

**Sub-atoms**

# Breakpoints

- [Overview](#)
- [Media Queries](#)

## Overview

---

Breakpoints are defined points or ranges based on a screen's viewport which dictates the styles applied at each of those levels thereby making a page **responsive**.

The platforms typography, sizing, spacing and others are built relative to the specified responsive breakpoints using css media queries.

## Media Queries

---

Due to the high-density nature of the platform, the responsive breakpoints gives higher consideration to the desktop sizes but still accommodates mobile based sizes.

Below are the set of available breakpoints (taken from legacy code):

Mobile based:

- Extra Small (XS)
- Small
- Medium

Desktop based:

- Large
- Extra Large (XL)
- 2XL

Token	Breakpoint	Description
<code>breakpoint-sm</code>	min-width of 768px	Viewports 768px and wider (Tablet Portrait)
<code>breakpoint-md</code>	min-width of 992px	Viewports 992px and wider (Tablet Landscape)
<code>breakpoint-lg</code>	min-width of 1200px	Viewports 1200 and wider (Laptop)
<code>breakpoint-xl</code>	min-width of 1600px	Viewports 1600 and wider (Desktop)
<code>breakpoint-2xl</code>	min-width of 1800px	Viewports 1800 and wider (Desktop)

## Changelogs:

- 08/29/23 - Released initial docs (Raul)
- 09/06/23 - Removed XS breakpoint since its not necessary (Raul as per Andrew)

# Elevation

- [Overview](#)
- [Elevation Levels](#)
- [Usage guidelines](#)

## Overview

Elevation in design measures the perceived depth or distance between different elements on a digital interface. It gauges how prominently an element stands out or recedes visually, often through techniques like shadows, highlights, or borders. This visual hierarchy aids users in comprehending the arrangement and significance of elements, contributing to an intuitive and organized design layout.

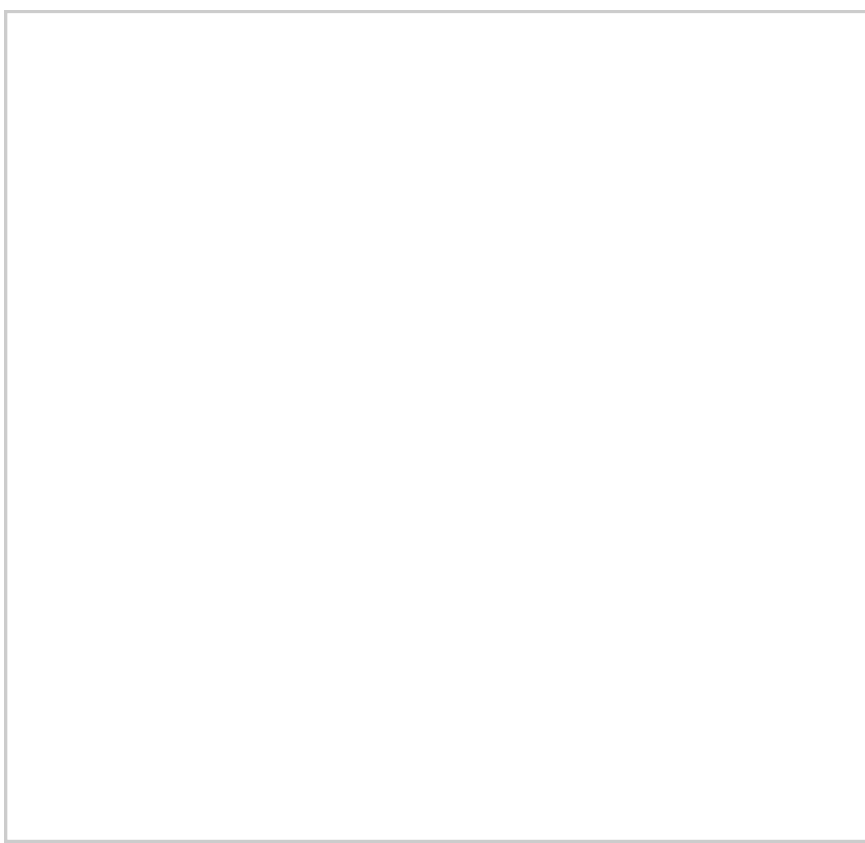
## Elevation Overlay



### tui-elevation-1

```
box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.1);  
border-style: none;
```

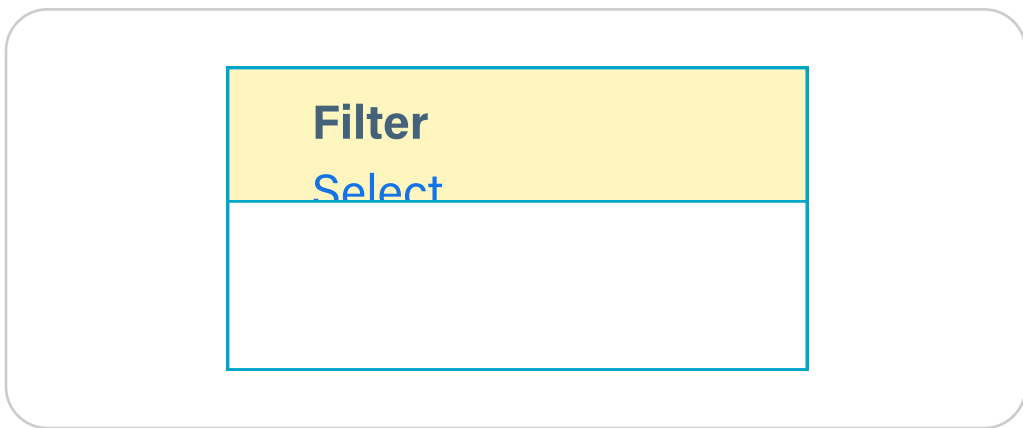
## No Elevation



### tui-elevation-0

```
box-shadow: 0px 0px 0px rgba(0, 0, 0, 0);
```

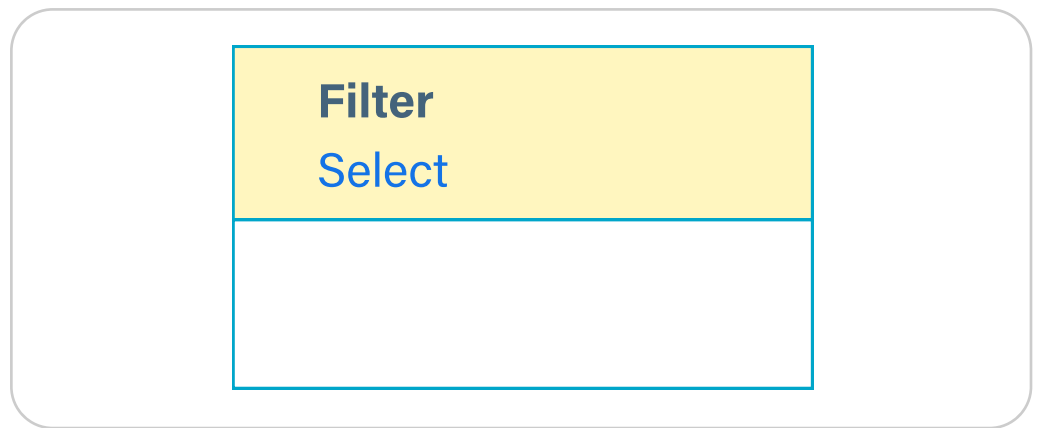
## Usage guidelines



### ✔ Do

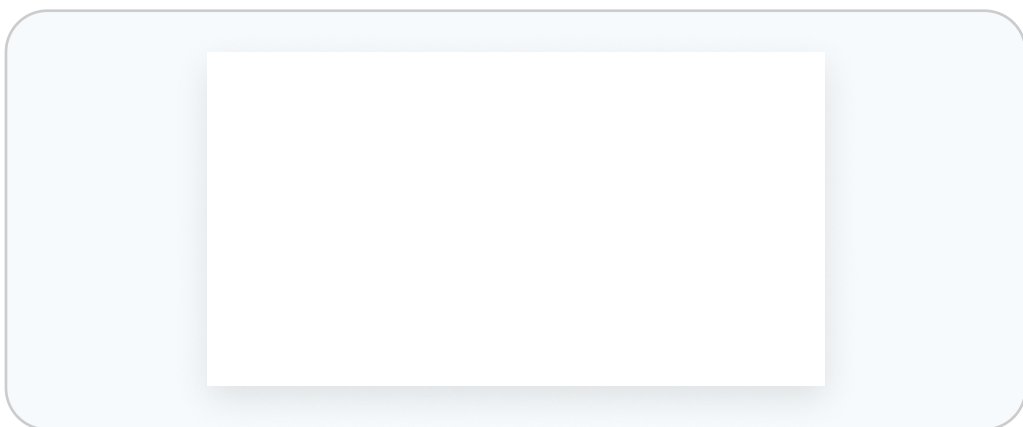
For filter dropdowns such as in *“View Audiences”*,

The white box layers over and partially obscures “Select” to really emphasize that the white box is higher in elevation than the yellow box. It’s telling us that we should draw our focus more to the content in the white box. Using visual separation forges a strong connection between the elements for more efficient navigation creating a more intuitive user experience.



### ✘ Don't

**Not** layering the white card and obscuring “Select” makes it seem like all the boxes are on the same level of elevation. We **don't** want that.

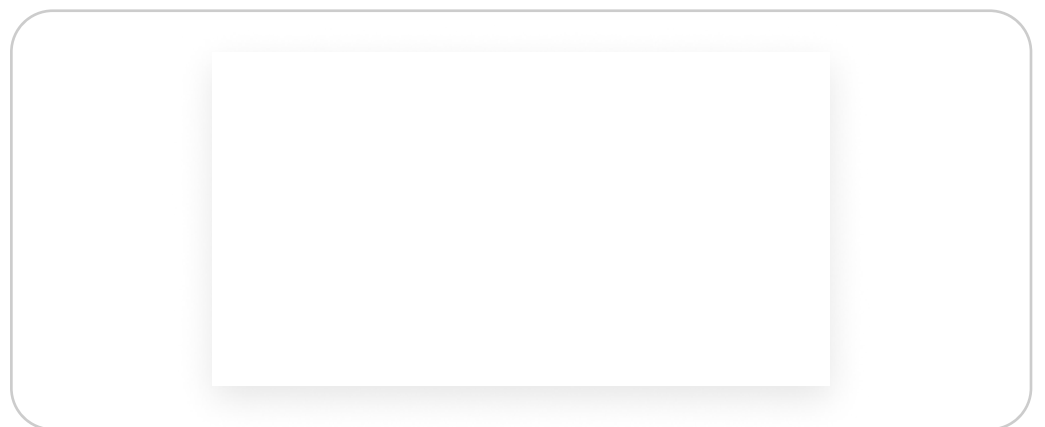


### ✔ Do

White cards usually use

```
box-shadow: 0px 5px 15px rgba(0, 0, 0, 0.1);  
border-style: none;
```

against a darker background



### ✘ Don't

Don't lay white cards on top of a white background

# Motion

- [Overview](#)
- [The Purpose of Motion](#)
- [Motion Values](#)
- [Motion Names](#)
- [Elements that use Motion](#)
- [Usage Guidelines](#)
- [FAQ's](#)

## Overview

Motion in UI interactions serves several important purposes that contribute to enhancing user experience and usability. There are several important considerations for motion in UI design, which will be briefly explained. At this early iteration of the Design System, we will create a simple structure of Motion settings, learn from user testing how well they work, and enhance our interactions based on our learnings.

## The Purpose of Motion

### Feedback and Visual Clues

Motion provides immediate visual feedback to users, confirming that their action has been recognized. This feedback helps users understand that they've interacted with an element successfully.

### Increased Engagement

Well-designed motion can make interactions more engaging and enjoyable for users. Delightful animations can create a positive emotional response and encourage users to explore further.

### Visual Hierarchy

Motion can guide users' attention and convey hierarchy. For example, a subtle animation can draw attention to a call-to-action button or highlight an important piece of information.

### Context and Continuity

Transitions and animations can establish a sense of context and continuity, helping users understand where they are in an app or website's navigation flow. Smooth transitions prevent disorientation.

### Affordance

Motion can enhance the perceived "clickability" or "tappability" of interactive elements. Subtle animations on buttons, links, and cards can make them feel more like physical objects.

### Micro-Interactions

Micro-interactions, such as button hover animations or loading spinners, provide real-time feedback about system status or user interactions, reducing uncertainty and anxiety.

### Storytelling and Conveying a Process

Motion can be used to tell a visual story or guide users through processes. Animations can explain functionality and help users understand how to use a product.

### Visual Interest

Static interfaces can be visually monotonous. Motion adds dynamism and variety, making the UI visually appealing and preventing user fatigue.

### Increased engagement

Well-designed motion can make interactions more engaging and enjoyable for users. Delightful animations can create a positive emotional response and encourage users to explore further.

### Concealing Latency

During interactions that involve data processing or loading, motion can be used to conceal perceived latency, making users feel that the system is responding more quickly.

### Navigation and Context Shifts

When moving between different sections or screens, motion helps users understand the spatial relationships and context changes. This prevents disorientation.

### Usability and Accessibility

Thoughtful motion can improve usability and accessibility by guiding users, simplifying complex processes, and providing visual cues for those with cognitive or sensory impairments.

### Brand Identity

Custom animations can become part of a brand's visual identity, making the UI memorable and recognizable.

## Motion values

### Legacy values

The legacy values currently used by Transunion are listed below, the units of time being seconds, and way in which it transitions is the default "ease function": the transition starts slowly, accelerates in the middle, and slows down at the end.

```
.smooth-ultra { transition: all 0.05s ease;}
.smooth-fast { transition: all 0.15s ease;}
.smooth { transition: all 0.5s ease;}
.smooth-long { transition: all 1s ease;}
```

### Transition values

For an animated illustration of several ease types, see easings.net.

How each UI element behaves should reflect the metaphor for the object. For example, a dropdown should mimic an object under the acceleration of gravity. This alone should move the dropdown from slow to faster as it descends, then stopping abruptly (easeInExpo). A "feeling" for greater quality adds slowing at the very end, as if cushioned (easeInOutCubic). So, for each interactive element, we should be mindful of what transition behavior corresponds to that object, and test with likely users to validate our choices.

We expect that several UI elements and components will share the same easing settings, so these common easings will get a names, as coined by Andrew, like productive, expressive, etc. These would point to settings like cubic-bezier(0.2, 0, 0.38, 0.9), or whatever is needed. The advantage is that these details can be managed when we get testing feedback with the least fuss and bother.

### Duration

How long each transition takes its the duration. Generally, this is measured in thousandths of seconds, or ms. Although there are no absolute values for how long interactions should last, we should consider first the time it takes people to respond to stimulus. The time it takes for a human to respond to a stimulus varies depending on the type of stimulus, the individual's sensory processing, and the specific response being measured. Two relevant aspects for digital UI are:

#### Visual Stimulus Response (Simple Reaction Time):

In simple reaction time tasks, where a person responds to a visual stimulus (such as a light or a color change), the average response time is around 200 to 250 milliseconds (ms). This is the time it takes for the brain to process the visual input and initiate a motor response.

#### Complex Tasks and Decision-Making:

For more complex tasks that involve decision-making, cognitive processing, and motor responses, response times can vary significantly. These tasks can take anywhere from a few hundred milliseconds to several seconds, depending on factors like the complexity of the task, the individual's familiarity with the task, and cognitive load.

It's important to note that response times can be influenced by factors such as the individual's age, alertness, attention, and overall cognitive processing speed. Additionally, the context of the stimulus and the task being performed play a significant role in determining response times.

For these reasons, durations of around 100ms will seem almost immediate, but will not just change in an instant. Durations of 200ms will seem quick, as they will generally be over before you can act on them. Longer timings will give more time for the person to consider what is happening. Until we test these durations with expected users, we won't know what their expectations or reactions are to different durations, so we will create a range of durations to choose from, and after testing can adjust to. This will give consuming teams other duration options for circumstances we can't anticipate at the moment.

💡 A scale of durations (multiplying the base value by the square root of 2):

- infinite (unlimited amount of time, seldomly used, ex. spinners)
- slow\_ultra 560ms
- slow\_very 400ms
- slow 280ms
- moderate 200ms
- fast 140ms
- fast\_very 100ms
- fast\_ultra 70ms

## Motion names

Using the same Naming Convention as other Subatoms, names should follow these guidelines:

- Use lowercase for all names.
- Hyphenate: Separate different parts of the name with hyphens. Do not put spaces on either side of the hyphen.
- Descriptive: use real words for the name of things, not abbreviations, or numbers.

The form of names should be:

motion-easing-(type)-duration-intent

For example, we might have motion-easing-productive-fast\_very

## Elements that use Motion

Interaction motion can be applied to various UI elements to enhance user experience, provide feedback, and guide users through interfaces. Here are some common UI elements that are expected to have interaction motion:

#### Buttons:

Button animations can include changes in color, scale, or shadow on hover or press to communicate interactivity.

#### Links:

Link hovers can trigger underline animations, color changes, or subtle movement to indicate clickable elements.

#### Dropdown Menus:

Dropdown menus can have animations like sliding or fading in and out to make the transition smoother and provide visual context.

#### Icons:

Icons can respond to interactions with animations, such as changing color or enlarging slightly on hover (unlikely).

#### Cards and Tiles:

Card or tile elements can animate when hovered or clicked to provide a sense of depth or interaction.

#### Tabs and Navigation:

Tabbed navigation can include animations like sliding indicators or content transitions to show the selected tab.

#### Form Elements:

Input fields, checkboxes, and radio buttons can have animations that highlight the focused element or provide feedback upon interaction.

#### Tooltips:

Tooltips can fade in and out or slide into view alongside their respective elements.

#### Modals and Dialogs:

Modals can have entrance and exit animations like fading in or sliding up, enhancing their appearance and interaction.

#### Sliders and Progress Bars:

Sliders and progress bars can include animations that smoothly update as users interact with them.

#### Pagination:

Pagination components can use animations to transition between pages or display the active page.

#### Hover Cards:

Hovering over certain elements can trigger a card with additional information, often animated to appear smoothly.

#### Loading Indicators:

Loading animations provide visual feedback while data is being fetched or processed, preventing user frustration.

#### Toggle Switches:

Toggle switches can include animations that smoothly transition between on/off states.

#### Scrolling Animations:

As users scroll, certain UI elements can appear, fade, or animate into view, adding visual interest.

#### Charts and Graphs:

Animated transitions in charts and graphs can help users understand data changes more intuitively.

***We should review each element and component as it is moved out of the backlog to determine if Motion is an aspect of that item.***

## Usage guidelines

### To follow...

## FAQ's

### Have questions?

Please ask, and your answers will appear here.

## Changelogs:

- (Fred)



# Radius

- [Overview](#)
- [Radius levels](#)
- [Usage guidelines](#)

## Overview



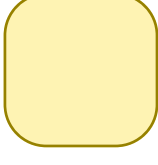
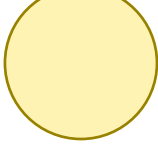
Stylistic brand expression describing the level of roundness a corner has for every shape, UI elements included.

The TU brand uses a lot of of rounded “bubble” shapes in its digital communication and marketing so it's only natural to extend that visual continuity inside the library as well.

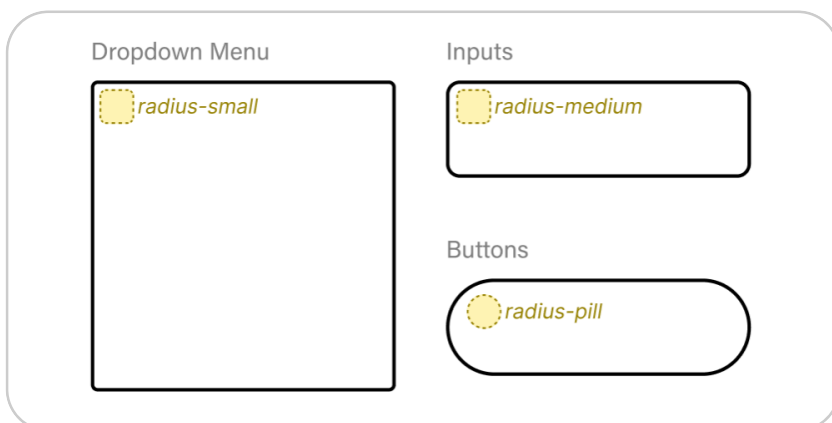
## Radius levels

Most interactive components are rounded with few exceptions, with which the level varies accordingly.

For accessibility purposes, we defined the scale in **EM**.

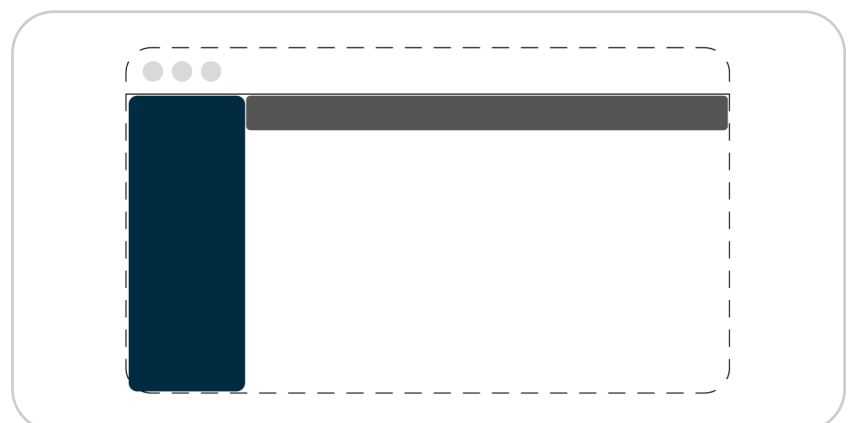
Token	Levels (EM)	Recommended Usage	Example
<code>radius-small</code>	0.333 (4px)	Dropdown Menus	
<code>radius-medium</code>	0.667 (8px)	Inputs	
<code>radius-large</code>	1.333 (16px)	Cards, Modals	
<code>radius-pill</code>	50 (600px)	Used in majority of the UI elements to create that distinct pill shape.  Buttons, Dropdown Button, Badges	

## Usage guidelines



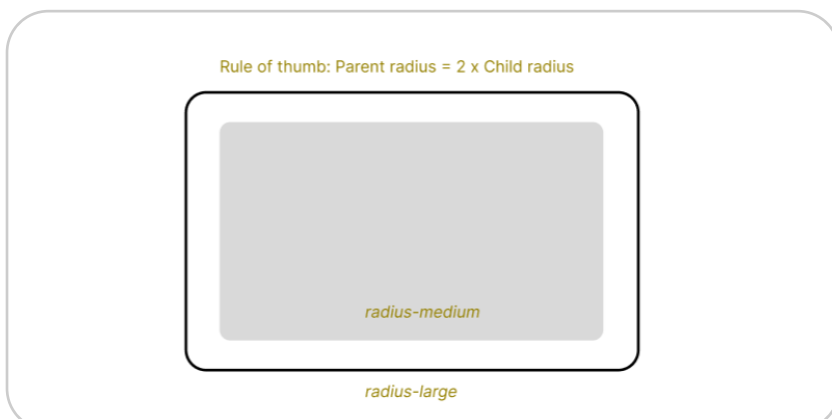
### ✓ Do

Use appropriate rounded corner if and when necessary; radius should be proportional to element size



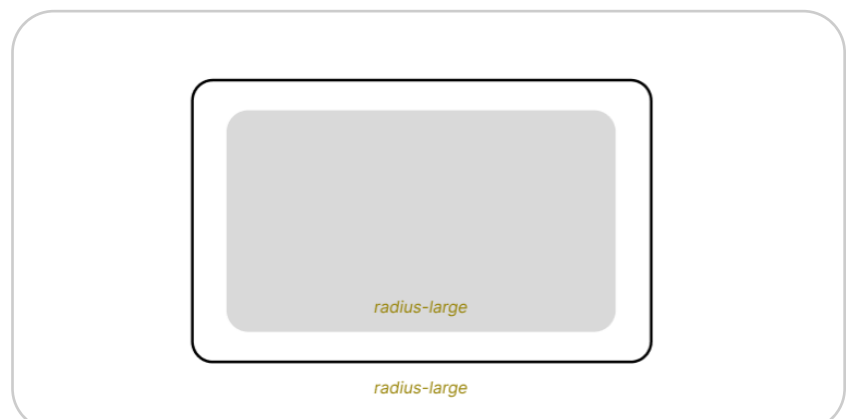
### ⊘ Don't

No corners for anything that spans full width (Header, Toolbar, Left Menu)



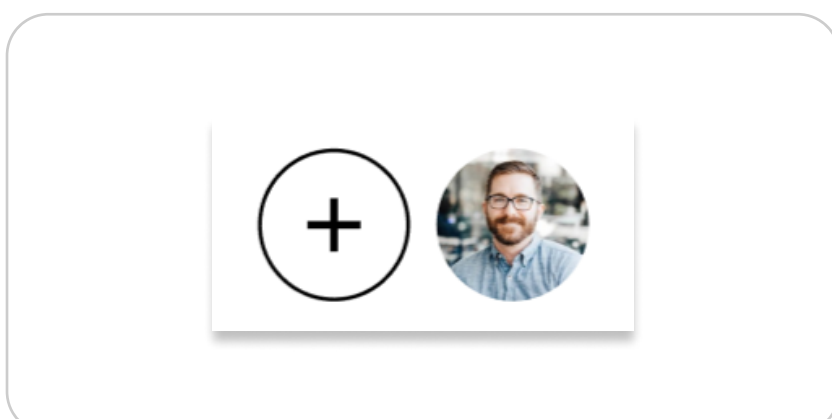
### ✓ Do

When using nested corners, the parent element must have a bigger radius than its child element



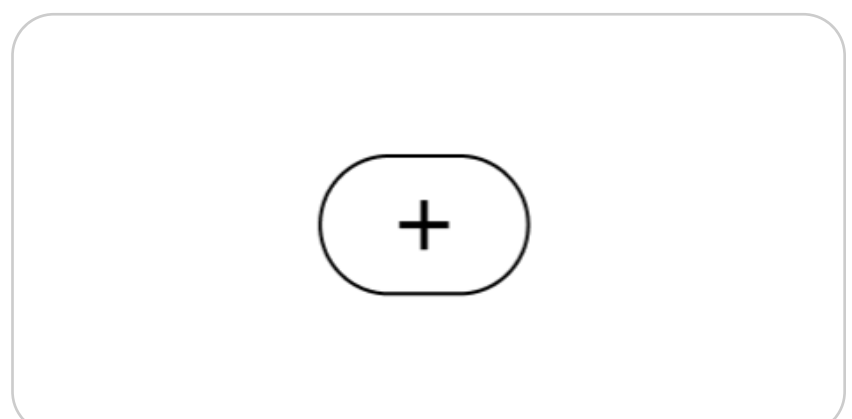
### ⊘ Don't

Avoid giving the child element same or much bigger rounded corner than its parent



### ✓ Do

For icon-only buttons and avatars, make sure that it's styled as a perfect circle than as a pill



### ⊘ Don't

No pill shaped buttons for icon-only except when making responsive buttons with icon & label

# Sizing

- [Overview](#)
- [Size Tokens](#)
- [Applying Sizes](#)
- [Usage guidelines](#)

## Overview

The design system provides various ways to create a consistent way of scaling the dimensions of a UI element in regards to its width and height properties using both discrete and relative methods.

## Size Tokens

### Component Sizes

#### Height

Measures the vertical dimension of a UI element given that the corresponding width is relative to content/context

Token	EM	Recommended Usage
<code>size-height-xsmall</code>	1.333 (16px)	Radio, Checkbox, Switch/Toggle
<code>size-height-small</code>	2 (24px)	Pills/Badges, Search filters
<code>size-height-medium</code> (Default)	2.667 (32px)	Basic Inputs, Dropdowns, Buttons, Action group
<code>size-height-large</code>	3.33 (40px)	Inputs with floating labels, Mega options

#### Width

Measures the horizontal dimension relative to the size of the viewport or screen resolution given that the corresponding height is relative to content/context.

Token	Values	Recommended Usage
<code>size-width-narrow</code> (Default)	25% or 25vw or 3 cols (in a 12 col grid)	Popovers, Menu, Dialog
<code>size-width-regular</code>	50% or 50vw or 6 cols (in a 12 col grid)	Modals with basic/simple forms
<code>size-width-wide</code>	75% or 75vw or 9 cols (in a 12 col grid)	Modals with longer forms, section
<code>size-width-full</code>	100% or 100vw or 12 cols (in a 12 col grid)	Largest modal size for rich contents

## Applying sizes

### Sizing by Density

The actual size of a UI element is calculated not only on the specified discrete height but includes spacing and typography choices.

Density of a UI element varies depending on the use case. Increasing or decreasing the padding for example as well as its font style (size, line-height, etc.), makes an element to appear bigger or smaller, respectively.

In most cases, providing a minimum height is recommended for proper scaling and better accessibility especially when enabling large font mode.

Note: Code snippets of mixins below are just pseudo-codes

#### • Small

Used in high-density displays such as data tables as well as elements that carry numerical/statistical data. Recommended minimum height of 24px

```
1 /*Small*/
2 @include text-small()
3 @include padding-small()
4 @include min-height-small()
```

#### • Medium

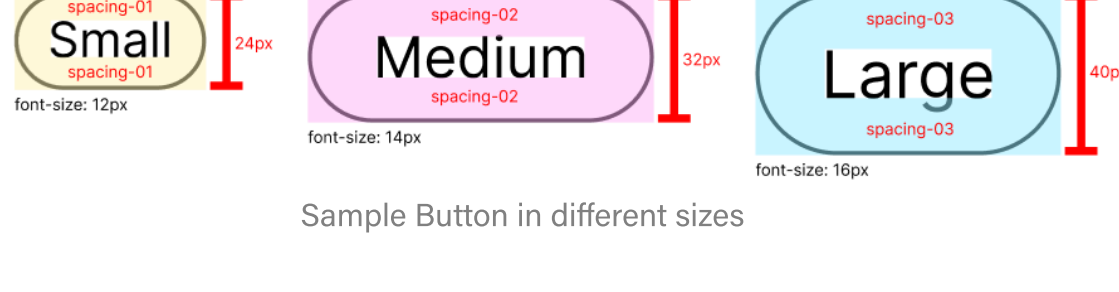
A default for majority of UI elements; best used when joining elements together  
Recommended minimum height of 32px

```
1 /*Medium*/
2 @include text-medium()
3 @include padding-medium()
4 @include min-height-medium()
```

#### • Large

Showcasing rich contents such as cards, modals, inputs with floating labels.  
Recommended minimum height of 40px

```
1 /*Large*/
2 @include text-large()
3 @include padding-large()
4 @include min-height-large()
```

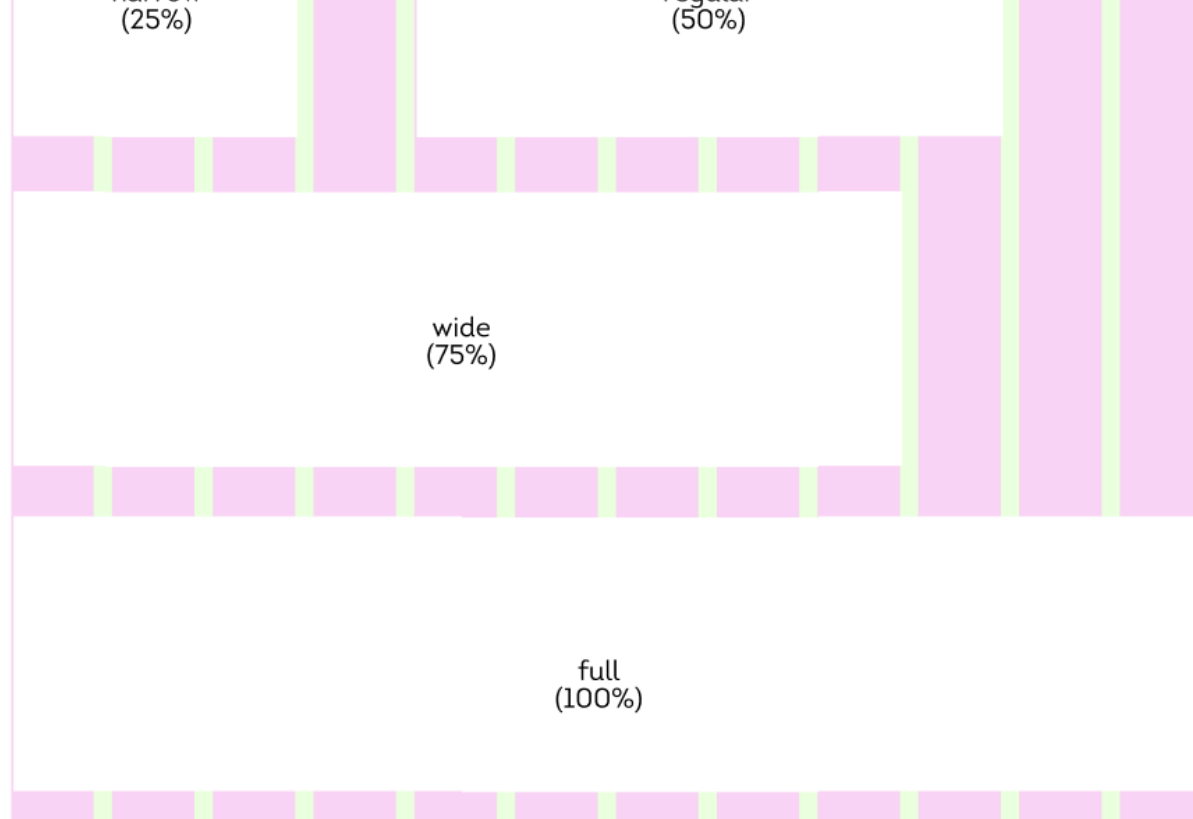


Sample Button in different sizes

### Sizing w/ Widths-only

Selecting the appropriate width size will depend on the amount of content that will be added. Each of the width sizes dynamically changes relative to the screen or browser size. These values are particularly useful for UI elements which persists as a single instance in the app (modals, popovers, dropdown menus).

For collections or sets, apply Sizing by Layout (see below).



Percentage based width sizes

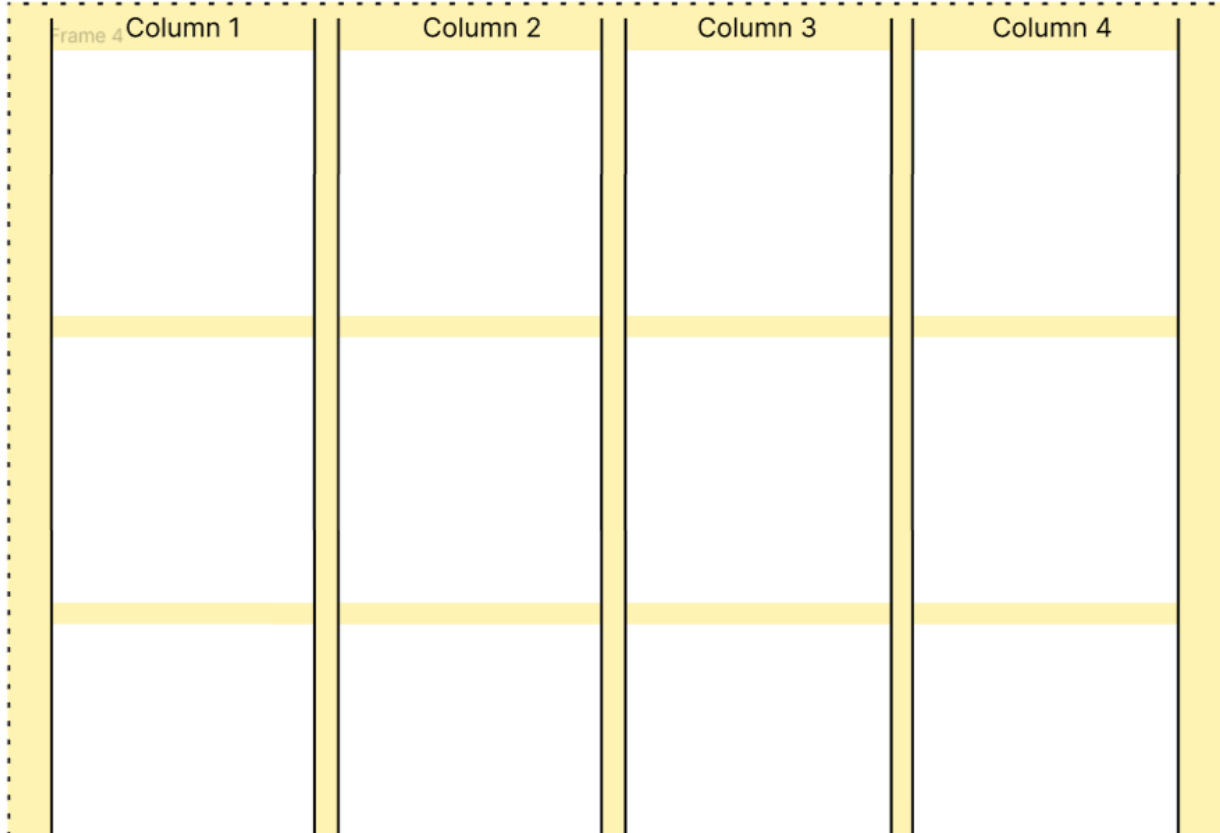
### Sizing by Layout

When organizing a collection of UI elements, they can be **arranged** as **columns** wrapped in a parent container (using **grid** or **flexbox**). The expected number of columns determines the width of each element:

```
1 100%(width of parent container) / number of columns = element width %
```

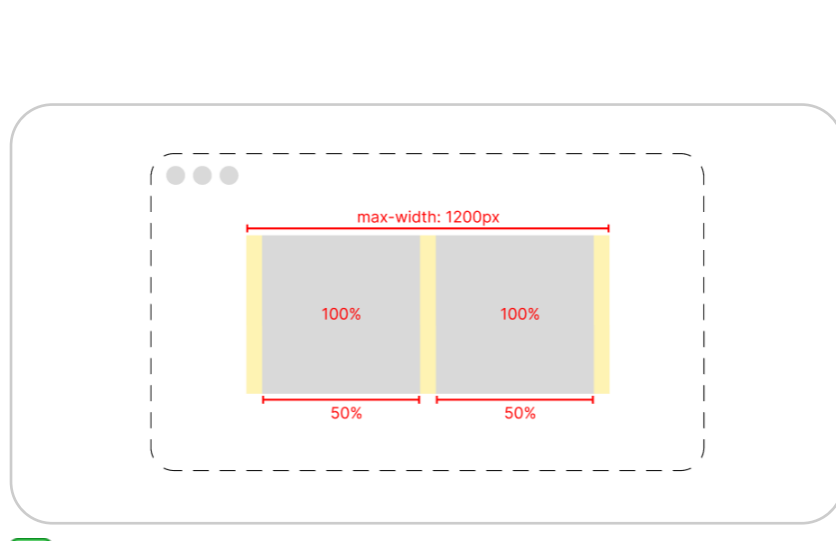
Given you want a collection of cards to be arranged in a 4 column layout. The width of each card will then be more or less 25% each.

```
1 100% / 4 = 25%
```



Relative Column Sizing

## Usage guidelines



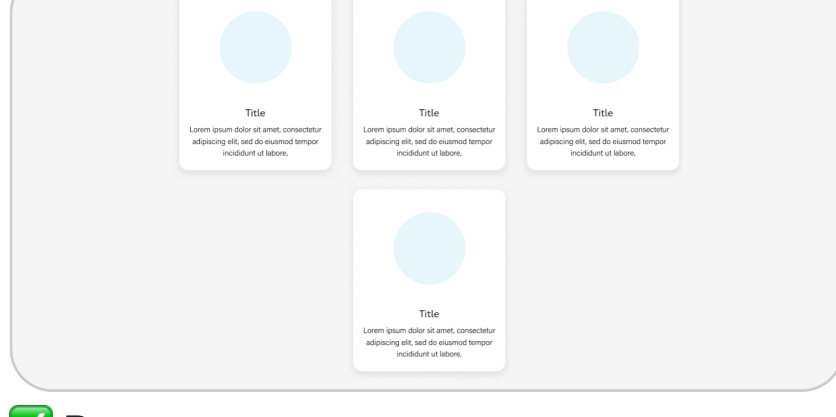
#### ✓ Do

By default, UI elements may fit according to content or occupy entire space of its parent container (100%)



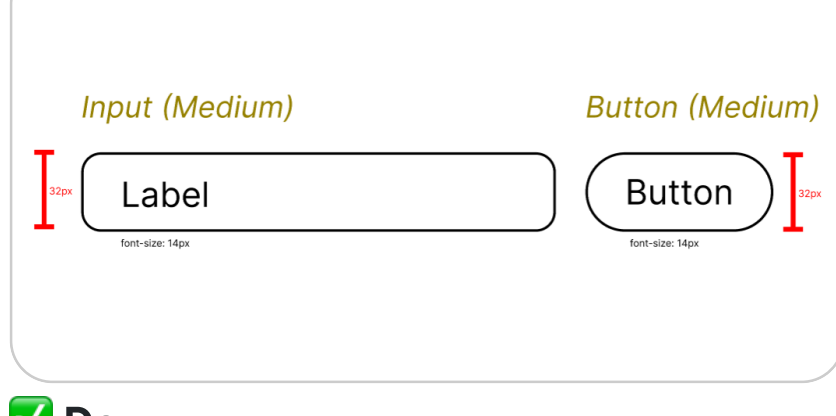
#### ✗ Don't

Avoid using discrete dimensions in components as much as possible



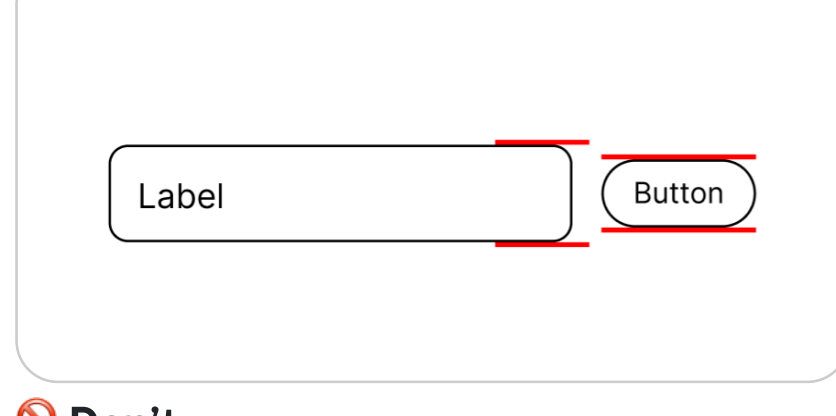
#### ✓ Do

Apply sizing by layout when displaying a collection of ui elements e.g Primary cards in Welcome page



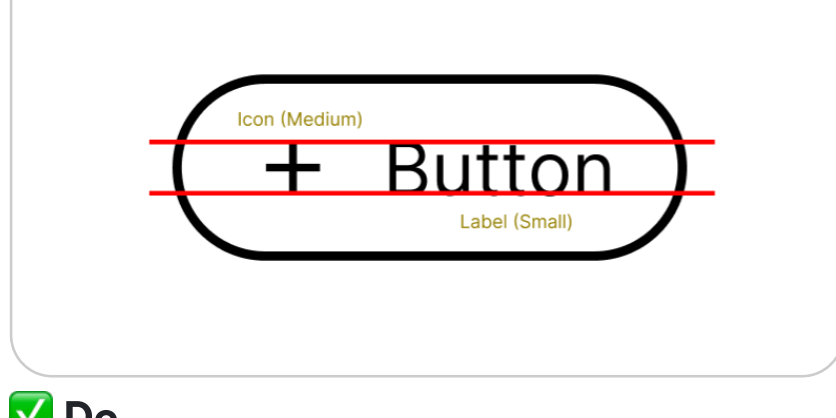
#### ✓ Do

When grouping together form based UI elements, make sure to use the same sizing convention for better alignment. (Action group, Forms, ...)



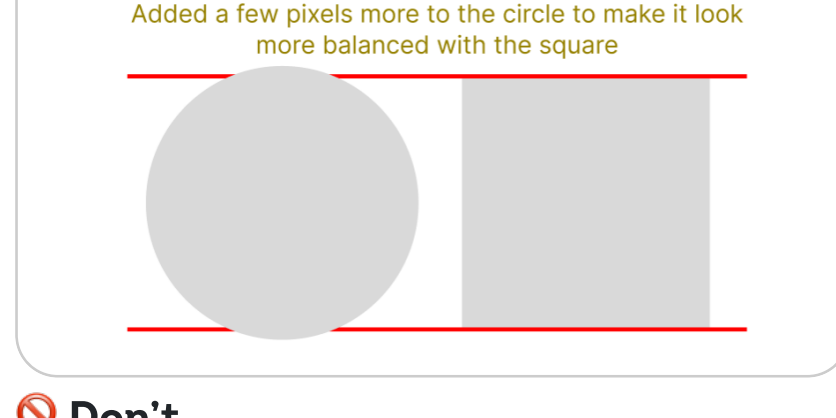
#### ✗ Don't

Avoid inconsistent UI element heights



#### ✓ Do

Take into account how our eyes perceived sizes, not just by pixels e.g icon size vs text size



#### ✗ Don't

Do not discard visual/optical findings in favor of sticking with sizing standards

## Changelogs:

- 08/08/23 - Released initial docs (Raul)
- 09/07/23 - Removed Icon sizes section here; moved it to Icons doc (Raul)



# Spacing

- Overview
- Spacing values
- Usage guidelines
- FAQ's

## Overview

Below are the defined tokens and its respective (em to px) values to control the white space between UI elements. These values and units are pre-determined to establish visual consistency and rhythm across all layouts and pages in the platform.

## Spacing values

The spacing system follows the 4 point baseline = 0.333em, where every increment in the scale is **base value \* multiplier = scale**

### Responsive scale

In order to ensure consistent spacing across all devices, the system provides corresponding responsive sizes for every unit. The base size defines the standard values for use in desktop (tablet included) screens then downsizes for mobile respectively.

For accessibility purposes, we defined the scale in **EM**. For viewport horizontal sizes under 768px (Mobile), the EM is 12px. For 768px up to 1200px (Tablet), it is 13px. Over 1200px (Desktop) it is 14px.

Based on the default mobile **12px font size**, we can categorize and derive the increments based on the following:

- extra small to small (+3 increments)
- small to large (+6 increments)
- large to extra large (+16 increment)

Legend: ■ Mobile (base) ■ Tablet ■ Desktop

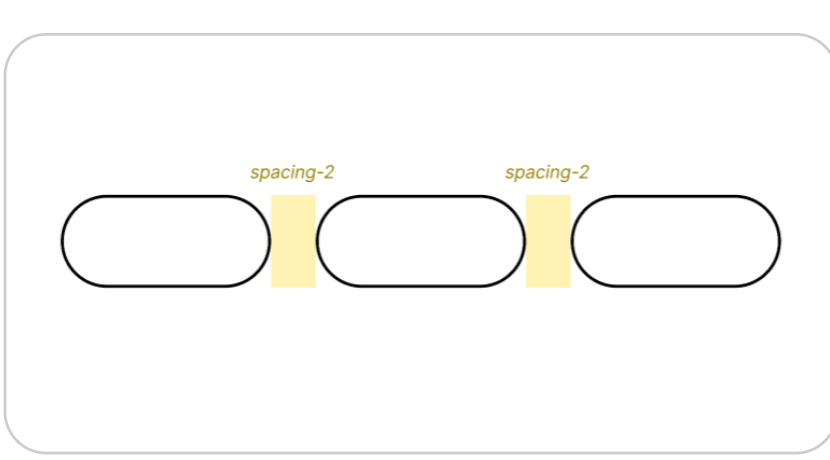
Token	Category	Mobile (EM)	Tablet (EM)	Desktop (EM)	Example
spacing-0	n.a	0	0	0	
spacing-1	extra-small	0.25 (3px)	0.25 (3.25px)	0.25 (3.5px)	
spacing-2	extra-small	0.5 (6px)	0.5 (6.5px)	0.5 (7px)	
spacing-3	extra-small	0.75 (9px)	0.75 (9.75px)	0.75 (10.5px)	
spacing-4	small	1 (12px)	1 (13px)	1 (14px)	
spacing-5	medium	1.5 (18px)	1.5 (19.5px)	1.5 (21px)	
spacing-6	large	2 (24px)	2 (26px)	2 (28px)	
spacing-7	extra-large	2.5 (30px)	2.5 (32.5px)	2.5 (35px)	
spacing-8	extra-large	3 (36px)	3 (39px)	3 (42px)	
spacing-9	extra-large	5.333 (64px)	5.333 (69.333px)	5.333 (74.667px)	
spacing-10	extra-large	8 (96px)	8 (104px)	8 (112px)	

## Usage guidelines

### Proximity

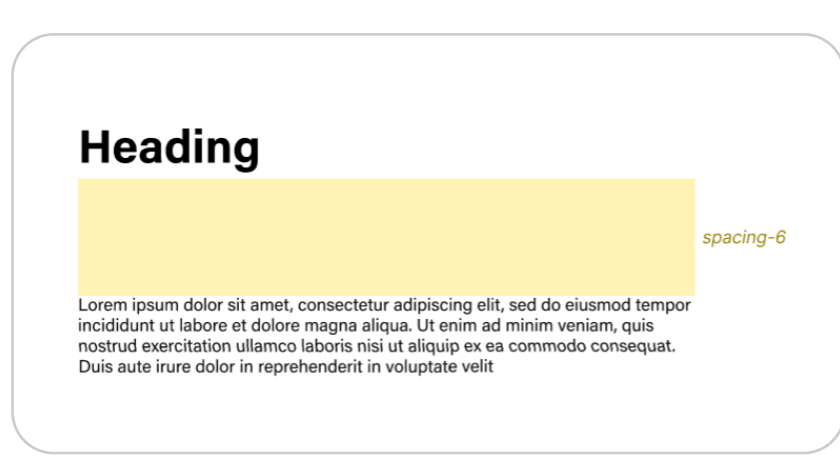
Place elements closer together that are part of a group such as those that share a relationship or context.

Example:



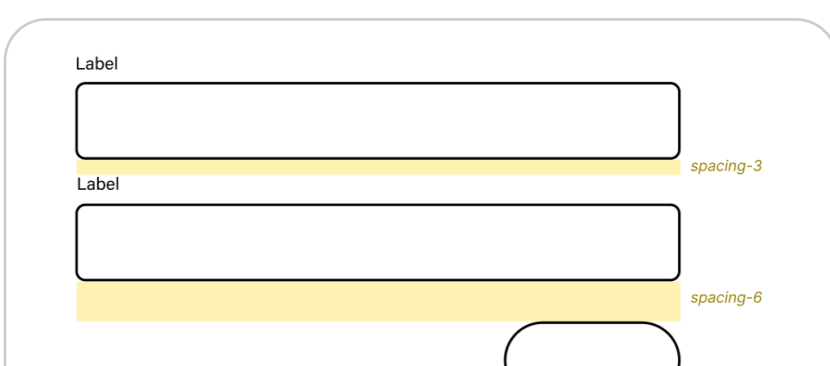
**Do**

Use small spacer (spacing-1 to spacing-4) when spacing UI that goes together



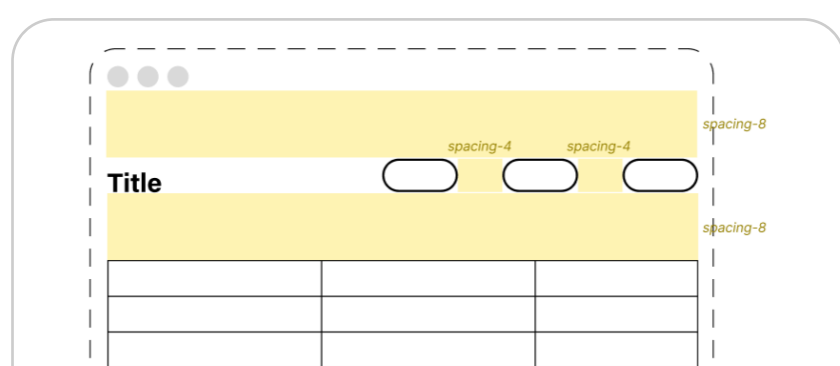
**Don't**

Avoid using large white spaces solely for visual purposes



**Do**

Double the spacing by using larger spaces for unrelated groups

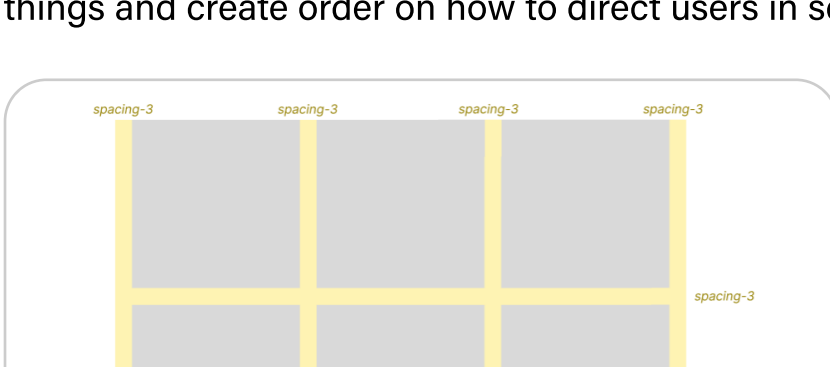


**Don't**

Don't overuse large spaces especially when screen real-estate is limited

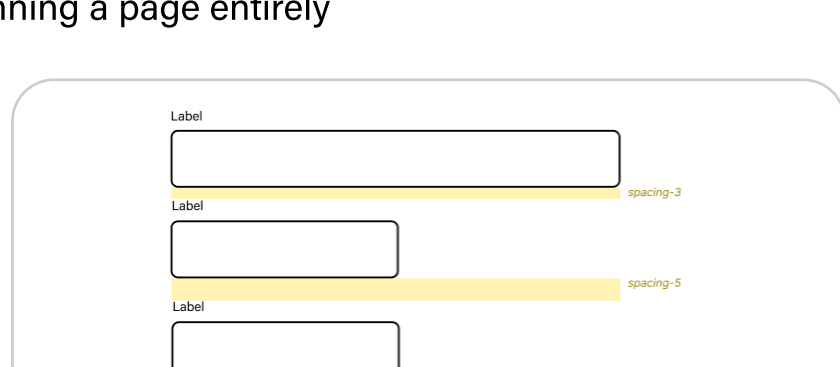
### Hierarchy

Spacing can also reinforce order of importance depending on the amount of whitespace is added in between. Varying the space between elements, sections and groups to drive focus on certain things and create order on how to direct users in scanning a page entirely



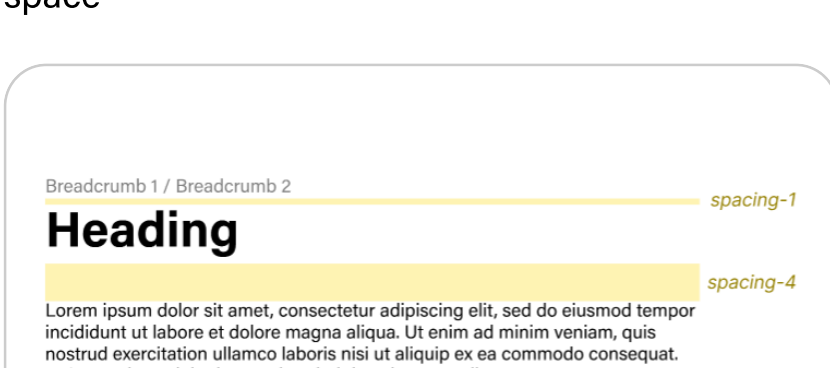
**Do**

Establish visual rhythm by using same amount of space



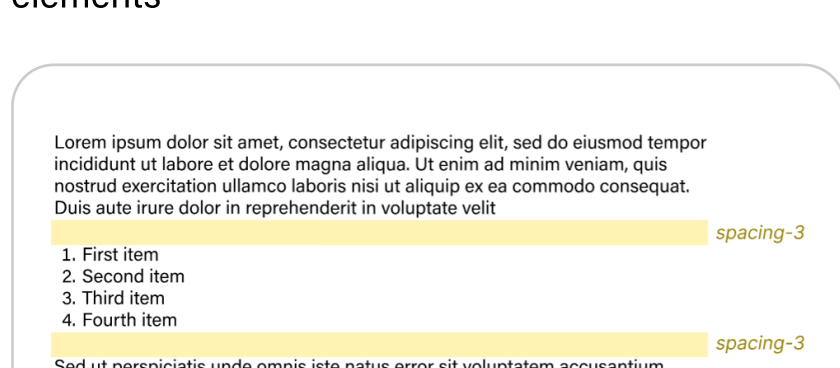
**Don't**

Avoid randomly using spacing values between elements



**Do**

Define visual and information hierarchy by mixing variation in spaces especially when working with content



**Don't**

For readability, avoid using multiple spacing values between paragraphs

## FAQ's

### Can we apply the spacing system to all spatial concepts?

Yes this should be applicable to paddings, margins, gaps and the likes for both vertical and horizontal orientation

### Is it possible to break-off from these values?

If necessary, going outside the system can be allowed but sparingly and should be done with caution

## Changelogs:

- 07/28/23 - Released initial docs (Raul)
- 09/28/23 - Updated and fixed values of mobile column of the Responsive scale table (Raul)



# Typography

- [Typefaces](#)
- [Weights](#)
- [Sizes](#)
- [Font Sets \(HTML Styles\)](#)
- [Usage guidelines](#)

## Typefaces

### Primary

The prevailing preference for Acumin in approximately 99% of instances rests upon its exceptional legibility and efficient utilization of page real estate. It's advisable to align Acumin with the H1 tags, as these represent primary headings, enhancing both prominence and readability. The versatility of Acumin lends itself to diverse applications, encompassing titles, headers, and standard text, thereby weaving a harmonious visual tapestry across our design spectrum.

## font-family-acumin Acumin Pro

### Marketing

We've opted for the Intro font to mirror Transunion's marketing materials, as seen on the business side of transunion.com. Our intention is to seamlessly match their use of Intro in headings and highlights, creating a cohesive visual presentation.

## font-family-intro Intro

### Weights

#### Acumin

font-thin	font-weight:300
font-italic	font-weight: 300
font-reg	font-weight: 400
font-med	font-weight: 500
font-bold	font-weight: 600
font-black	font-weight: 800

#### Intro

font-intro-thin	font-family: IntroLight
font-intro-reg	font-family: IntroRegular
font-intro-med	font-family: IntroSemibold
font-intro-bold	font-family: IntroBold

### Sizes

#### Base font size

We need to provide a way to reset or set a standard unit for every parent container inside each UI element. Below are the **responsive** values in **em** based units ([See Breakpoints](#)):

Token	Small	Medium	Large
font-size-base	12px	13px	14px

#### Scale

Token	Text Size em	Line Height em	Spacing
font-size-2xsmall	font-size-base	1	Initial
font-size-xsmall	1.167 (14px)	1.1	Initial
font-size-small	1.333 (16px)	1.2	Initial
font-size-medium	1.667 (20px)	1.2	Initial
font-size-large	2 (24px)	1.2	Initial
font-size-xlarge	2.667 (32px)	1.2	Initial
font-size-2xlarge	3.333 (40px)	1.2	Initial
font-size-3xlarge	4.333 (52px)	1.2	Initial
font-size-4xlarge	5.333 (64px)	1.2	Initial

## Font Sets (HTML styles)

Caption  
font-size: 1 (12px)  
line-height: 1 em  
font-weight: 300

Small  
font-size: 1 (12px)  
line-height: 1 em  
font-weight: 400

Body  
font-size: 1.167 (14px)  
line-height: 1.2 em  
font-weight: 400

#### Primary Headings

**Acumin H6**  
font-size: 1.167 (14px)  
line-height: 1.2 em  
font-weight: 600

**Acumin H5**  
font-size: 1.333 (16px)  
line-height: 1.2 em  
font-weight: 600

**Acumin H4**  
font-size: 1.667 (20px)  
line-height: 1.2 em  
font-weight: 600

**Acumin H3**  
font-size: 2 (24px)  
line-height: 1.2 em  
font-weight: 600

**Acumin H2**  
font-size: 2.667 (32px)  
line-height: 1.2 em  
font-weight: 600

**Acumin H1**  
font-size: 3.333 (40px)  
line-height: 1.2 em  
font-weight: 600

#### Marketing Headings

**Intro H6**  
font-size: 1.167 (14px)  
line-height: 1.2 em  
font-family: IntroRegular

**Intro H5**  
font-size: 1.333 (16px)  
line-height: 1.2 em  
font-family: IntroRegular

**Intro H4**  
font-size: 1.667 (20px)  
line-height: 1.2 em  
font-family: IntroRegular

**Intro H3**  
font-size: 2 (24px)  
line-height: 1.2 em  
font-family: IntroRegular

**Intro H2**  
font-size: 2.667 (32px)  
line-height: 1.2 em  
font-family: IntroSemiBold

**Intro H1**  
font-size: 3.333 (40px)  
line-height: 1.2 em  
font-family: IntroBold

## Usage guidelines

### Data Table Text Wrapping

Text wrapping involves formatting and adjusting text to fit within a given space, maintaining readability and visual appeal. It prevents overly long lines or cluttered layouts, enhancing the user's experience by making content easy to read and aesthetically pleasing. Effective text wrapping is crucial for optimal communication and user engagement.

Example:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod nllskl l sdl klsdkl kdsl klsdrnkl lsdrn lk
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmodnik nl nkl...

**Do**  
If black text is longer than the column width, wrap the text to the next line. Each row height should be wide enough to accommodate 2 lines of text even if there is only 1 line of text. If text is longer than 2 lines, wrap to next line. The row height should be able to accommodate the additional lines.

This is black text that is longer than 2 lines and it keeps going and going and g
This is black text that is longer than 2 lines and it keeps going and going and ...

**Don't**  
If black text is longer than 2 lines, don't use ellipses or cut off the text.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod nlsnlknl knlkdsnlsml kklsmllks
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod nlsnlknl knlkdsnlsml kklsmllks

**Do**  
For blue text, wrap to next line. The row height should change to accommodate the extra lines.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod mllkslmlln 12345678
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod mklsmllnn 12345678

**Don't**  
If text extends past 2 lines, don't use ellipses or cut off text.

### Data Table Text Alignment

Text alignment in data table columns refers to the positioning of text within each cell. It determines whether the text aligns to the left, right, or center of the cell, ensuring clear presentation and easy data comparison. Proper alignment enhances the visual structure of the table and aids in interpreting the information it contains.

Example:

Audience Name
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do

**Do**  
Text should always be left aligned to the column.

Audience Name
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod

**Don't**  
Never center align or anything besides left align.

Lorem ipsum dolor sit amet, connect
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod

**Do**  
If there is only 1 line of text, it should be aligned to the center of the row. The height of the row should still be able to accommodate 2 lines of text.

Lorem ipsum dolor sit amet, connect
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
adipiscing elit, sed do eiusmod

**Don't**  
Don't align to anything but the center of the row if there is only 1 line of text.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod 12345678
Lorem ipsum dolor sit amet, consectetur 12345678

**Do**  
In **"View Insights"**, the clickable blue text may span 2 lines and have an ID number below. The row height should be able to accommodate 2 lines and an ID number. If there is only 1 line of blue text and an ID number, it should be aligned to the center of the row.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod 12345678
Lorem ipsum dolor sit amet, consectetur 12345678

**Don't**  
If there is only 1 line of blue text, never align to top left or anything besides the center of the row.

Price CTV	Price Mobile
\$1.88	\$1.88
\$1.88	

**Do**  
Exceptions to align left in a column can be found in **"Taxonomy Management"**. Price content should be aligned to the right side of the header.

Price CTV	Price Mobile
\$1.88	\$1.88
\$1.88	\$1.88

**Don't**  
Do not align left.

### Data Table Date Formatting

Date formatting involves presenting dates in a consistent and intuitive manner, aiding user comprehension and interaction. This enhances user experience by ensuring clarity and ease of interpreting time-related information.

Example:

Date
October 23, 2023

**Do**  
In **"Data Audiences"**, **"Distribution"**, **"Data Collaboration >> Clean Room >> Multi Party Clean Room**, and **"Insights >> Attribute Builder"**, write in long date format. Using the long date format, such as "October 23, 2023," enhances clarity and reduces ambiguity for users, ensuring better understanding of the date information and facilitating smoother communication.

Date
10/23/2023
Oct 23, 2023

**Don't**  
Don't write in numeric/short date format or abbreviated date format.

Date
Oct 23, 2023

**Do**  
In **"Audience"** >> **"View Insights"**, **"Modeling"**, **"Marketplace"**, **"Data Collaboration >> Request Demo >> Manage Partners"**, write in abbreviated date format. Using an abbreviated date format offers a balance between brevity and clarity, aiding quick scanning and recognition of dates. This format optimizes space and maintains visual harmony, enhancing user experience by efficiently conveying time-related information without overwhelming the interface.

Date
10/23/2023
October 23, 2023
Oct 23, 2023

**Don't**  
Don't write in numeric/short date format, long date format, or abbreviated date format.

### Font Applications

Incorporating Intro font for brand consistency and Acumin font for optimal content readability on a website strikes a balance between maintaining a unified visual identity and ensuring a seamless user experience. Intro aligns with marketing materials, fostering recognition and brand cohesion, while Acumin prioritizes legibility and engagement for primary content consumption. This dual-font approach harmonizes brand aesthetics with effective communication, enhancing both brand image and user interaction.

Example:

Hello World
Hello World description goes here lorem ipsum dolor amet jklsdlkl kl dkjld jkldsjskl

**Do**  
This is for marketing content. To highlight headers in **"Data Collaboration"** and **"Audience >> Transunion Credit Attributes"**, use **"font-intro-bold font-weight: 600"**. Use **"Acumin font-thin font-weight: 300"** for body content

Hello World
Hello World description goes here lorem ipsum dolor amet jklsdlkl kl dkjld jkldsjskl nd lsdkl klsdn kln lksdnkln lks nllkn kl dnllkds

**Don't**  
Intro should never be used in paragraphs

Hello World
Hello World
Hello World
Hello World

**Do**  
Acumin can be used for small headers and large amounts of content like in data tables

Hello World
Hello World
Hello World
Hello World

**Don't**  
Don't use Intro for large amounts of content

Audience Name
Description goes here

**Do**  
For non-marketing content pertaining to main information, use **"font-intro-reg font-weight: IntroRegular"** for the header and **"Acumin font-thin font-weight: 300"** for descriptions.

Audience Name
Description goes here

**Don't**  
Unlike in **"Data Collaboration"** don't use **"font-intro-bold font-weight: 600"** for non-marketing headers

Examples: **"Welcome"** page cards and **"View Audiences >> Loves Trucks and Yoga"** color-blue-dark card.

### Changelogs:

- 06/28/23 - Released initial docs (EI)
- 08/18/23 - Updated name and values for Scales section (EI)
- 08/22/23 - Updated content for base font size and scales (Raul)
- 08/28/23 - Added new sections for Primary Headings and Marketing Headings under HTML styles (Raul)
- 09/08/23 - Updated values for base font size table (Raul)
- 10/24/23 - Fix doc content for small tag (Raul)