referenced when the design needs to be changed for a particular breakpoint range, otherwise it's generally recommended to open designs for an XS and an L+ viewports.

While XS viewports have a minimum of 1px width, the minimum width we account for is 320px, therefore all mobile designs should account for a minimum screen size of 320px.

Breakpoint	Viewport (px)	Columns	Margin	Gutter	Grid Width (Recommendation in bold)
XS	1–599	4	16	8	**100% Fluid**
S	600–1023	8	32	12	**100% Fluid**
M	1024–1279	12	48	16	**100% Fluid**
L	1280–1439	12	48	16	**100% Fluid**
XL	1440–1919	12	48	16	**1440px Fixed** or 100% Fluid
XXL	1920+	12	48	16	**1440px Fixed** or 100% Fluid

Understanding Layout

The responsive layout grid is primarily used to organize content and components in a layout's body region. When scaling a layout's body region to different sizes on smartphone, the responsive grid adjusts margin and fluidly widths as well as the number of columns in the layout.

Column
One or more vertical blocks of content within a pane.

Margin
The space between the edge of the screen and any elements inside of it

Gutter
The space between columns that helps separate content

Fluid
A fluid container spans the entire width of the screen throughout all breakpoints.

Fixed
A fixed container has a maximum width set (typically 1440px) to contain designs on very large screens and are typically fluid at smaller breakpoints.

Screens

Apply layouts to your designs by creating a new frame container (or pasting one of the examples below into your workspace) and select your the appropriate Layout Grid from the right-side tool panel.
The vertical height is set to account for an anticipated viewport fold. This fold is just an estimate and it may be different per the users device.

Extra Small

Small

Medium

Large

Extra Large (Fixed*)

Extra Extra Large (Fixed*)

Extra Extra Large (Fluid)

zelis

Spacing

Design Kit Figma

Every part of a UI should be intentional including the empty space between elements. The amount of space between items creates relationships and hierarchy.

Elements in a design that are near each other are seen as being meaningfully related. More space between elements weakens their perceived relationship. Patterns created through spacing can also create relationships. Elements arranged in the same spacing pattern are seen as related pieces that have equal weight. Space can also be used to denote groups of associated information. This creates content sections on a page without having to use lines or other graphical elements as a divider.

Building on the 8px base unit, the main foundation of our spacing system is the spacing scale. This scale is a limited set of spacing values that is used to lay out UI elements in a consistent way.

All UI elements should adhere to the 8px grid while type, icons, and other small content within components may use the 4px grid. To standardize dimensions across build languages we annotate measurement specs as: px / rem or using the space token.

px = pixels rem = value relative to the base value (16px)

Pixels	REM	Token	Visual Representation	Unique Tags
0px	0rem	space-0		
4px	0.25rem	space-0.5		For small content
8px	0.5rem	space-1		
12px	0.75rem	space-1.5		For small content
16px	**1rem**	space-2		**Base Value**
24px	1.5rem	space-3		
32px	2rem	space-4		
40px	2.5rem	space-5		
48px	3rem	space-6		
56px	3.5rem	space-7		
64px	4rem	space-8		
72px	4.5rem	space-9		
80px	5rem	space-10		
88px	5.5rem	space-01		
96px	6rem	space-01		
104px	6.5rem	space-11		
112px	7rem	space-12		
120px	7.5rem	space-13		
128px	8rem	space-14		
136px	8.5rem	space-15		
144px	9rem	space-16		
152px	9.5rem	space-17		
160px	10rem	space-18		

Spacing Methods

Spacing methods are more granular than the layout grid. Spacing methods are a set of rules around how to place elements within layouts and components.

Padding

Padding is the space between elements within a component. We also specify padding in the form of margins, gutters, and other layout details.

Dimensions

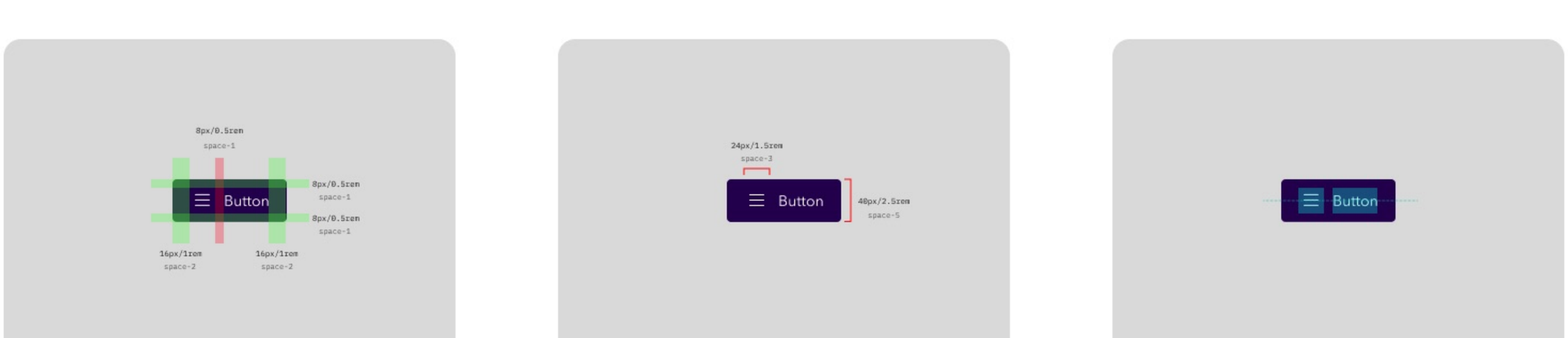
Dimensions describe the width and height of component elements.

Alignment

Alignment is the placement of elements within a component or layout

Aspect Ratios

Aspect ratios maintain consistency throughout a layout and should be applied to elements like images, surfaces and screen sizes.



zelis

Elevation

Design Kit Figma

Elevation is measured as the distance between components along the z-axis in density-independent pixels (dps).

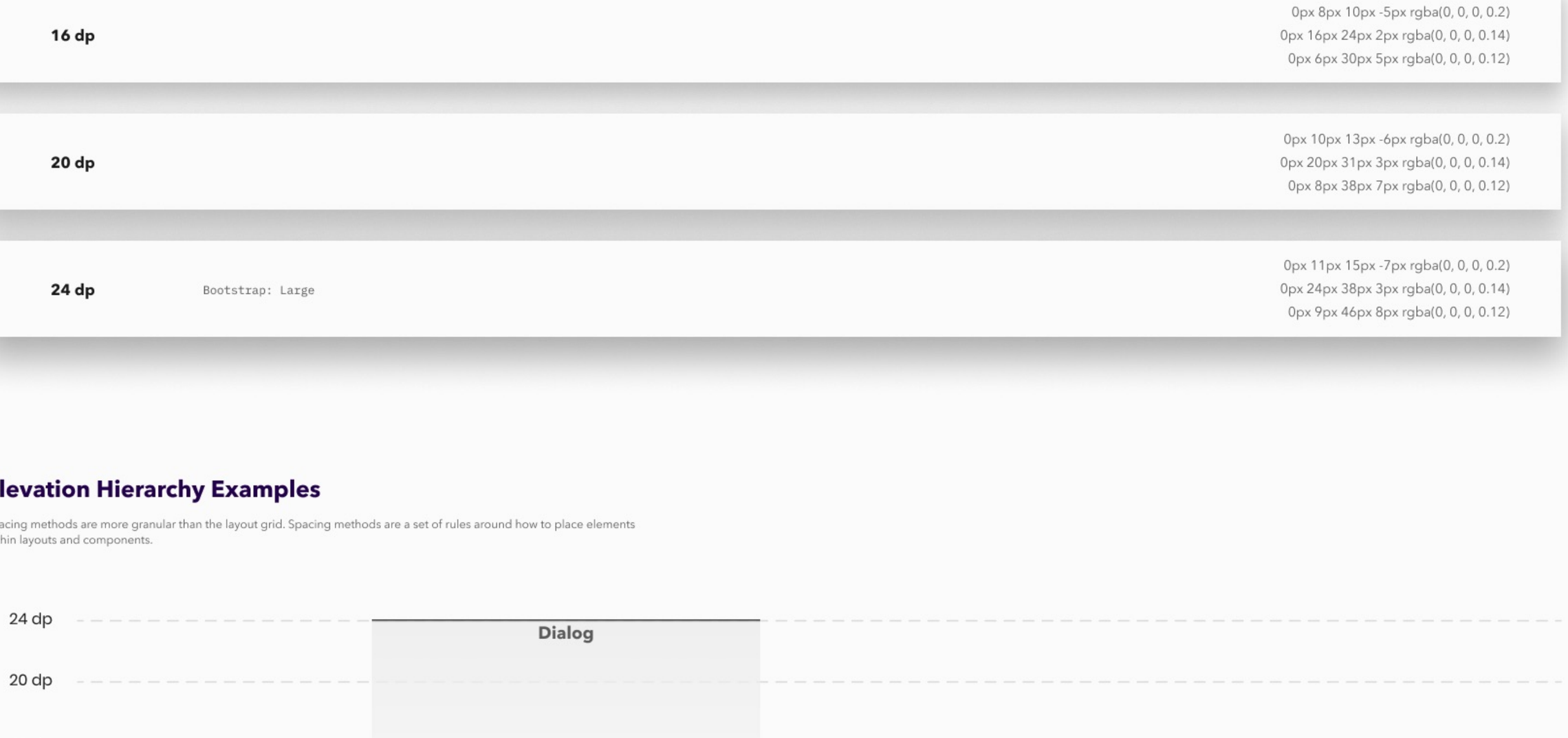
Elevation Scale

Using 'drop shadows', elevation provides important visual cues about objects' depth and directional movement. They are the only visual cue indicating the amount of separation between surfaces.

0 dp		0px 0px 0px 0px rgba(0, 0, 0, 0) 0px 0px 0px 0px rgba(0, 0, 0, 0) 0px 0px 0px 0px rgba(0, 0, 0, 0)
1 dp		0px 2px 1px -1px rgba(0, 0, 0, 0.2) 0px 1px 1px 0px rgba(0, 0, 0, 0.14) 0px 1px 3px 0px rgba(0, 0, 0, 0.12)
2 dp	`Bootstrap: Small`	0px 3px 1px -2px rgba(0, 0, 0, 0.2) 0px 2px 2px 0px rgba(0, 0, 0, 0.14) 0px 1px 5px 0px rgba(0, 0, 0, 0.12)
4 dp		0px 2px 4px -1px rgba(0, 0, 0, 0.2) 0px 4px 5px 0px rgba(0, 0, 0, 0.14) 0px 1px 10px 0px rgba(0, 0, 0, 0.12)
6 dp		0px 3px 5px -1px rgba(0, 0, 0, 0.2) 0px 6px 10px 0px rgba(0, 0, 0, 0.14) 0px 1px 18px 0px rgba(0, 0, 0, 0.12)
8 dp	`Bootstrap: Regular`	0px 5px 5px -3px rgba(0, 0, 0, 0.2) 0px 8px 10px 1px rgba(0, 0, 0, 0.14) 0px 3px 14px 2px rgba(0, 0, 0, 0.12)
12 dp		0px 7px 8px -4px rgba(0, 0, 0, 0.2) 0px 12px 17px 2px rgba(0, 0, 0, 0.14) 0px 5px 22px 4px rgba(0, 0, 0, 0.12)
16 dp		0px 8px 10px -5px rgba(0, 0, 0, 0.2) 0px 16px 24px 2px rgba(0, 0, 0, 0.14) 0px 6px 30px 5px rgba(0, 0, 0, 0.12)
20 dp		0px 10px 13px -6px rgba(0, 0, 0, 0.2) 0px 20px 31px 3px rgba(0, 0, 0, 0.14) 0px 8px 38px 7px rgba(0, 0, 0, 0.12)
24 dp	`Bootstrap: Large`	0px 11px 15px -7px rgba(0, 0, 0, 0.2) 0px 24px 38px 3px rgba(0, 0, 0, 0.14) 0px 9px 46px 8px rgba(0, 0, 0, 0.12)

Elevation Hierarchy Examples

Spacing methods are more granular than the layout grid. Spacing methods are a set of rules around how to place elements within layouts and components.



zelis

Logos

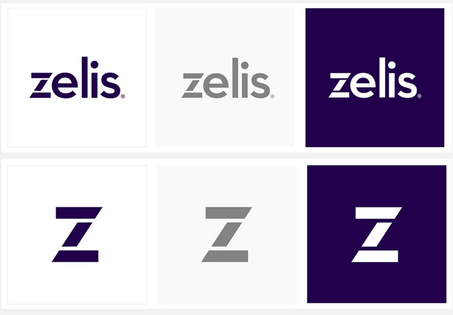
Design Kit Figma

The Zelis brand logo is clear, impactful, and deliberately minimal, ensuring that our core message and product solutions are the primary focus. Our approach is not about being flashy or self-aggrandizing; instead, it signifies a quiet assurance in our service and capabilities.

See brand central

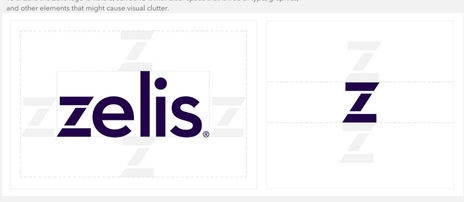
Logo design guidelines

The Zelis logo is available in three color schemes: blue, neutral, and white. The blue version works best on a white or light background. The white version should be used on dark backgrounds. The neutral version can be used when the hierarchy calls for the logo to recede, but should always be evaluated for adequate contrast.



Clearance

To ensure the Zelis logo is visible, surround it with clear space that is free of type, graphics, and other elements that might cause visual clutter.

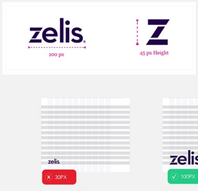


Minimum clearance

Use the brandmark "Z" of the Zelis logo to define the minimum clearance around the logo.

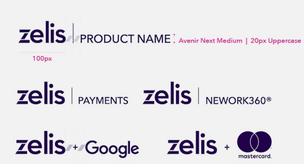
Logo sizing

As a rule, the logo should be at least 10% of the page width unless the width is <600 pixels. To maintain best accessibility practices, never size the logo smaller than 100px wide. Use the primary logo whenever possible. In rare situations when you need to use the Zelis brandmark or solo Z, please observe these minimum sizes.



Co-Branding spacing

We separate our product names with a vertical bar | and the product name is written in uppercase Avenir Next Medium 20px. Use the oblique shape from the solo Z to separate the logo, vertical bar, and product name.



Logo Crimes

To maintain the integrity of the Zelis logo and to promote the consistency of the brand, please do not misuse it. Some logo crimes are listed below.

zelis

Date Picker: Guidance

Design Kit Figma

Date pickers let users select a date or range of dates. Ensure any date pickers included are suitable for the context in which they appear in the design.

Anatomy

Date pickers provide multiple ways for users to select dates. This design element is ideal for the utility it gives users when selecting a date from a range of options (like on a calendar).

Day

Month

Year

Present

Selected
Primary 500

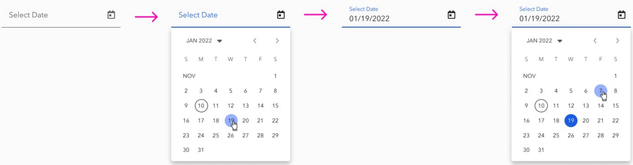
Hover
Primary 100

Date Range
Primary 50

Date range hover

Single Date Selection

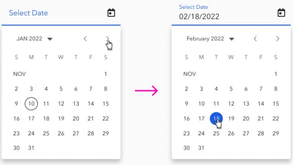
For single dates, a user can manually input a date into the field or choose to use the date picker by clicking on the calendar icon.



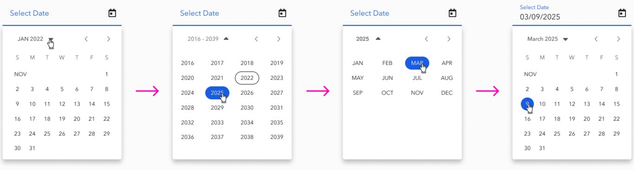
Changing Year + Month

When a date picker opens, it should display the user's current month. When users must select a date beyond the current month, they can toggle through the calendar one month at a time using the arrows located on the element. Users may also choose the month/year text button to specify the month and year they wish to select. For some use cases, starting the user on the year selection may make sense.

Month by month

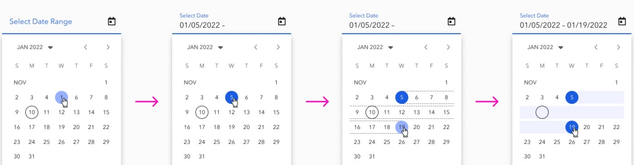


Year - month - day



Date Range Selection

When selecting a date range, users will choose a start date and then an end date.

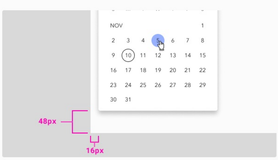


Interactivity

The placement of the date picker menu will vary depending on the location of an input form within the design.

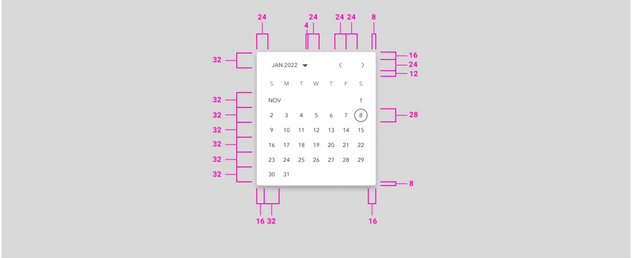
Placement

Position date pickers relative to the element that generates them and the edges of the screen or browser. While typically positioned below, date pickers can also appear in front of, beside, or above the element that generates them.



Spacing

The size of the date picker menu should never need to change.



zelis

Dialog

Design Kit Figma

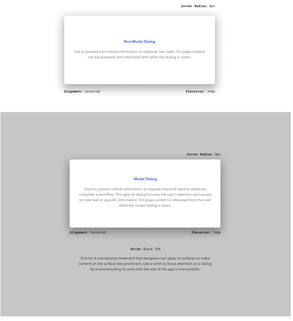
Dialog is a type of notification that appears in front of page content to provide critical information or ask users to decide on options.

Modality

Dialogs are purposefully interruptive, so they should be used sparingly. Please see the Notification Guideline for more information when deciding what notification component to use.

Full-Screen

More extensive dialogs may hit a breakpoint where the dialog changes presentation to full-screen. Full-screen dialogs should not have any content within the dialog. Still, it will affect the modality as the entire display will be filled with the dialog's content, thus nullifying the modality.



Content types

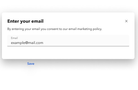
Modal and non-modal dialogues have prevent, transactional, and acknowledging classifications depending on a notification's content.

Passive

Passive modal dialogs present information users need to know about their current workflow. It contains no actions for the user to take and should not include any data that needs submission.

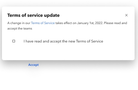
Transactional

Transactional modal dialogs validate user decisions or gain secondary confirmation. They inform users about consequences of their actions before the modal is be completed and closed. Such dialogs often have cancel and primary action buttons. Some transactional modal dialogs have multiple steps where the user will progress through consecutive screens of a given dialogue.



Acknowledging

Acknowledging modal dialogs are used when the system requires the user to acknowledge the presented information. It contains only a single button, commonly "OK". Acknowledgment is confirmed when the user clicks the primary button.



Dialog Dismissal

You may leverage two elements to indicate dialog dismissal, but it can be challenging for users to know which element to use.

✕ DON'T

Do not include the dismissal "X" icon when a "Cancel" or "Dismiss" button is in use.

✓ DO

Do prioritize the "Cancel" or "Dismiss" button over using the dismissal "X"

✓ DO

Do use the dismissal "X" when both CTAs are in use.

✓ DO

Do use the dismissal "X" when no CTAs are in use.

Sizing

Dialogs should maintain a minimum 48px distance from a screen's leading and trailing edges. Dialogs have four width sizes adhering to the 12 column grid. When a dialog reaches the maximum width for its size, it expands or contracts vertically to support content up to a 560px maximum height.

Small

Modals with short messages should use a small or default size to avoid long, single lines. Small dialogs typically only include additional content within the header/footer. Additional content can be supported in small dialogs, but may be dealing with content. Small dialogs also don't convert to full screen on smaller devices.

Default

While small dialogs don't support additional content beyond the elements provided in the header/footer - default dialogs can support other content, including forms. Default dialogs are considered the "base" dialog size for most scrolling due to a max height. Consider using the next modal size up.

Big

The big dialog will be more accommodating for complex components, like data tables or long lists.

If the big dialog height is still not enough space to effectively contain all of the content, consider using a full page.

Choosing a Size

Determining the size of a dialog can be challenging. It helps to have the content information laid out beforehand to choose a dialog size that best suits the content.

	320-767px	768-1023px	1024px+
Little Dialog	✓	✓	✓
Default Dialog	Fullscreen	✓	✓
Big Dialog	Fullscreen	Fullscreen	✓

✓ = Dialog maintains the size

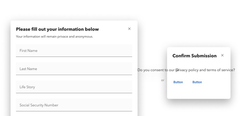
Build Considerations

When using a dialog component, there are various decisions you may choose to make along the way. Use the steps below as a guide for effectively using this component while knowing where to deviate.

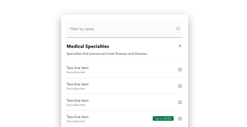
Guidance

Don't

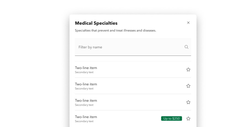
Do



Don't

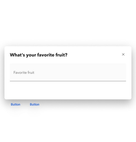


Do



Don't

Do



zelis

Divider

Design Kit Figma

Dividers are thin lines that group content in lists or other containers

Description

- Dividers should be noticeable in a layout, but not jarring.
- Dividers should only be used if elements cannot be separated using white space.
- Use dividers sparingly, to create groupings rather than separate items.

Types

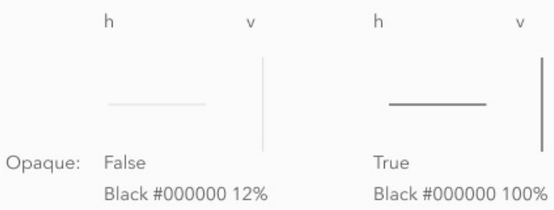
Full-bleed dividers
Full-bleed dividers separate content into sections and span the entire length of a layout.

Middle dividers
Middle dividers space related content and are centered in a layout or list.

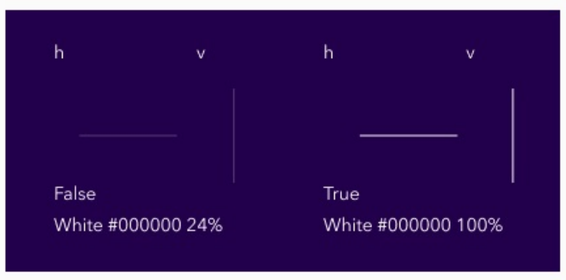
Inset dividers
Inset dividers separate related content, anchored by elements that align with the app bar title.

Thin Divider - 1px

On white



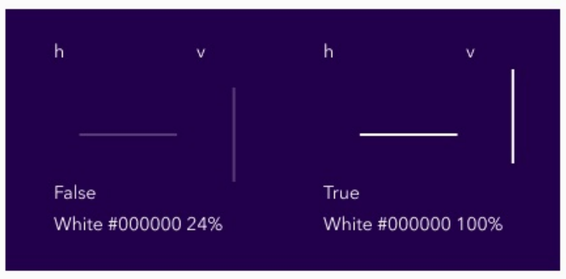
Overlay



Medium Divider - 2px

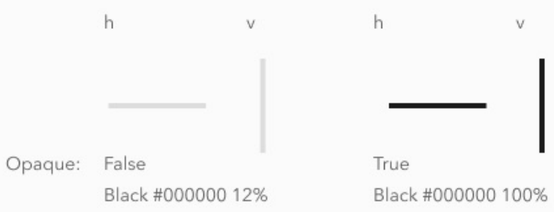
On white

Overlay

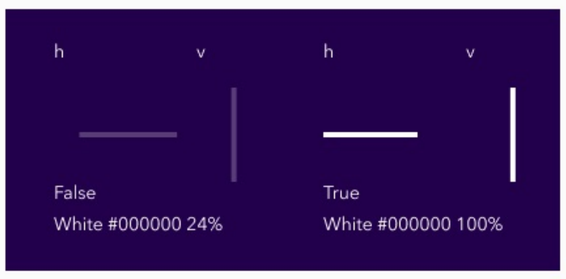


Thick Divider - 4px

On white



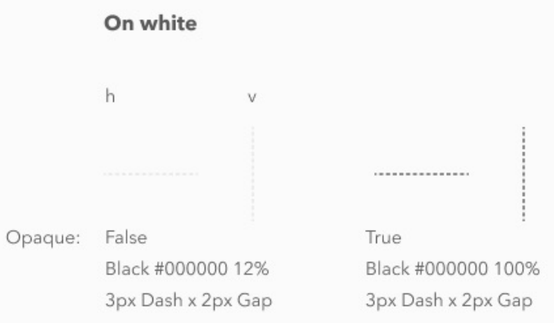
Overlay

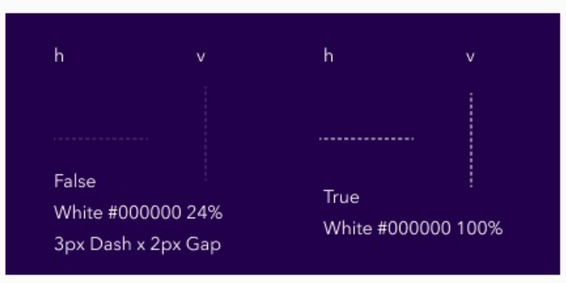


Dotted Line - 1px

Used vertically for data callouts in the cost overview layer of a provider profile

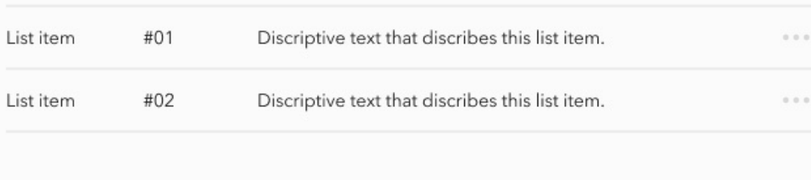
On white



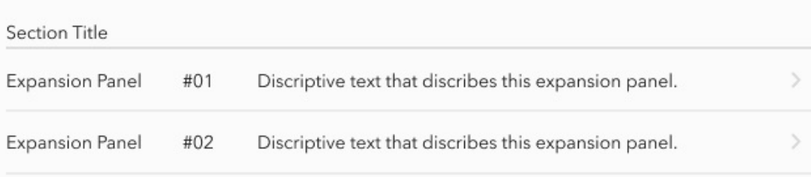


Examples

The small divider is the default size. This is used to divide similar components such as table rows, action button groups, and components within a panel.



The small divider is the default size. This is used to divide similar components such as table rows, action button groups, and components within a panel.



The large divider should only be used for page titles or section titles.

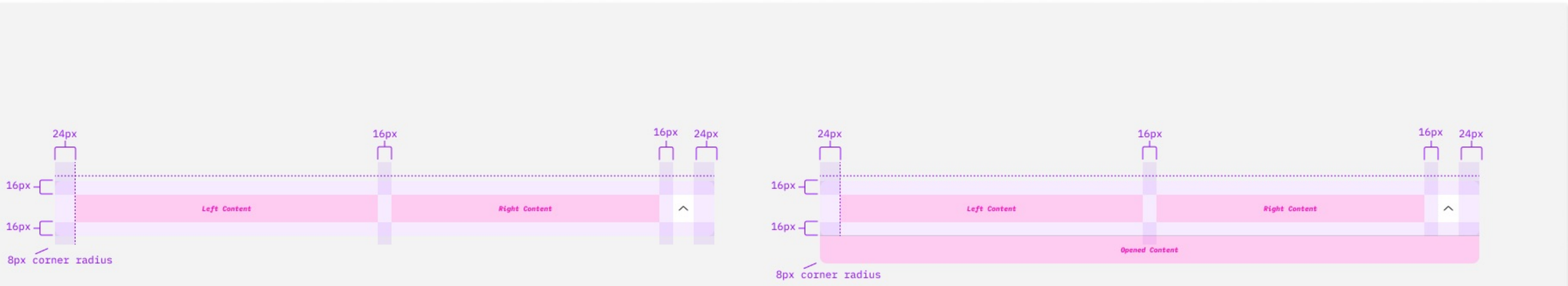
zelis

Expansion Panel

Design Kit Figma

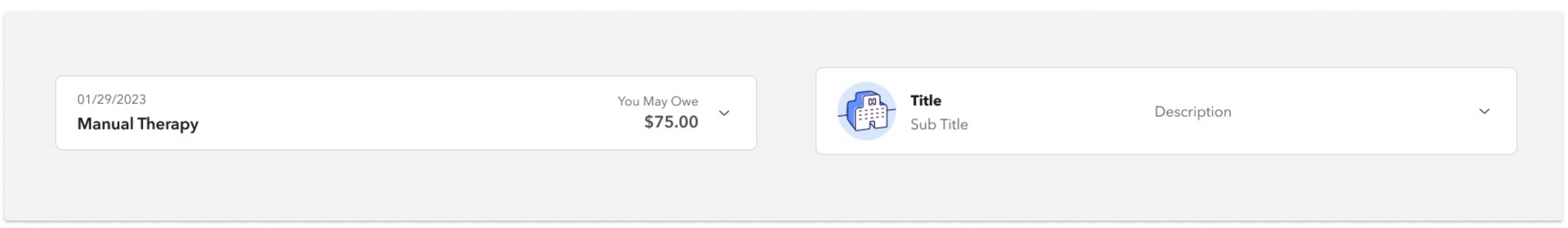
An Expansion Panel is a UI element that reveals hidden content when activated, usually by clicking a header. Its function is to save screen space and offer a cleaner user interface by showing additional information on demand.

Measurements

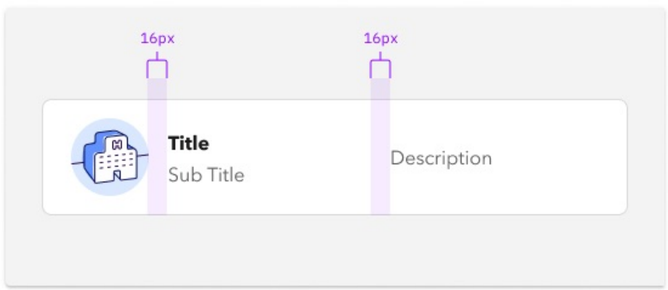


Customizations

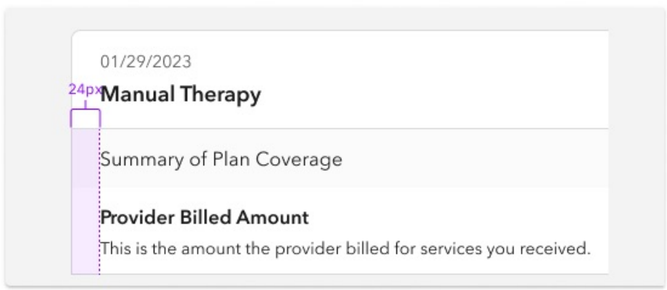
The customizable expansion panel allows various business units to create and manage their unique content. The expansion panel includes the option to add left and right content, which BUs can fully customize to meet their specific needs and preferences.



The L&R content should use 16px padding between elements

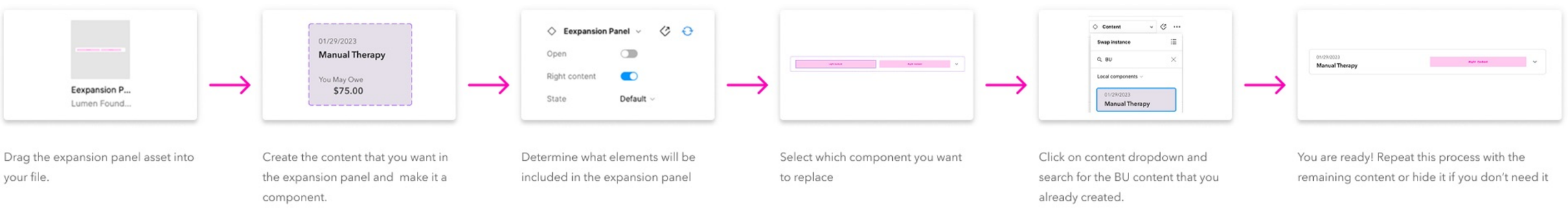


The Opened content should have a 24-margin with the exception of dividers



Build Considerations

When using an expansion panel component, there are various decisions you may choose to make along the way. Use the steps below as a guide for effectively using this component while knowing where to deviate.



Drag the expansion panel asset into your file.

Create the content that you want in the expansion panel and make it a component.

Determine what elements will be included in the expansion panel

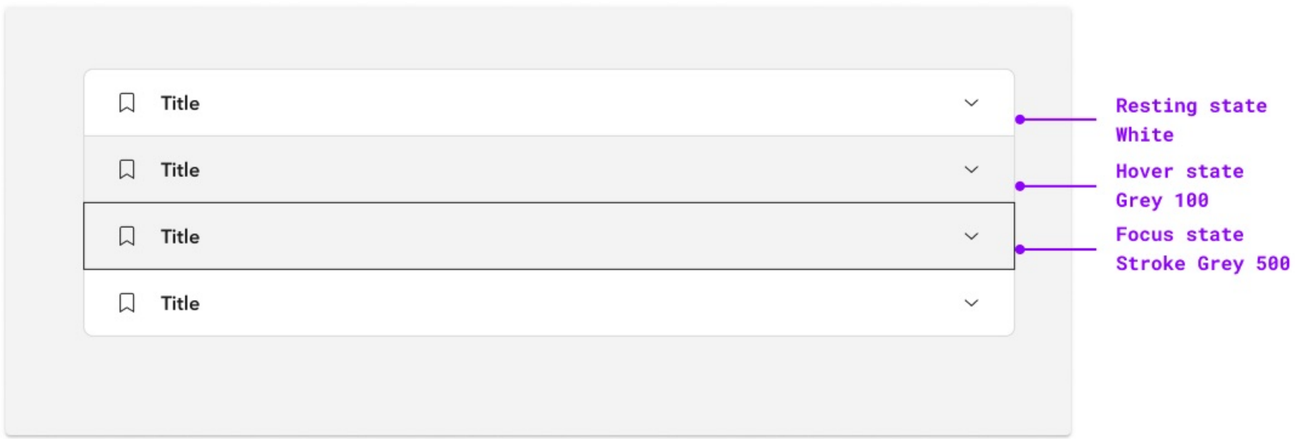
Select which component you want to replace

Click on content dropdown and search for the BU content that you already created.

You are ready! Repeat this process with the remaining content or hide it if you don't need it

Focus & State

An expansion panel uses a focus state to provide focus to individual list items. An expansion panel uses a focus state to concentrate attention on individual list items. A focus state is a centerpiece component that captures user attention. Focus state components emphasize the most significant aspect of the web page and convey the website's primary purpose.



zelis

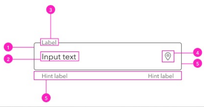
Input Field: Specifactions

Design Kit Figma

Input fields are an essential interface element that allows users to enter non-standardized information. This distinction means that input field elements can vary, so it is necessary to follow these specifications to ensure consistency across a design.

Specs - Fill and outline appearance

Padding:
1. Fill container
2. Input text
3. Label text
4. Trailing icon (optional)
5. Activation indicator (Underline)
6. Assistive elements (optional)

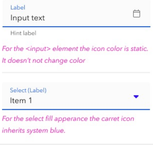


Padding:
1. Outline container
2. Input text
3. Label text
4. Trailing icon (optional)
5. Assistive elements (optional)

Fill Appearance

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Enabled**	None	Grey 50	Grey 500 ; 1 px stroke	Grey 500	None	Grey 500	Grey 500

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Hover**	None	Grey 50	Grey 900 ; 2px stroke	Grey 500	None	Grey 500	Grey 500



Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Focused**	None	Grey 50	System Blue ; 2px stroke	System Blue	Grey 900	Grey 500	Grey 700
							System blue	

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Activated**	None	Grey 50	Grey 500 ; 1px stroke	Grey 500	Grey 900	Grey 500	Grey 500

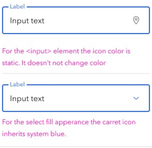
Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Text area	**Error**	None	Grey 50	Red 500 ; 2 px stroke	Red 500	Grey 900	Red 500	Red 500

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Disabled**	None	Grey 50	Grey 300	Grey 300	Grey 300	Grey 300	Grey 300

Outline Appearance

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select; Text area	**Enabled**	Grey 500	None	None	Grey 500	None	Grey 500	Grey 500

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Hover**	Grey 900	None	None	Grey 500	None	Grey 500	Grey 500



Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Focused**	System Blue ; 2px stroke	None	None	System Blue	Grey 900	Grey 500	Grey 500
							System Blue	

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Activated**	Grey 500	None	None	Grey 500	Grey 900	Grey 500	Grey 500

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Error**	System Red ; 2 px stroke	None	None	System red	Grey 900	System red	System red

Form field type	State	Container Stroke	Container Fill	Bottom Stroke	Label	Input text	Icon	Hint text
Input ; Select ; Text area	**Disabled**	Grey 300	None	None	Grey 300	Grey 300	Grey 300	Grey 300

zelis

Input Field

Design Kit Figma

Input fields are an essential interface element that allows users to enter non-standardized information. This distinction means that input field elements can vary, so it is necessary to follow these specifications to ensure consistency across a design.

Form Fields

Form Fields should be:

- **Discoverable.** They should indicate that users can input information.
- **Clear.** Form field states should be clearly differentiated from one another.
- **Efficient.** Form fields should make it easy to understand the requested information and to address any errors.

Breakdown of the Form Field Component

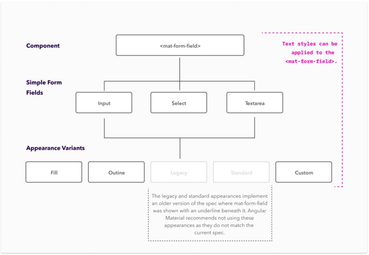
The <mat-form-field> component is used to wrap (encompase) several components such as input, textarea and, select. These are designed to work inside a <mat-form-field>.

In Angular Material the <mat-form-field> supports 4 different appearance variants such as fill, outline, legacy and standard. Morpho also includes custom form fields.

Text styles from Material such as underline, floating label and, hint messages ect. can be applied to the <mat-form-field>.

In this diagram:

- Form Field refers to the wrapper component <mat-form-field>
- Simple form fields refers to the component that the <mat-form-field> is wrapping (e.g. input, textarea and, select).
- Appearance refers to the text style applied to the <mat-form-field>



Simple Form Field

The <input> Element

The <input> HTML element is used to create interactive controls for web-based forms in order to accept data from the user.

The <mat-select> Element

The <mat-select> is a form control for selecting a value from a set of options. See the Menu section of Morpho for more detail.

The <textarea> Element

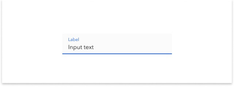
The <textarea> HTML element represents a multi-line plain-text editing control, useful when you want to allow users to enter a sizeable amount of free-form text, for example a comment on a review or feedback form.

Anatomy - Fill and outline appearance

Label Text

Label text is used to inform users as to what information is requested. Every form field should have a label.

Label text should be aligned with the input line, and always visible. Label text should move from the middle of the form field to the top (when the field is selected).



Input Text

Input text is text the user enters into a text field. A cursor indicates interactivity and the ability to type

Assistive Elements

Assistive elements provide additional detail about text entered into text fields.

1. Hint labels

Hint labels convey additional guidance about the input field, such as how it will be used. They appear below the form field's underline. It should only take up a single line, being persistently visible or visible only on focus.

A <mat-form-field> can have up to two hint labels; one start-aligned (left in an LTR language, right in RTL), and one end-aligned.

2. Character Count

Character or word counters should be used if there is a character or word limit. They display the ratio of characters used and the total character limit.

3. Error Message

When text input isn't accepted, an error message can display instructions on how to fix it. Error messages are displayed below the input line, replacing hint labels until fixed.

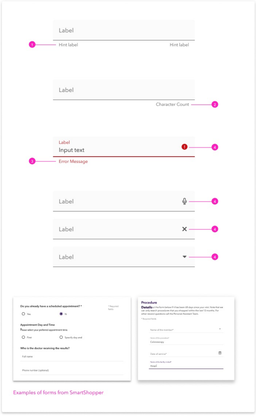
4. Icons

Form field icons can describe valid input methods (such as a microphone icon), provide affordances to access additional functionality (such as clearing the content of a field), and can express an error. Pair an error icon with an error messages to provide redundant alerts. A dropdown arrow indicates that a form field has a nested selection component.

4. Required Text Indicator

To indicate that a field is required, display an asterisk (*) next to the label text and mention near the form that asterisks indicate required fields.-

- If some fields are required, indicate all required ones
- If most fields are required, indicate optional fields by displaying the word "optional" in parentheses next to the label text
- If required text is colored, that color should also be used for the asterisk



Examples of forms from SmartShopper

Appearance Variants

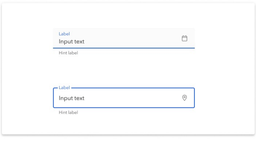
In Angular Material the <mat-form-field> supports 4 different appearance variants such as fill, outline, legacy and standard. Morpho also includes custom form fields.

1. Fill Appearance

Fill form fields have more visual emphasis than outline form fields, making them stand out when surrounded by other content and components.

2. Outline Appearance

Outline form fields have less visual emphasis than fill form fields. When they appear in places like forms, where many form fields are placed together, their reduced emphasis helps simplify the layout..



Input types

Text fields can display user input in the following ways:

- Single line text fields display only one line of text
- Multi-line text fields grow to accommodate multiple lines of text
- Text areas are fixed-height fields

Single-line fields

In single-line fields, as the cursor reaches the right field edge, text longer than the input line automatically scrolls left. Single-line fields are not suitable for collecting long responses. For those, use a multi-line text field or text area instead.

Multi-line fields

Multi-line text fields show all user input at once. Overflow text causes the text field to expand (shifting screen elements downward), and text wraps onto a new line.

These fields initially appear as single-line fields, which is useful for compact layouts that need to be able to accommodate large amounts of text.

Text areas

Text areas are taller than text fields and wrap overflow text onto a new line. They are a fixed height and scroll vertically when the cursor reaches the bottom of the field. The large initial size indicates that longer responses are possible and encouraged.

These should be used instead of multi-line fields on the web. Ensure the height of a text area fits within mobile screen sizes.

zelis

List Item

Design Kit Figma

Lists are a continuous group of text or images. They are composed of items containing primary and supplemental actions, which are represented by icons and text.

Description

Lists are a continuous group of text or images. They are composed of items containing primary and supplemental actions, which are represented by icons and text.

- Logical - Lists should be sorted in logical ways that make content easy to scan, such as alphabetical, numerical, chronological, or by user preference.
- Actionable - Lists present content in a way that makes it easy to identify a specific item in a collection and act on it.
- Consistent - Lists should present icons, text, and actions in a consistent format.

Anatomy

Depending on what your users need to get from your list UI design, the content can take various forms. Basically, list items are grouped into three main categories by order of importance as: supporting visuals, primary text and metadata.

- Supporting visuals - Supporting visuals draw attention to the list item, helping to convey basic information about the item to your users.
- Primary Text - Primary text is the main text element in a list item. It should represent the most important piece of information for your users. After this comes secondary text which usually takes the form of a subtitle and offers a slight elaboration on the primary text.

Variations

A list should be easily scannable, and any element of a list can be used to anchor and align list item content. Scannability is improved when elements (such as supporting visual and primary text) are placed in consistent locations across list items.

Primary Text - Used as the primary inforamtion for all list items.

Secondary Text - Use secondary text to add additional context to a list item when the primary text isn't sufficient.

Left Icon [Primary Action] - Used for primary action with there isn't an avatar, the left icon is descriptive or illustrative of a function.

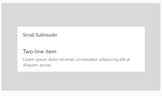
Right Icon [Secondary Action] - Used for a secondary action or primary when avatars are in use, the right icon is descriptive or illustrative of a function.

Lozenge - Lozenges are small status descriptors/visual indicators for list items.

Middle + Full Bleed Dividers - Middle dividers space related content and are centered in a layout or list. Place a middle divider between rows with lots of content, such as those with three-line lists. Full-bleed dividers separate content into sections and span the entire length of a layout. Place a full-bleed divider between list sections or when a list is not in a container.

Subheads

Subheaders delineate sections of a list. They appear on list rows. A subheader should be left-aligned with an avatar or icon in a list.



Small Subheader - Used for most lists as a way to indicate sections.

Large Subheader - Used for longer lists to improve scanability when sorting. (example: A-Z lists)

Secondary Subheader - Used to provide supporting context to right justified metadata, lozenges, and icons when needed.

Full-bleed divider - Full-bleed dividers separate content into sections and span the entire length of a layout.

Styling

Using the styles and atoms laid out throughout this design system, we have to option to levridge various styling to increase heirarchy, visual contrast, accessability, branding, etc. Here are some examples of style options:

Bold font

List Construction

Select all of the list items in your list and add an auto layout to the group.

Make sure the space between the list items and subheads is zero.

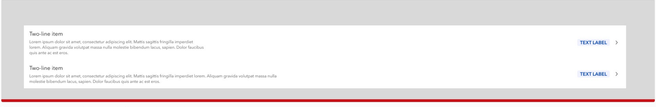
Select each individual list item + subhead and change "fixed width" to "full container".

Example

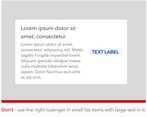
Scaling lists

Avoid excessively long lines of text when expanding containers and text-heavy components. This often means changing margins and typography properties as the container scales.

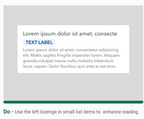
Do - adjust the amount of text per line to meet accessability standards of roughly 12 words per line or roughly 504px (31.5rem) for caption text.



Dont - allow text to run on in a single line beyond roughly 12 words or set irregular text box widths.



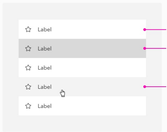
Don't - use the right lozenger in small list items with large text in it.



Do - Use the left lozenge in small list items to enhance reading

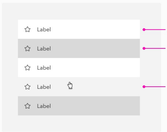
Single-Selection

All list item variations can include single select interactions. Many menus contain single select interactions. When lists allow for a selection, hover and selection states should be used as indicators for interaction. Only one selection state should be displayed at a time.



Multi-Selection

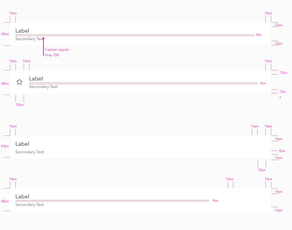
Unlike single selection, multi-selection lists are only list items with a checkbox taking the place of the left or right icon. When lists allow for a selection, hover and selection states should be used as indicators for interaction. There may be multiple selection states displayed at a time.



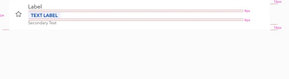
Measurements \

One line

Two line



Tree line



zelis

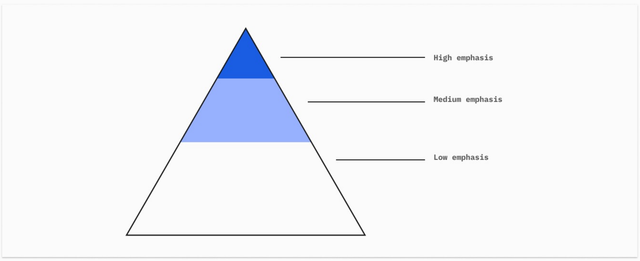
Lozenges

Design Kit Figma

A lozenge is a visual indicator used to highlight an item's status for quick recognition.

Hierarchy and Placement

Our platforms can show more than one lozenge in a layout at a time. A high-emphasis lozenge can be accompanied by a medium-and low-emphasis lozenge that performs less important actions. A lozenges level of emphasis helps determine its appearance, typography, and placement.



An app can show more than one button in a layout at a time, so a high-emphasis button can be accompanied by medium- and low-emphasis buttons that perform less important actions. When using multiple buttons, ensure the available state of one button doesn't look like the disabled state of another.button (high emphasis).

Types of Lozenges & Usage

High Emphasis Lozenges

High-emphasis lozenges communicate important information to the user. They are used to label, categorize, differentiate or highlight items using keywords that describe them.

1. PREMERA DESIGNATED PROVIDER
2. 250 250
3. PCP
4. OUT OF NETWORK

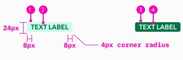
Low Emphasis Lozenges

Low-emphasis lozenges communicate communicate secondary information to the user. They are used to label, categorize, differentiate or highlight items using keywords that describe them.

1. MEMBERS SAVINGS SITE
2. KEYWORD: "PRIMARY CARE" LOCATION
3. HOME DELIVERY AVAILABLE NAME
4. NOT AVAILABLE
5. SITE OF SERVICE

Anatomy & Specs - High Emphasis Lozenges

Green lozenge

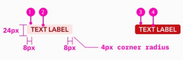


Low Emphasis
1.Container color: Green 50
2.Text label: Green 900
High Emphasis
3.Container color: Green 500
4.Text label: White100%

Blue lozenge

Low Emphasis
1.Container color: Blue 50
2.Text label: Blue 900
High Emphasis
3.Container color: Blue 500
4.Text label: White100%

Red lozenge



Low Emphasis
1.Container color: Red 50
2.Text label: Red 900
High Emphasis
3.Container color: Red 500
4.Text label: White100%

Grey lozenge

Low Emphasis
1.Container color: Grey 100
2.Text label: Grey 900
High Emphasis
3.Container color: Grey 500
4.Text label: White100%

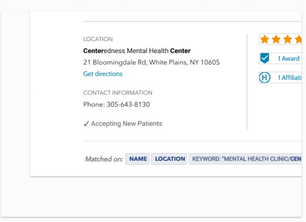
Grey lozenge

Low Emphasis
1.Container color: Yellow 200
2.Text label: Grey 900
NO High Emphasis

Styling

Text Label Font & Weight:

Depending on the context and significance of a lozenge, font weights (regular and medium) can be used together. Furthermore, when multiple lozenges are being used together, the type case (lower and uppercase) can be used together. This can be used to emphasize content and create hierarchy. Use this option only as an exception and after discussion with team members.



zelis

Menu

Design Kit Figma

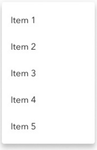
A menu displays a list of choices on a temporary surface. The options appear when users interact with a button, action, or other control element.

Anatomy

A menu's options can be related and grouped and include unrelated options, too.

Dropdown menus

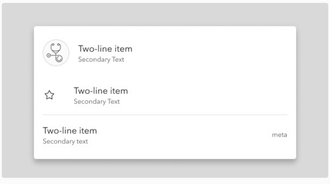
Dropdown menus display a list of options, triggered by an icon, button, or action. Their placement varies based on the element that opens them.



- **Elevation** - 8dp
- **Card** - 4px corner radius
- **Card** - White
- **List item** - Any variation

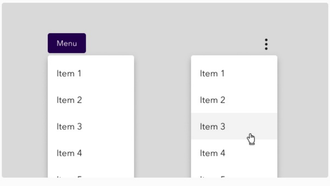
Content: List Items

Menus leverage list items to populate the content within them for best scannability. List items are highly variable within a menu, and item types depend on the intended user interaction and information the menu conveys.



Triggers

Menus appear when users select an interactive UI element, such as an icon, button, action, or content, such as selected items or text.

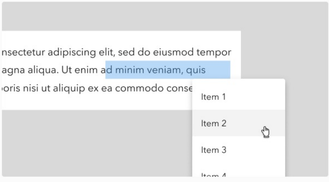


Button

Filled buttons, text links, and clickable icons can all trigger dropdown menus. These menues often include single selection list items that will navigate the user to another screen.

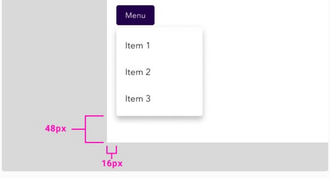
Contextual

Contextual menus are triggered by right clicking on a non-interactive element like text or the background layer.



Interactivity

A menu's content and desired user interaction allow some variations for more flexibility.

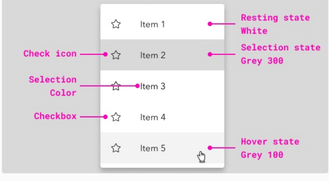


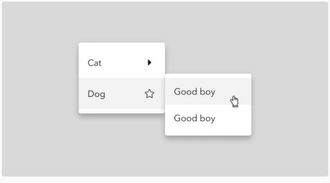
Placement

Position menus relative to the element that generates them and the edges of the screen or browser. Typically positioned below, menus can also appear in front of, besides, or above the element that generates them.

Selection Indication

Content determines how to signify a selection. However, regardless of content, use a hover and selection gray for the list item background. Use color and iconography as supporting indicators.



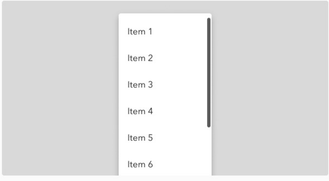


Cascading Menu

Cascading menus allow users to choose from various options by displaying menus with multiple hierarchy levels. Cascading menus comprise parent list items with subordinate child list items. Depending on the design, child list items may appear to the right or left of parent list items.

Scrolling

A scrollable menu allows users to interact with a persistent scroll bar to view and select menu options. A scrollable menu loses scroll functionality if all menu options are selected simultaneously.



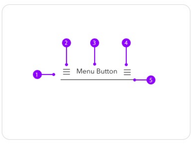
zelis

Nav Buttons

Design Kit Figma

Navigation Buttons are design elements that function as navigation tools for users. Navigation buttons vary, including horizontal bars, dropdown menus, hamburger navigation, vertical sidebars, and foot menus.

Specs | Spacing



1. Container
2. Leading Icon
3. Text Label
4. Trailing Icon
5. Underline

Specs | States

*States apply to all variants

	RESTING button with no interaction	HOVER cursor over the button	SELECTED clicked button	PRESSED process of clicking button/user tap
Default Button	Nav Button	Nav Button	Nav Button	Nav Button
Button with Icons	Nav Button	Nav Button	Nav Button	Nav Button
		if expandable, caret will rotate 180° on hover		fast "loading bar" behavior, looks like quickly fading the Gray 300 into the Gray900

Specs | Styles

RESTING STATE

Menu Button

Menu Button

Container Fill	Transparent
Container Corner Radius	0px
Text + Icon Color	Gray 10
Text Label Type	Body Small Medium

HOVER STATE

Menu Button

Menu Button

Container Fill	Transparent
Container Corner Radius	0px
Text + Icon Color	Gray 10
Text Label Type	Body Small Medium
Underline Stroke	1px
Underline Color	Black 24% Opacity

SELECTED + FOCUS STATE

Menu Button

Menu Button

Container Fill	Transparent
Container Corner Radius	0px
Text + Icon Color	Gray 10
Text Label Type	Body Small Medium
Underline Stroke	1px
Underline Color	Gray 10

RESTING STATE

Menu Button

Menu Button

Container Fill	Transparent
Container Corner Radius	0px
Text + Icon Color	White
Text Label Type	Body Small Medium

HOVER STATE

Menu Button

Menu Button

Container Fill	Transparent
Container Corner Radius	0px
Text + Icon Color	White
Text Label Type	Body Small Medium
Underline Stroke	1px
Underline Color	White 40% Opacity

SELECTED + FOCUS STATE

Menu Button

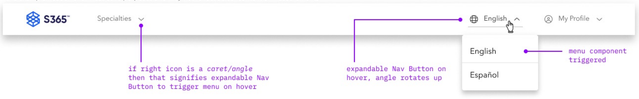
Menu Button

Container Fill	Transparent
Container Corner Radius	0px
Text + Icon Color	White
Text Label Type	Body Small Medium
Underline Stroke	1px
Underline Color	White

Guidance | Dropdown Behavior

For Global Navigation, it is best practice to automatically trigger menus for expandable Nav Buttons on hover. This allows users to reduce the number of clicks performed to find the intended task's path.

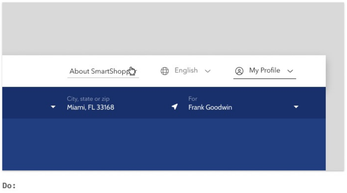
*this example header is for illustrative purposes and may not reflect actual content or measurements



Guidance | Usage

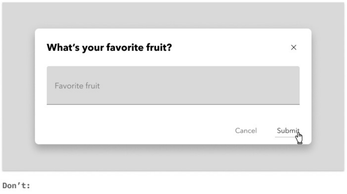
The intended use for Nav Buttons is to **act as menu item components** within the S365 Global Navigation Header. If, in the future, there is a need for Nav Buttons to be used in a capacity that is not related to navigation menus, please revisit the guidelines below and edit them as needed.

Do



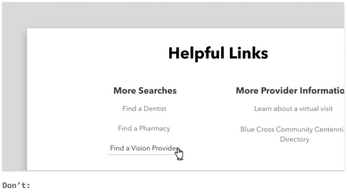
Do:
Use Nav Buttons as menu items within a larger navigation component.
*This is not a real component

Don't



Don't:
Do not use Nav Buttons to replace Text Buttons. Only use Nav Buttons in the context of a larger navigational element.

Don't



Don't:
Do not use Nav Buttons to replace Text Buttons. Only use Nav Buttons in the context of a larger navigational element.

zelis

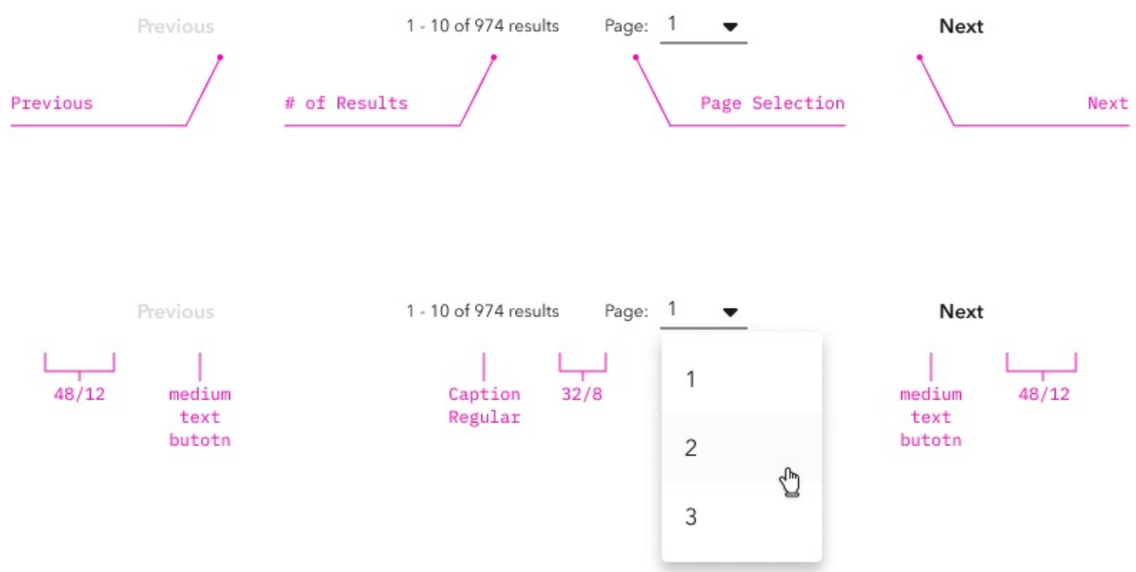
Pagination

Design Kit Figma

Steppers display progress through a sequence of logical steps. They may also be used for navigation.
Steppers may display a transient feedback message after a step is saved.

Anatomy

A stepper should be used as a subtle indicator of what step or a process the user is on. They may also be used for carouselling content like an image gallery or subsequent slides.



Sizing

When a device's width conflicts with established breakpoints, the paginator will adjust to include the screen's primary information.

Page: 1

Reduced side padding from 48px/12 to 16px/4
Removal of Prev./Next text and #of results text

Mobile: 640px

Previous 1 - 10 of 974 results Page: 1 Next

Desktop: <768px

Behavior

zelis

Progress Indicators

Design Kit Figma

Progress indicators display to users the progress of an ongoing task or operation, such as file uploads, downloads, form submissions, or data processing. They provide feedback to users, letting them know that something is happening and how long action takes to complete

Determinate and Indeterminate

Progress spinners may be determinate or indeterminate.
As more information about a process becomes available, a progress indicator can switch from an indeterminate to a determinate state.

Determinate

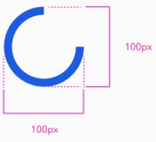
Determinate progress spinners fill the invisible, circular track with color, from 0% to 100%.

Indeterminate Indicators

Indeterminate progress spinners grow and shrink in size while moving along the invisible track. This type indicates that something is happening without specifying the completion time.

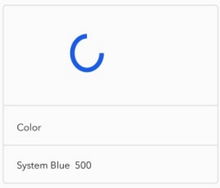
Indeterminate Progress Spinner

Indeterminate circular indicators grow and shrink in size while moving along the invisible track.

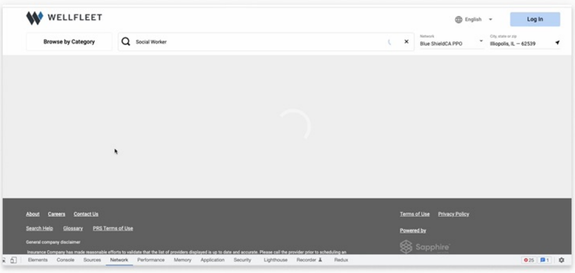


Indeterminate Progress Spinner

Do



Don't



Placement

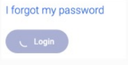
Progress spinners are positioned to indicate the process function that they represent.

- When centered on the screen, they indicate the initial loading of screen content.
- When placed above or below existing content, they draw attention to where new content will appear.

Integrating with actions

A progress spinner can be integrated into a button or actionable icon to express a connection between an interaction and a specific item. They are typically used to express when an interaction, such as clicking a button again, isn't available.

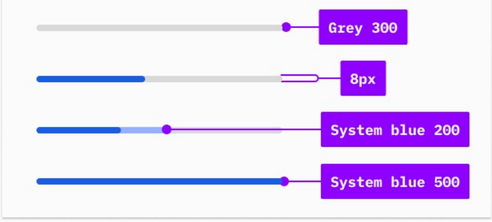
They should be used for short, indeterminate activities (between 2-5 seconds). Longer activities may require alternate methods of communication, such as snackbars or notifications.



See icons for progress spinner icon

Two-Color (Buffer) Progress Bar

A two-color bar is used to show a task's percentage of completeness. This bar does not move like the intermediate progress indicators.



zelis

Radio Button

Design Kit Figma

A radio button is an element used to select only one choice among multiple choices. This functionality results in radio button groupings called radio groups, where only one radio button in a group is selectable by users to complete an action or select an option.

Radio Button Usage

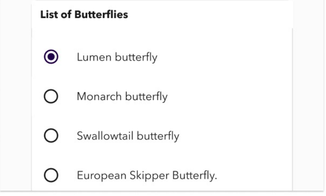
Use radio buttons to:
- Select a single option from a list
- Expose all available options

If available options can be collapsed, consider using a dropdown menu instead, as it uses less space.

Usage - Do's & Dont's

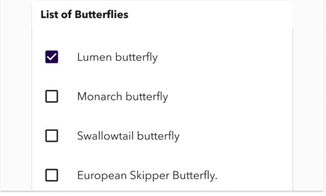
Radio buttons should be used instead of checkboxes if only one item can be selected from a list.

Do



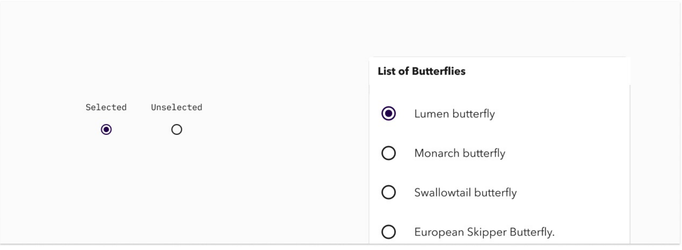
Do:
Use radio buttons when only one item can be selected from a list.
Note this is not a real component it was created for illustrative purposes

Don't



Don't:
Don't use checkboxes when only one item can be selected from a list. Use radio buttons instead.
Note this is not a real component it was created for illustrative purposes

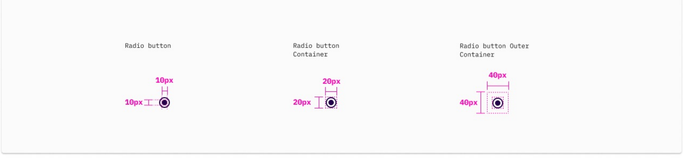
Radio Buttons - Behavior



Selected and unselected radio buttons.

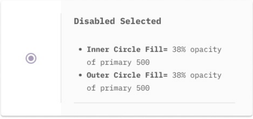
Radio buttons allow the user to select one option from a set. Radio buttons allow users to select a single option for a single option from a visible list.
Note this is not a real component it was created for illustrative purposes.

Radio Button - Specs



Enabled Selected
- **Inner Circle Fill=** primary 500
- **Outer Circle Fill=** primary 500

Enabled UnSelected
- **Outer Circle Fill=** Grey 900



Disabled Selected
- **Inner Circle Fill=** 38% opacity of primary 500
- **Outer Circle Fill=** 38% opacity of primary 500

Disabled UnSelected
- **Outer Circle Fill=** 38% opacity of Grey 900

Hover Selected
- **Inner Circle Fill=** primary 500
- **Outer Circle Fill=** primary 500
- **Hover Circle=** 4% opacity of primary 500
- **Cursor=** Hand

Hover UnSelected
- **Outer Circle Fill=** Grey 900
- **Hover Circle=** 4% opacity of Grey 900
- **Cursor=** Hand

Focused Selected
- **Inner Circle Fill=** primary 500
- **Outer Circle Fill=** primary 500
- **Hover Circle=** 12% opacity of primary 500

Focused UnSelected
- **Outer Circle Fill=** Grey 900
- **Hover Circle=** 12% opacity of Grey 900

Pressed Selected
- **Inner Circle Fill=** primary 500
- **Outer Circle Fill=** primary 500
- **Hover Circle=** 8% opacity of primary 500

Pressed UnSelected
- **Outer Circle Fill=** Grey 900
- **Hover Circle=** 8% opacity of Grey 900

zelis

Scrim

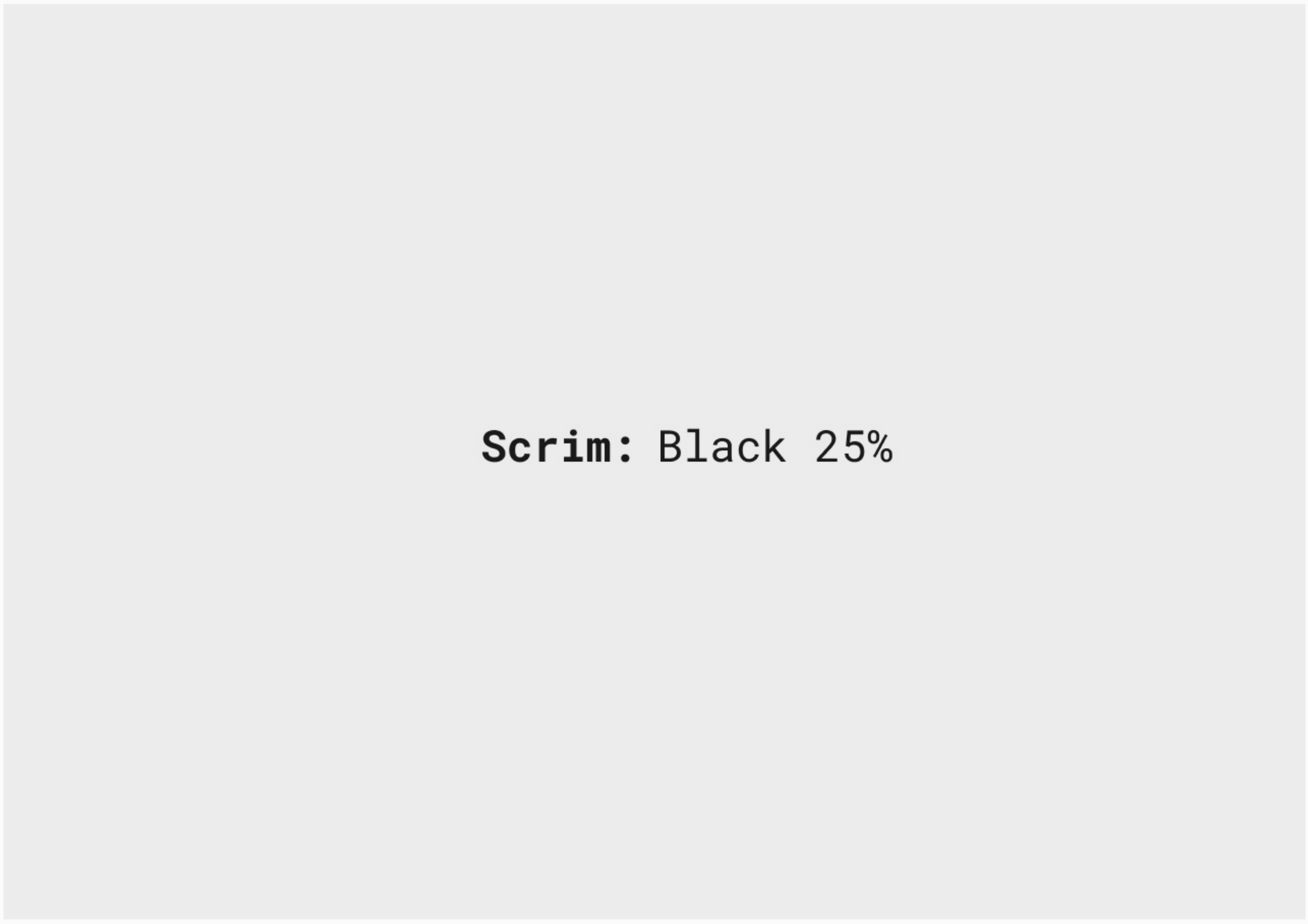
Design Kit Figma

A scrim component is a user interface (UI) element often used to create a semi-transparent overlay or backdrop that can be placed on top of other content to draw attention to a specific part of the UI or to provide a modal or popup-like experience. It is typically used to improve the visibility and focus of a particular section of a web or mobile application.

About

A scrim is used to express that the rest of the app is inaccessible and to focus attention on components like dialogs, and the search bar. The surfaces behind these components are scrimmed. A scrim is a temporary treatment that can be applied to surfaces for the purpose of making content on the surface less prominent.

Desktop



Mobile

zelis

Search Bar: **Guidance**

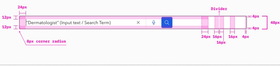
Design Kit Figma

Search allows users to locate content across an app quickly. Basic search lets users input a query into a search text field to view related results.
You can include search query input methods to have historical suggestions, auto-completion of queries, and voice input.

Anatomy & Choosing a Size

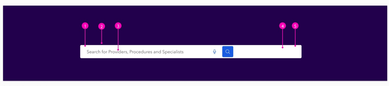
Elements that make up a search bar (
1.Container
2.Placeholder text
3.Text Cursor: When the user clicks in the search bar the cursor blinks to indicate interactivity.
4.Input text
5.Clear Icon: generates when the user inputs text
6.Divider: generates with the microphone icon
7.Microphone Icon: currently not available build for future consideration
8.Search button

Specs

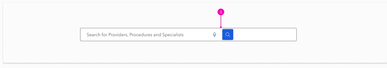


Default State

- As a default state, the search bar is resting with a search text field inside of the container, ready to receive focus.



1.Container: White
2.Container stroke: Grey 300 ; Stroke weight=1px
3.Placeholder text: Big body regular ; Grey 700
4.Microphone Icon: System blue
5.Search Icon Button: Flat blue button is client configurable.



The light ground search bar has the same elements and UI design as the search bar on dark ground, the difference being:

1.Container Stroke: grey 700 ; Stroke weight=1px

Default State Mobile and breakpoint

- There are two default search bars on mobile. One lives on the home screen 'default home search bar', and the other on the search screen 'default search bar'. The placeholder text and its behavior changes depending on the location.
  - Default home screen: The placeholder text on the mobile home screen says "Search".
  - Default search screen: An animation is applied to the placeholder text on the mobile search screen. The text cycles (every 3 seconds) between "Search for Providers" ; "Search for Procedures" ; "Search for Specialists".

Default Home Screen Search

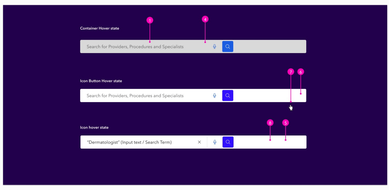
1.Container: White
2.Container stroke: Grey 300 ; Stroke weight=1px
3.Placeholder text: Body small regular ; Grey 700
4.Microphone Icon: System blue (currently we don't have this capability, therefore icon shouldn't display)
5.Search icon: Grey 900

Default Search Screen

1.Container: White
2.Container stroke: Grey 700 ; Stroke weight=1px
3.Placeholder text: Body small regular ; Grey 900. An animation is applied to the placeholder text on the mobile search screen. The text cycles (every 3 seconds) between "Search for Providers" ; "Search for Procedures" ; "Search for Specialists".
4.Microphone Icon: System blue (currently we don't have this capability)
5.Arrow Icon: Grey 900. The arrow icon replaces the search icon

Hover State

- Container Hover State: The hover state communicates that the search bar is an interactive element. The hover state occurs when a user has placed a cursor above the search bar. The hover state is an overlay on a dark ground, and the stroke inherits a darker color stroke on a light ground. The hover state applies to the entire component.
- Icon Button Hover State: The search button within the search bar has a hover state. The hover state occurs when a user has placed a cursor above the search button. It communicates that the search button is an interactive element.
- Clear and Microphone Icon Button: First, we want to see these with the default hover state in the code. However, note that this is something we may decide to remove.



1.Container upon Hover: Grey 200 fill ; stroke 300 ; Stroke weight=1px
2.Container Stroke: Grey 300
3.Placeholder text: Big body regular ; Grey 700
4.Cursor upon container hover: Text cursor
5."Microphone" Icon: No hover state
6.Search Icon Button: Flat accent button is client configurable. The button inherits a hover state when user hovers over the button - accent 900. The button's hover state is independent of the container hover state.
7.Cursor upon hover over button: hand cursor
8."Clear" HTML button: No hover state

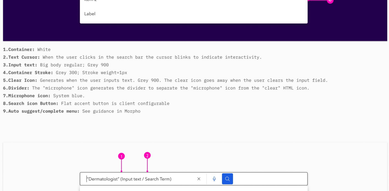
The light ground search bar has the same elements and UI design as the search bar on dark ground, the difference being:

1.Container Stroke:grey 900 ; Stroke weight=1px
2.Container Elevation: 04b elevation

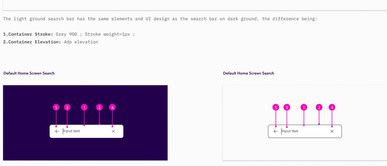
Input State

- Users can type a query in the search bar or select from the autocomplete menu. The autocomplete menu generates a list of relevant searches.
- When the user inputs a query, the "clear" icon populates.
- A divider generates the "microphone" icon (when voice will be an option). This divider separates the "clear" and microphone " icons.
- The input text takes on a darker color.
- The search submits when pressing return/enter.

Autocomplete menus are usually closable via the keyboard interaction (Esc key). When the autocomplete menu has been closed, but the search bar input is still in focus, there are very few visual indicators for the user. To assist the user, include a stroke to the search bar on the focus state to reinforce visual cues for the user.



1.Container: White
2.Text Cursor: When the user clicks in the search bar the cursor blinks to indicate interactivity.
3.Input text: Big body regular; Grey 900
4.Container Stoke: Grey 300; Stroke weight=1px
5.Clear Icon: Generates when the user inputs text. Grey 900. The clear icon goes away when the user clears the input field.
6.Divider: The "microphone" icon generates the divider to separate the "microphone" icon from the "clear" HTML icon.
7.Microphone Icon: System blue.
8.Search Icon Button: Flat accent button is client configurable
9.Auto suggest/complete menu: See guidance in Mozpha



The light ground search bar has the same elements and UI design as the search bar on dark ground, the difference being:

1.Container Stoke: Grey 900 ; Stroke weight=1px ;
2.Container Elevation: 44b elevation

Default Home Screen Search

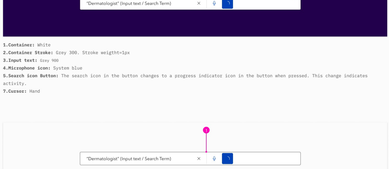
1.Container: White
2.Container stoke: Grey 300 ; Stroke weight=1px
3.Placeholder text: Body small regular ; Grey 700
4.Time Icon: Grey 900. The times icon replaces the microphone icon when the user enters a search term.
5.Arrow Icon: Grey 900. The arrow icon replaces the search icon

Default Home Screen Search

1.Container: White
2.Container stoke: Grey 900 ; Stroke weight=1px
3.Placeholder text: Body small regular ; Grey 900
4.Time Icon: Grey 900. The times icon replaces the microphone icon when the user enters a search term.
5.Arrow Icon: Grey 900. The arrow icon replaces the search icon

Pressed State

- Upon pressing the button, the search is submitted.
- A pressed state communicates a user-initiated tap or click by a cursor, keyboard, or voice input method. The button within the search bar maintains a pressed state.
- Upon pressing the button, the search icon changes to a progress indicator. This change shows the user that the request is confirmed.
- Upon pressing the button, the search submits.



1.Container: White
2.Container Stroke: Grey 300. Stroke weight=1px
3.Input text: Gray 900
4.Microphone Icon: System blue
5.Search Icon Button: The search icon in the button changes to a progress indicator icon in the button when pressed. This change indicates activity.
7.Cursor: Hand

The light ground search bar has the same elements and UI design as the search bar on dark ground, the difference being:

1.Container Stroke: grey 700 ; Stroke weight=1px

zelis

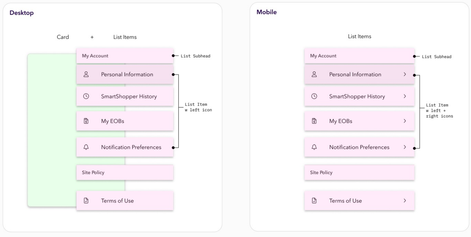
Side Navigation

Design Kit Figma

Side navigation allows users to review all of the content for a product or section. You can use side navigation for single or multi-level navigation.

Anatomy

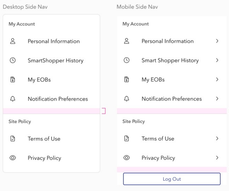
The Side Navigation component comprises a list of items stacked vertically on each other. In the desktop view, the list of items lies on top of an outlined card. In mobile, the list items lie directly in the background.



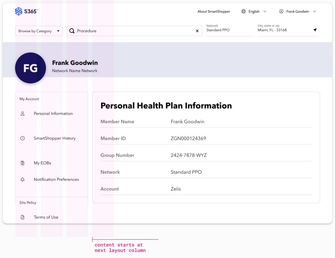
Spacing

Follow these rules for spacing between elements within the navigation and outside of the navigation.

Spacing In Side Nav



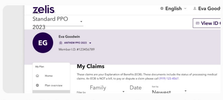
Spacing Around Side Nav



Usage

Side Navigation helps situate users in the site's architecture and allows users to jump from one page to another quickly. Use the Side Navigation Component when there are many pages (not necessarily closely related) that users need to be able to switch between.

Do



Do use the Side Navigation Component as the **primary navigation** or the **secondary navigation** if the primary navigation is horizontal.

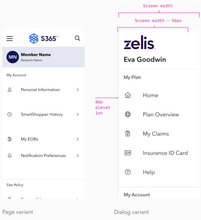
Don't

Don't use the Side Navigation Component as the following component if the previous navigation already utilizes the Side Navigation Component. Example: Don't implement secondary and tertiary navigations as the side navigation component.

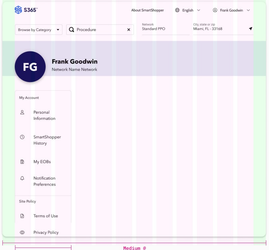
Sizing + Layout

As a general rule, the Side Navigation should take up the width of the first three columns.
*All mockups are shown in 75% of their original sizes.

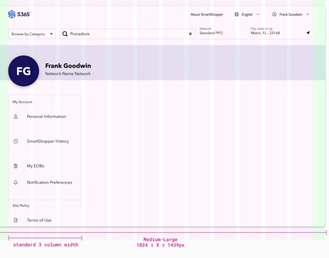
Mobile Component



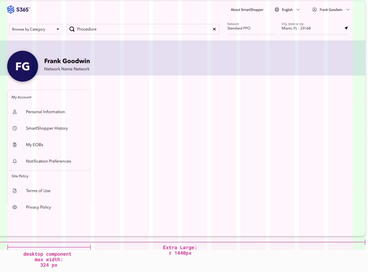
Desktop Component | Minimum Width



Desktop Component | Standard Width



Desktop Component | Max Width



zelis

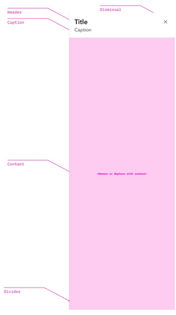
Side Sheet

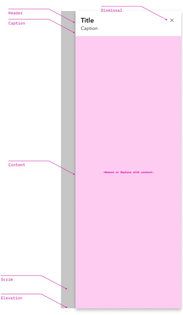
Design Kit Figma

Side sheets are surfaces containing supplementary content anchored to the left or right edge of the screen.

Anatomy

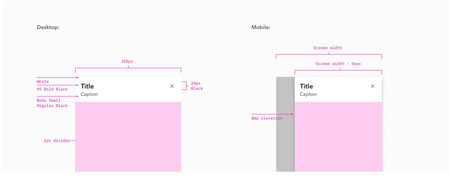
Side sheets are anchored to the right edge of the screen. They are a fixed width and typically span the height of the screen or content container they're referencing.





Sizing + Spacing

Note that the width always stays consistent while sheet heights are variable to the content or display area.

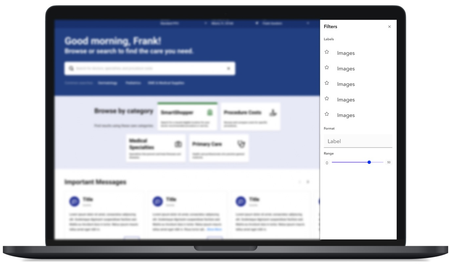


Content

Side sheets can display various content formats and layouts, ranging from a list of actions to supplemental content in a tabular layout.

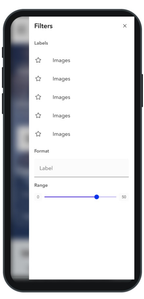
Non-Modal Side Sheet:

This non-modal side sheet features various items relevant to the context on the screen, blocking the need for a scrim.



Modal Side Sheet:

This modal side sheet is on mobile, requiring a scrim to direct user attention.



Building with the Side Sheet Component

Components disallow the insertion of items into content sections. Instead, detach the instance before inserting your chosen content. Headers are componentized within the side sheet, and header variants are swappable once the component is detached.

zelis

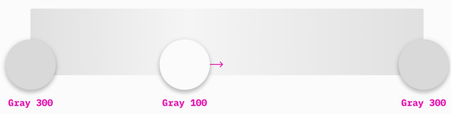
Skeleton Loaders

Design Kit Figma

Skeleton loaders are UI components that optimize user experience by reducing perceived latency during content loading. Skeleton loaders present a minimalist, wireframe-style structure that visually communicates the content's loading state to users. This visual indicator reduces perceived latency and user frustration.

Color/Animation

The skeleton objects have shadows / gradients moving from left to right. Content will replace these loading placeholders immediately when the data is retrieved.

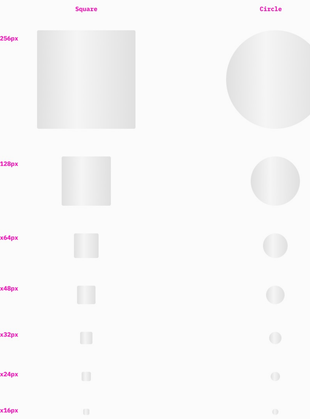


Type Loaders

Type loaders indicate the space where text populates. Morpho scale is the basis for sizing, so use a size consistent with your design. An 8px distance should always separate multiple lines.

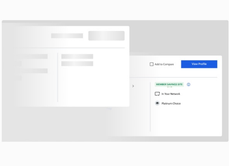
Shape Loaders

Shape loaders indicate the space where an element loads. Stretch or resize the element to match the content loading in.



Usage

Due to our white-labeled products supporting various authenticated and unauthenticated experiences, predicting what content displays to all users may prove challenging. Focusing more notional/high-level skeleton loaders on primary elements allows us to effectively communicate to the user expectations of where information populates in specified areas.

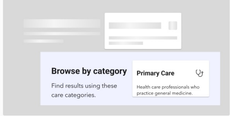


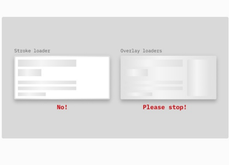
Mimicing Primary Information

Some elements are subject to configuration or variability - in these cases, it's often entirely appropriate to omit a skeleton loader for that bit of information. In other instances, omitting a loader for small elements, such as an ellipses icon, may be relevant as long as removing the loader doesn't affect primary on-screen information.

Color

Communicate account configuration colors (e.g., primary/secondary) through a solid gray or omitted from the skeleton screen.





Borders + Other Special Cases

Use skeleton loaders sparingly, and do not apply skeleton loaders instead of all elements (omit from lines and borders).

zelis

Slider

Design Kit Figma

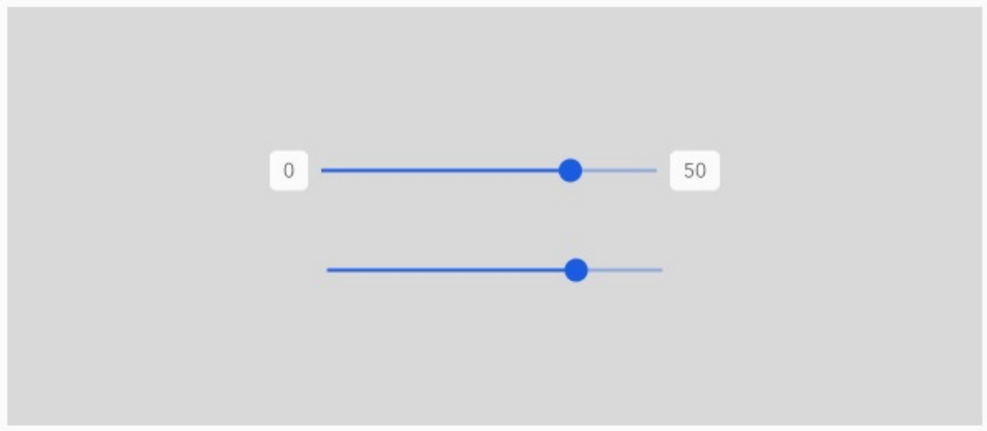
A slider, often called a slider control or slider input, allows users to select a value from a continuous or discrete range. Sliders are helpful in various applications and websites for adjusting settings, making selections, or controlling variables by dragging a handle along a track.

Usage

Sliders let users view and select content from a multiple-options bar. Sliders are ideal for adjusting settings with a wide range, such as volume and brightness or for applying image filters.
Sliders can use icons on both bar ends and for the values range to represent a numeric or relative scale.

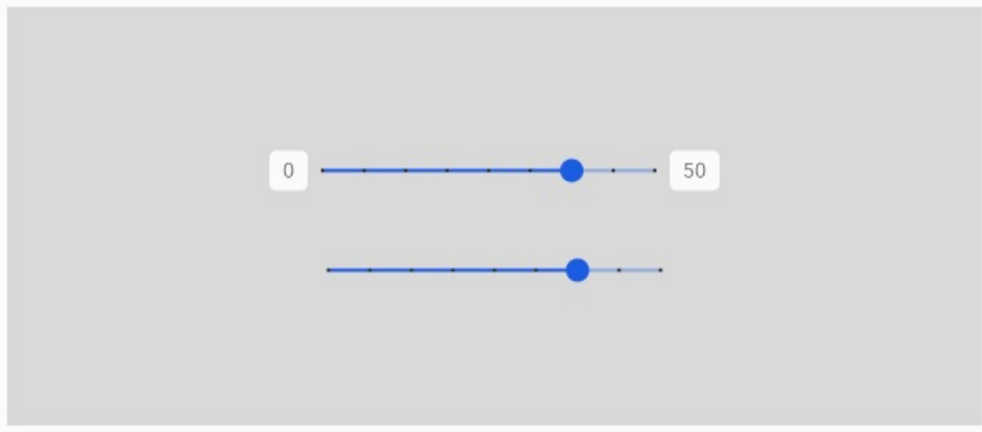
Continuous Sliders

Continuous sliders allow users to set and select a value along a subjective range.

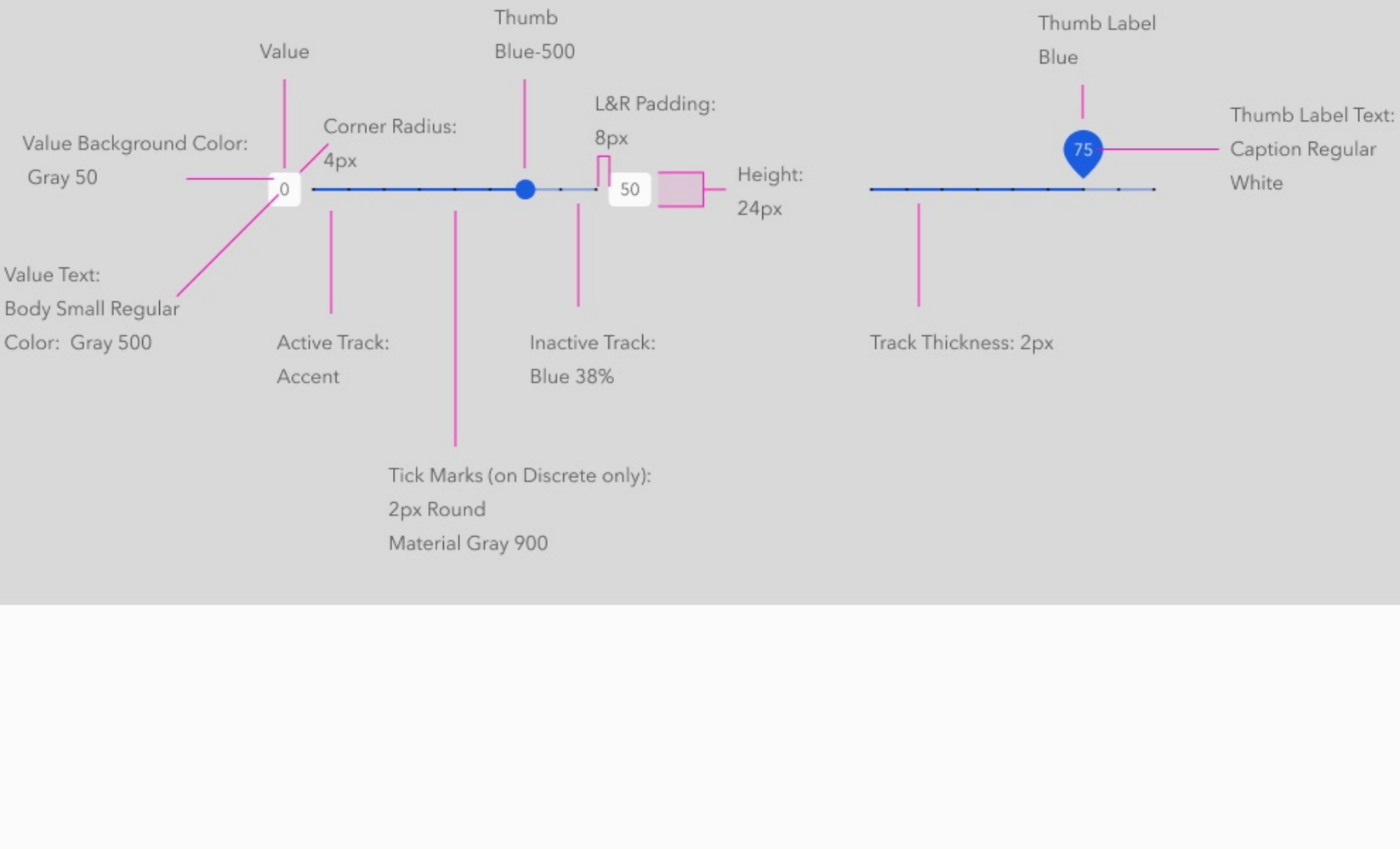


Discrete Sliders

Discrete sliders are adjustable by referencing their value indicator. Allowed selections may be organized and indicated with tick marks that a slider thumb will snap to.



Specs



zelis

Snackbar

Design Kit Figma

A snackbar (also known as “Toast”) is an unintrusive message that notifies the user with concise feedback. Snackbars will disappear automatically after a set amount of time but can also be dismissed by the user.

Anatomy

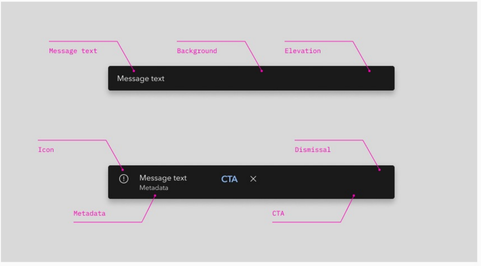
Snackbars notify users of active or soon-to-be active processes in an app. Snackbars temporarily appear on the bottom of a screen to avoid disrupting the user experience.

Constants

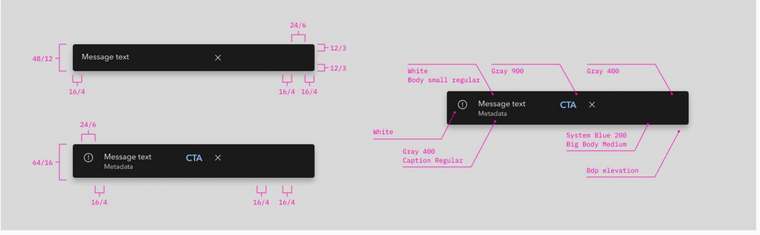
Elements that are always in the snackbar.

Additional Elements

Apply these elements to best serve the user.



Space + Sizing



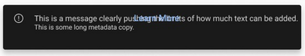
Maximum Content

Snackbars contain a text label directly relating to the active process. On mobile, the text label can contain up to two lines. Snackbar text should be at most roughly 40 characters. If the text of a message exceeds 40 characters, consider more concise wording, using the metadata property or choosing a dialog instead. Additionally, limit metadata copy to 20 or fewer characters.

Sizes

Large +

Medium



Small -

Max Width 960:
The snack bar should extend to fit the content within it so long as the display can accommodate its width using a maximum width of 896px (56rem).

Dismissal:
At the medium display breakpoint, the dismissal “x” should be removed to allow more room for text and encourage a swiping dismissal.

Exceeding 2 lines:
Messages should be at most two lines. Place a message beyond three lines (including translations) in a complementary ‘learn more’ button that triggers a dialog or choose a different component better suited to the content.

CTAs should shift to a vertical stack when the small breakpoint is hit.

Iconography

While snackbars allow for supporting iconography, using an icon may only sometimes be necessary.

Informational
Provide additional information to users that may not be tied to an action or task.

Success / Positive
Confirm a task was completed as expected.

Warning
Inform users that there may be changing or unexpected results.

Error / Critical
Inform users of an error or critical failure.

Other
Support a system message with complementary iconography for stronger comprehension.

None
Use when a message clearly stands on its own and doesnt require additional support.

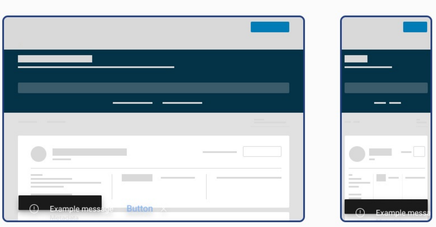
Duration + Placement

Snackbars appear without notice and don’t require user interaction. They automatically disappear from the screen after a minimum of four seconds and a maximum of 10 seconds unless there is a dismissal ‘X’; no time limit should apply.

When multiple snackbar updates are necessary, they should appear one at a time.

Snackbars live in the bottom left side of the screen and will maintain their width until their right-side lines up with the right padding of the window. Snackbars appear in front of page elements while sitting behind a scrim and dialog.

Special case:
Snackbars should not overlay navigation content (left-side navigation panels, bottom navigation on mobile, etc.)



zelis

Star Ratings

Design Kit Figma

Star ratings allow users to share their opinions on topics by assigning a value in stars to indicate satisfaction levels. Star ratings provide a simple and effective scale for conveying user feedback without text.

About

The star ratings atom is based off of the star ratings icon categorized as a “reference” icon. The default star ratting atom inherits fill and outline system colors.

Star Rating Icons - Variants

Star Ratings depict objects in the context of a ranged value system.

Empty Half Full

Anatomy

1.Stroke: Yellow 900
2.Fill: Yellow 700

zelis

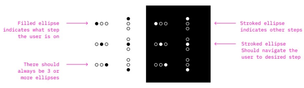
Stepper

Design Kit Figma

Steppers display user progress through a sequence of logical and numbered steps. Steppers are also helpful for navigation. Steppers may display a transient feedback message after saving a completed step.

CollinzDatz

Use this stepper as a subtle indicator of a user's progress. It is also helpful for carouselling content like an image gallery or slide deck.



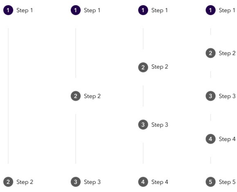
Within the larger stepper component, some steps take on various styles depending on what they're indicating.

Stepper

A stepper component can include unlimited steps. To stay concise and intuitive, do not exceed five steps. If you have more than five steps, consider consolidating steps for a better user experience.

Horizontal

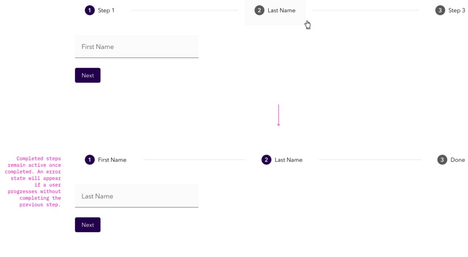
Vertical



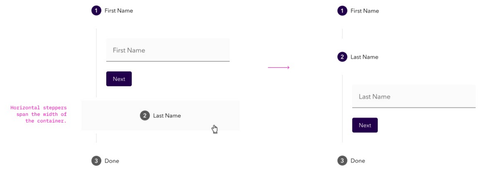
Behavior

Steppers communicate both static and dynamic information about the elements within each step. When concluding a step, ensure any completed information refreshes with the subsequent step's information.

Horizontal



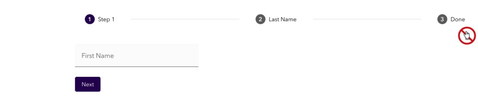
Vertical



Linear

We can force users to move through steps one at a time or allow users to hop freely between them. We call these two modes linear and non-linear.

Linear



Non-Linear

zelis

Switch

Design Kit Figma

Switches toggle the state of a single item on or off.

Slide Toggles Usage

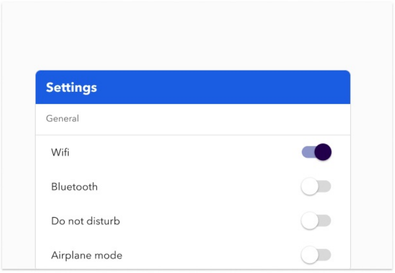
Slide toggles are the preferred way to adjust settings on mobile.
Use switches to:

- Toggle a single item on or off, on mobile and tablet.
- Immediately activate or deactivate an object.

Usage - Do's & Dont's

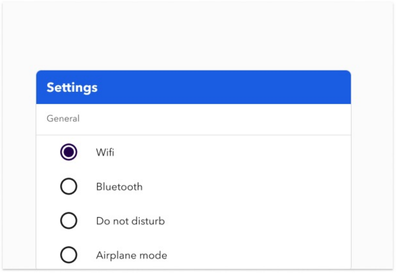
Slide toggles should be used instead of radio buttons if each item in a set can be independently controlled.

Do



Do:
Use switches to turn an item on or off instead of a checkbox especially on mobile.
Note this is not a real component it was created for illustrative purposes

Don't

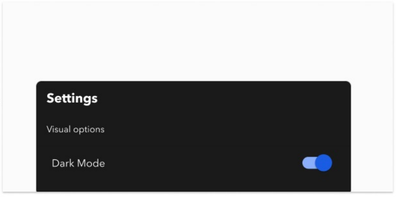


Don't:
Don't use radio buttons to toggle items on or off. Radio buttons convey that a set of items are options, and that only one can be selected at a time. Instead, use switches.
Note this is not a real component it was created for illustrative purposes

Behavior

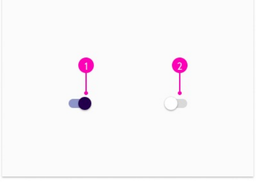
A slide toggle is successfully toggled when the thumb slides to the other side of the track upon user interaction.

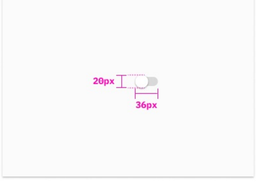
When a user toggles a switch, its corresponding action takes effect immediately.
Note this is not a real component it was created for illustrative purposes



When a user toggles a switch, its corresponding action takes effect immediately.
Note this is not a real component it was created for illustrative purposes

Anatomy & Specs





Configuration

Slide toggles are configurable according to material and angular.

Slide toggles may either use a Primary 500 or Blue 500 values when the Slide toggle is on and Grey 700 when it is off

State	Element	Color
Enabled On	Thumb	Primary 500
	Track	Primary 200
Disabled On	Thumb & Track	38% of primary 500
Hover On	Thumb	Primary 500
	Track	Primary 200
	Ripple	4% of primary 500
Focus On	Thumb	Primary 500
	Track	Primary 200
	Ripple	12% of primary 500
Pressed On	Thumb	Primary 500
	Track	Primary 200
	Ripple	8% of primary 500

State	Element	Color
Enabled On	Thumb	Blue 500
	Track	Blue 200
Disabled On	Thumb & Track	38% of Blue 500& Blue 200
Hover On	Thumb	Blue 500
	Track	Blue 200
	Ripple	4% of Blue 500
Focus On	Thumb	Blue 500
	Track	Blue 200
	Ripple	12% of Blue 500
Pressed On	Thumb	Blue 500
	Track	Blue 200
	Ripple	8% of Blue 500

State	Element	Color
Enabled On	Thumb	White
	Track	Grey 400
Disabled On	Thumb & Track	38% of Grey 400 and white
Hover On	Thumb	38% of Grey 400 and white
	Track	
	Ripple	4% of Grey 400
Focus On	Thumb	38% of Grey 400 and white
	Track	
	Ripple	12% of Grey 400
Pressed On	Thumb	38% of Grey 400 and white
	Track	
	Ripple	8% of Grey 400

zelis

Table

Design Kit Figma

A table is a fundamental element in web and application design used to organize and display structured data in a tabular format. Tables are essential for presenting information in a clear and structured manner, making it easier for users to compare, analyze, and understand data.

About

Data tables can contain:

- Interactive components (such as chips, buttons, or menus)
- Non-interactive elements (such as badges)
- Tools to query and manipulate data - future consideration

Organized
Information should be organized in a meaningful way, such as hierarchy or alphabetization.

Interactive (future consideration)
Data tables should allow user interaction so that a data display is customizable and interactive.

Intuitive
Data tables should be easy to use, with a logical structure that makes content easy to understand.

Variants

There are three styles/variants of cells: an outlined cell, a toplined cell and a header cell. A groupement of outlined cells form an outlined table, and a groupement of toplined cells form a Topline table. The header cell can be used in conjuction with both tables.

The outlined table is closely aligned with the angular table and is best used for a basic data bases. The toplined table is more suitable for our compare feature which is contains more complex information and interactions .

Outlined Cells and Table

Outlined cells and tables feature an outer stroke of 1 and grey 300. A stroke delineates each row and column.

Outlined Cell

Top Left Outlined	Middle Outlined	Top Right Outlined	
Middle Outlined	Middle Outlined	Middle Outlined	
Middle Outlined	Middle Outlined	Middle Outlined	
Bottom Left Outlined	Middle Outlined	Bottom Right Outlined	

Outlined table

Toplined Cells and Table

Toplined cells and tables feature top & bottom strokes of 1 and grey 300. Delineate the columns with a grey 50 fill, which carries through the entire column.

Toplined Cell

Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider

Toplined table

Styling

Based on the design needs, cells and tables can inherit various atoms, including icons, styling, and color. The styling of the cells and tables will be called out at the page and project level. By default, the copy of a cell is Body - AVE Regular and Body Medium for emphasis.

Anatomy

Outlined Table, with Top Emphasis

1. Emphasis can be placed along the top or down the left side by using a medium weight font to differentiate content.
2. Default text is Body-Regular (all content defaults to Mat Gray 900).
3. Cells utilize a Mat Gray 300, 1px stroke. Tables contain no elevation. Also, a "middle" cell is any cell that is not a corner.
4. Corner radius: 4px

Toplined Table , with Top Emphasis (this area can be sticky upon scroll)

Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider
	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider

1. Same as Outlined Table,
2. Same as Outlined Table
3. Cells utilize a Mat Gray 300, 1px top and bottom stroke. Tables contain no elevation.
4. No corner radius on a toplined cell
5. Alternate coolmns can be delinated (not an obligation) by a grey 50 fill which carries through the entire column. This will enhance legibility/scannability.

Toplined Table, with Left Emphasis

Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider

1. Depending on the information contained in the table and the visual hierarchy needed, emphasis can be placed along the top row, or down the left column with the Designer simply changing the Type Style 'on the fly'to "Body Medium".

Sticky Table Header , with alternating columns

Provider Name	Provider Name	Provider Name	Provider Name

1. Title text is H5 Bold ; Grey 900 ; Is a text link
2. Elevation 6dp - Sticky headers similar to FAB or snackbar since they are non modal and sit above the UI.

Behaviors

Hover State

Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider

1. Default behavior: When a user hovers on a row, that row changes color for visual feedback: Mat Gray 100. Build the hover state to turn on and off. Not all designs require a hover state.

Scaling

4 columns

Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider

963px

3 columns

Toplined w/top & bottom divider	Toplined w/top & bottom divider	Toplined w/top & bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider
Toplined w/bottom divider	Toplined w/bottom divider	Toplined w/bottom divider

963px

1.The columns scale/adjust in width according to the size of the table and also according to the number of columns in the table
2.Rows scale in height according to the width of the columns and the amount of data/input.

Interactive behaviors within the cells

Toplined w/right icon & top/bottom divider

Expand and close Behavior

Cells can expand and collapse displaying the same behavior as expansion panels

For Designers - How to use and adjust table

Top Left Outlined	Middle Outlined	Middle Outlined	Top Right Outlined
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt	Middle Outlined	Middle Outlined	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
Middle Outlined	Lorem ipsum dolor sit amet	Middle Outlined	Middle Outlined
Bottom Left Outlined	Middle Outlined	Middle Outlined	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt

1. Depending on the content length within specific cells/rows or the layout width/ height of the table, the Designer may need to detach the entire table component or individual cells and rows and adjust the height, width, or auto layout. Or, change all three to force the table to conform to the design. Such adjustments will likely require some trial and error (and cursing), but there is no way to predict the needs/requirements of individual designs.
2. Default auto layout properties changed.
3. Default auto layout properties changed.

zelis

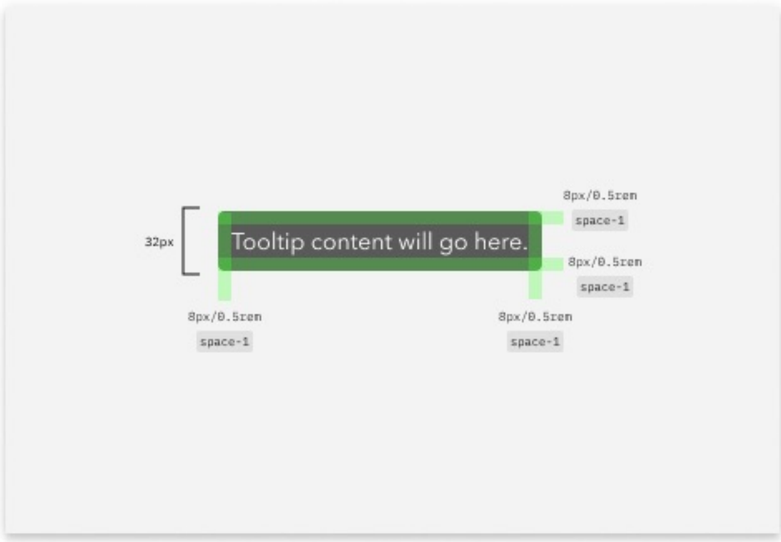
Tooltip

Design Kit Figma

A tooltip is a short, instructive message that appears when a user interacts with an element in a graphical user interface (GUI). Tooltips display in one of two ways: through a mouse or keyboard-hover gesture.

Usage

An activated tooltip displays a text label specifying an element, such as a description of its function.



Behavior

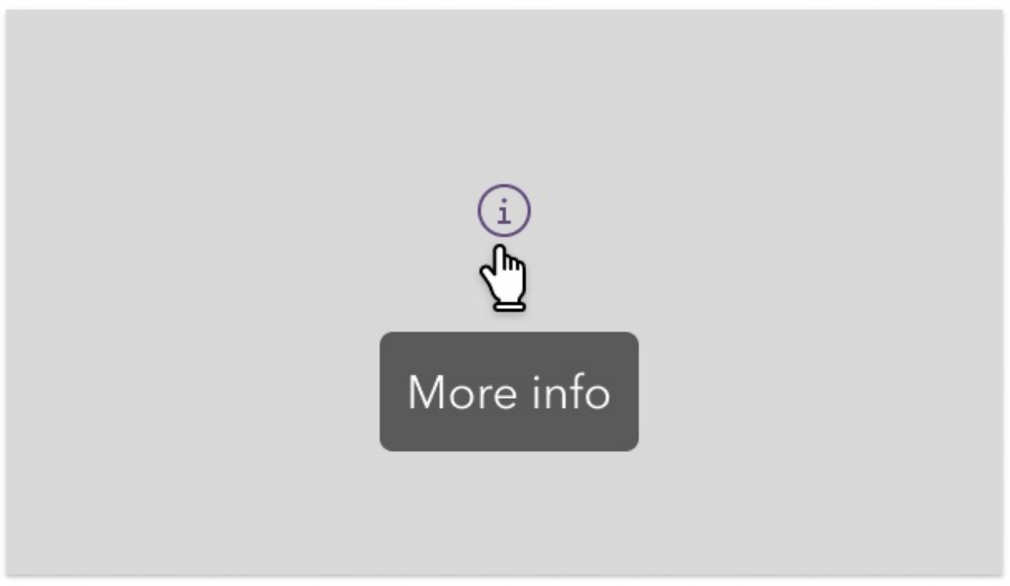
A tooltip displays when tapping and holding a screen element or component (on mobile) or hovering over it (on desktop). The tooltip remains displayed until the user ceases hovering over or long pressing the component.

Timing

Display the tooltip for 1.5 seconds. The tooltip will disappear if the user takes another action before that time ends.

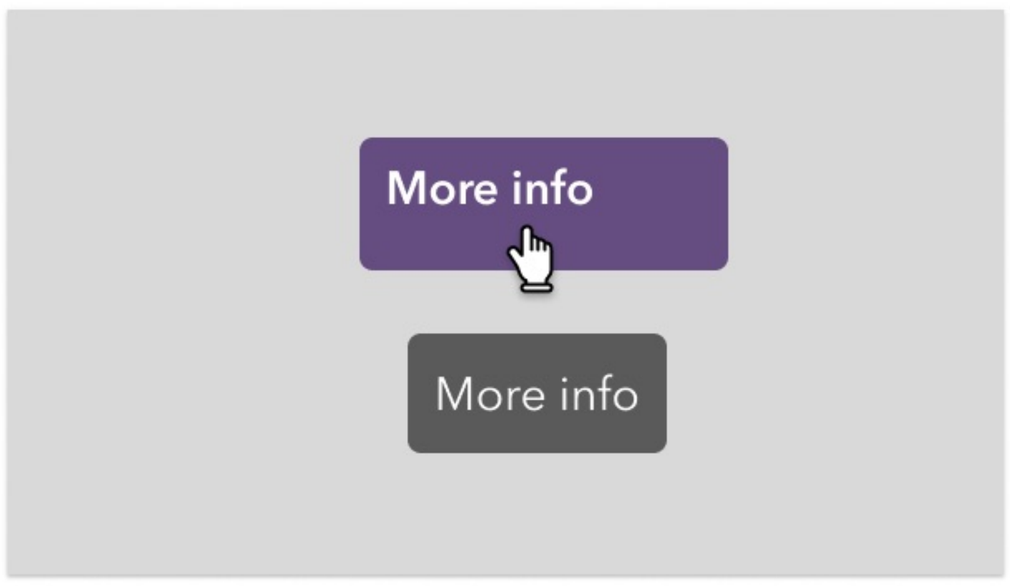
Guidance

Do



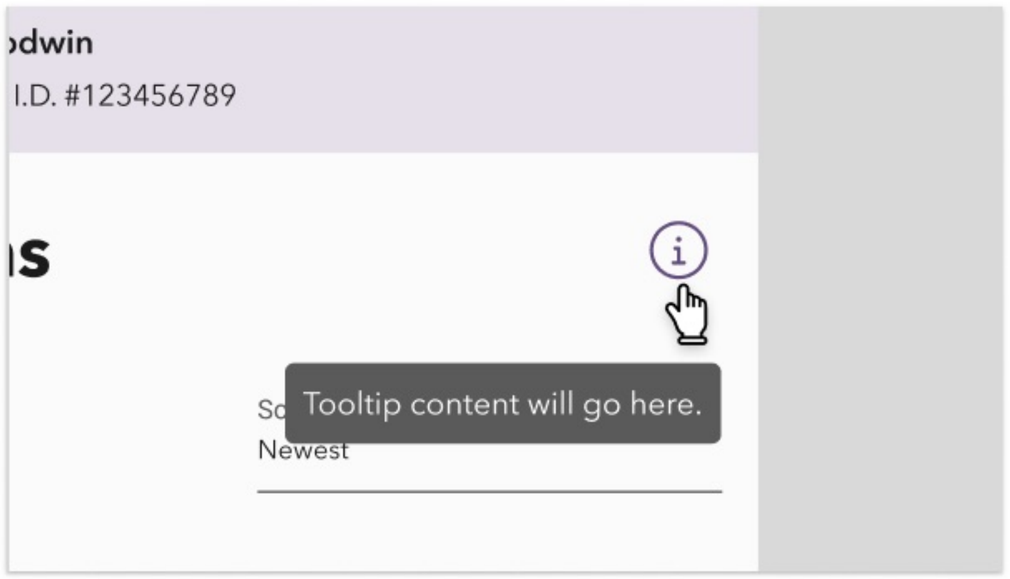
Do:
Use tooltips for interactive imagery.

Don’t



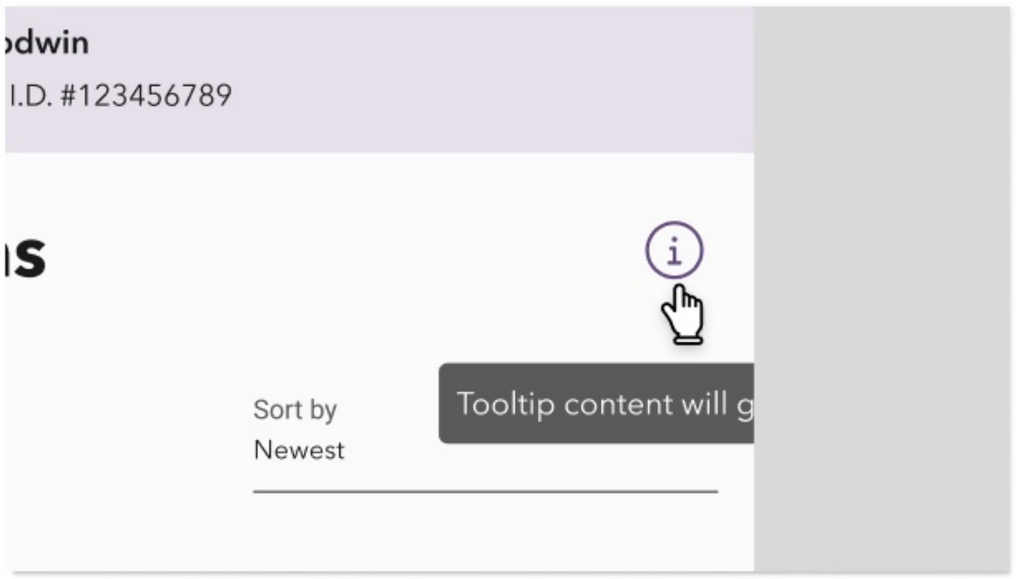
Don’t:
Don’t use tooltips to restate visible UI text.

Do



Do:
Use tooltips for interactive imagery.

Don’t



Don’t:
Don’t use tooltips to restate visible UI text.