

Emil Kowalski
Design Engineer

When done right, animations make an interface feel predictable, faster, and more enjoyable to use. They help you and your product stand out.

But they can also do the opposite. They can make an interface feel unpredictable, slow, and annoying. They can even make your users lose trust in your product.

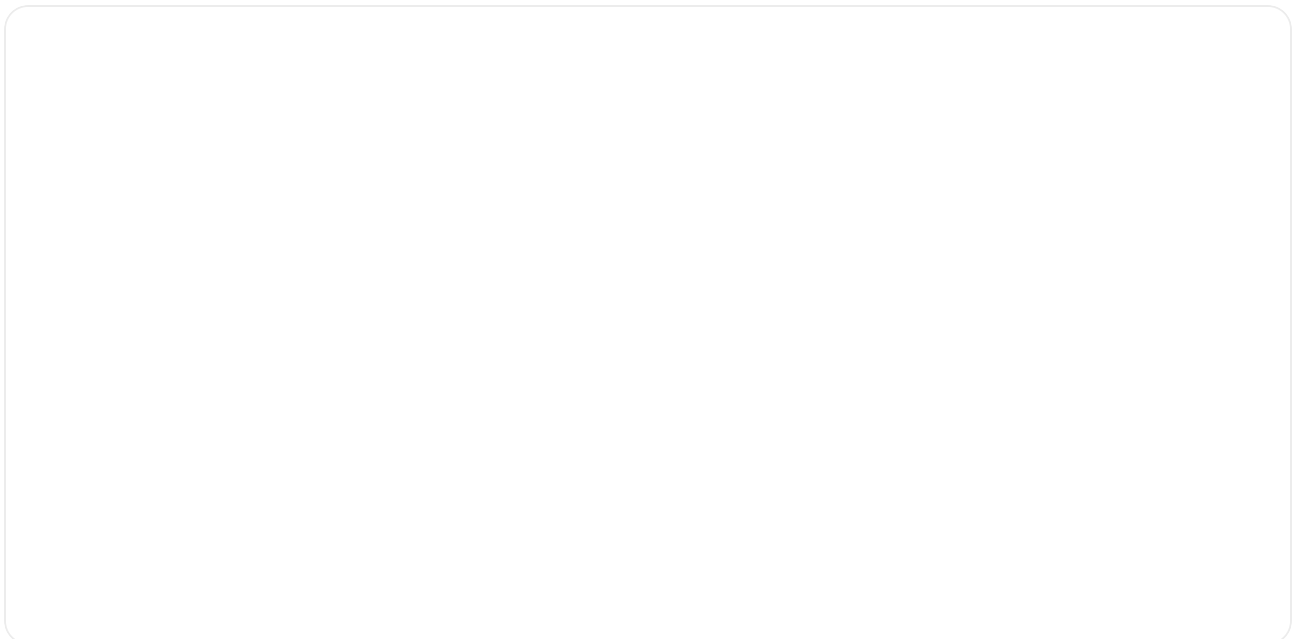
So how do you know *when* and *how* to animate to improve the experience?

Step one is making sure your animations have a purpose.

Purposeful animations

Before you start animating, ask yourself: what's the purpose of this animation?

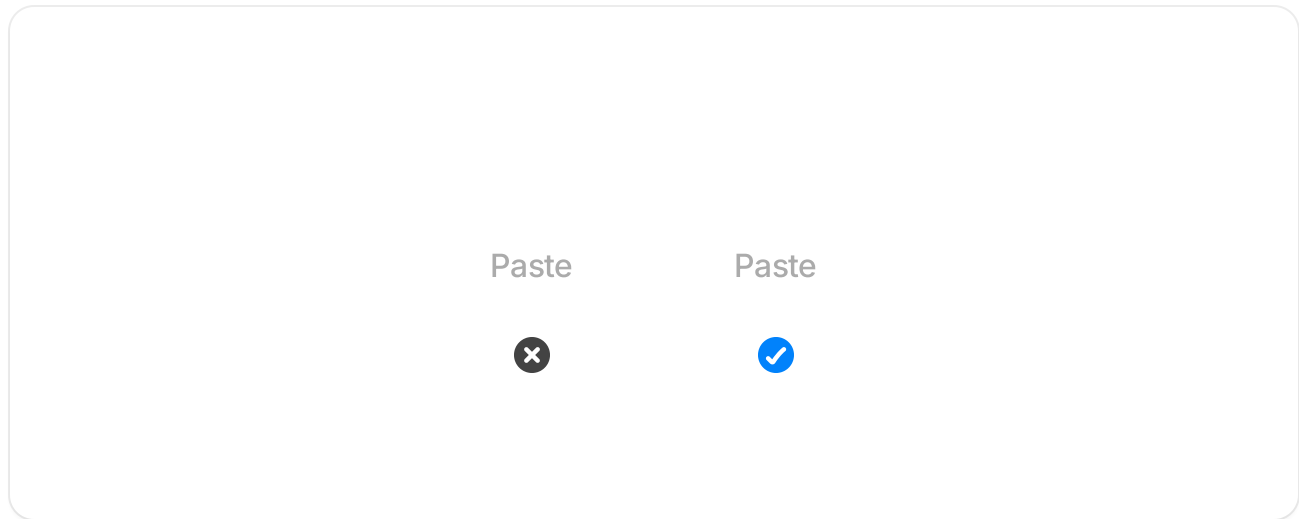
As an example, what's the purpose of this marketing animation we built at Linear?



You can view the full animation on linear.app/ai.

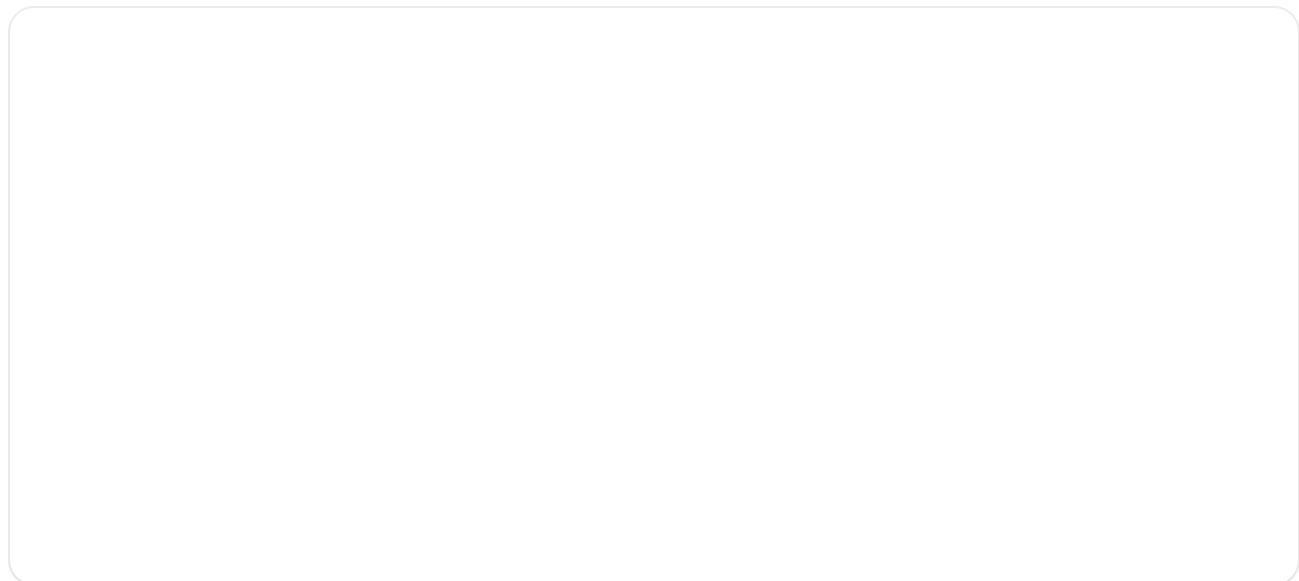
This animation explains how Product Intelligence (Linear’s feature) works. We could have used a static asset, but the animated version helps the user understand what this feature does, straight in the initial viewport of the page.

Another purposeful animation is this subtle scale down effect when pressing a button. It’s a small thing, but it helps the interface feel more alive and responsive.



Sonner’s enter animation, on the other hand, has two purposes:

- Having a toast suddenly appear would feel off, so we animate it in.
- Because it comes from and leaves in the same direction, it creates spatial consistency, making the swipe-down-to-dismiss gesture feel more intuitive.



But sometimes the purpose of an animation might just be to bring delight.

Morphing of the feedback component below helps make the experience more unique and memorable. This works as long as the user will rarely interact with it. It'll then become a pleasant surprise, rather than a daily annoyance.



Press on the button to see it morph.

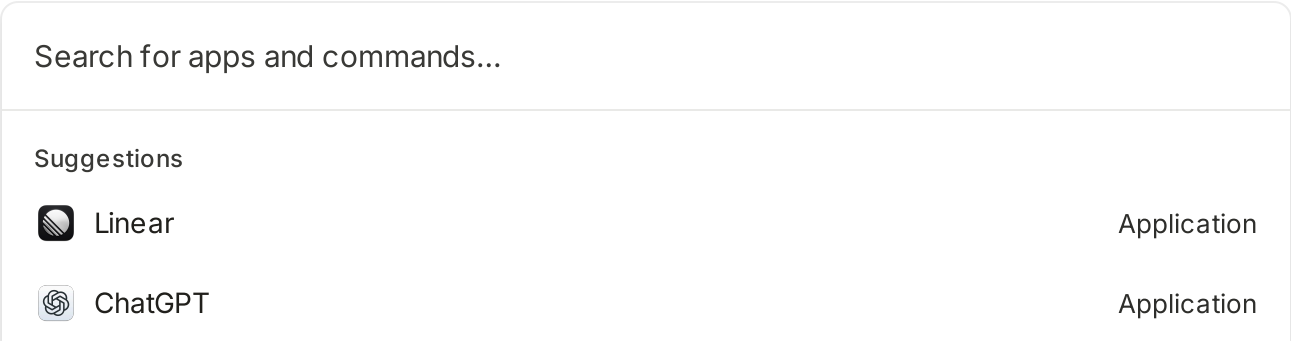
Used multiple times a day, this component would quickly become irritating. The initial delight would fade and the animation would slow users down.

How often users will see an animation is a key factor in deciding whether to animate or not. Let's dive deeper into it next.

Frequency of use

I use Raycast hundreds of times a day. If it animated every time I opened it, it would be *very* annoying. But there's no animation at all. That's the optimal experience.

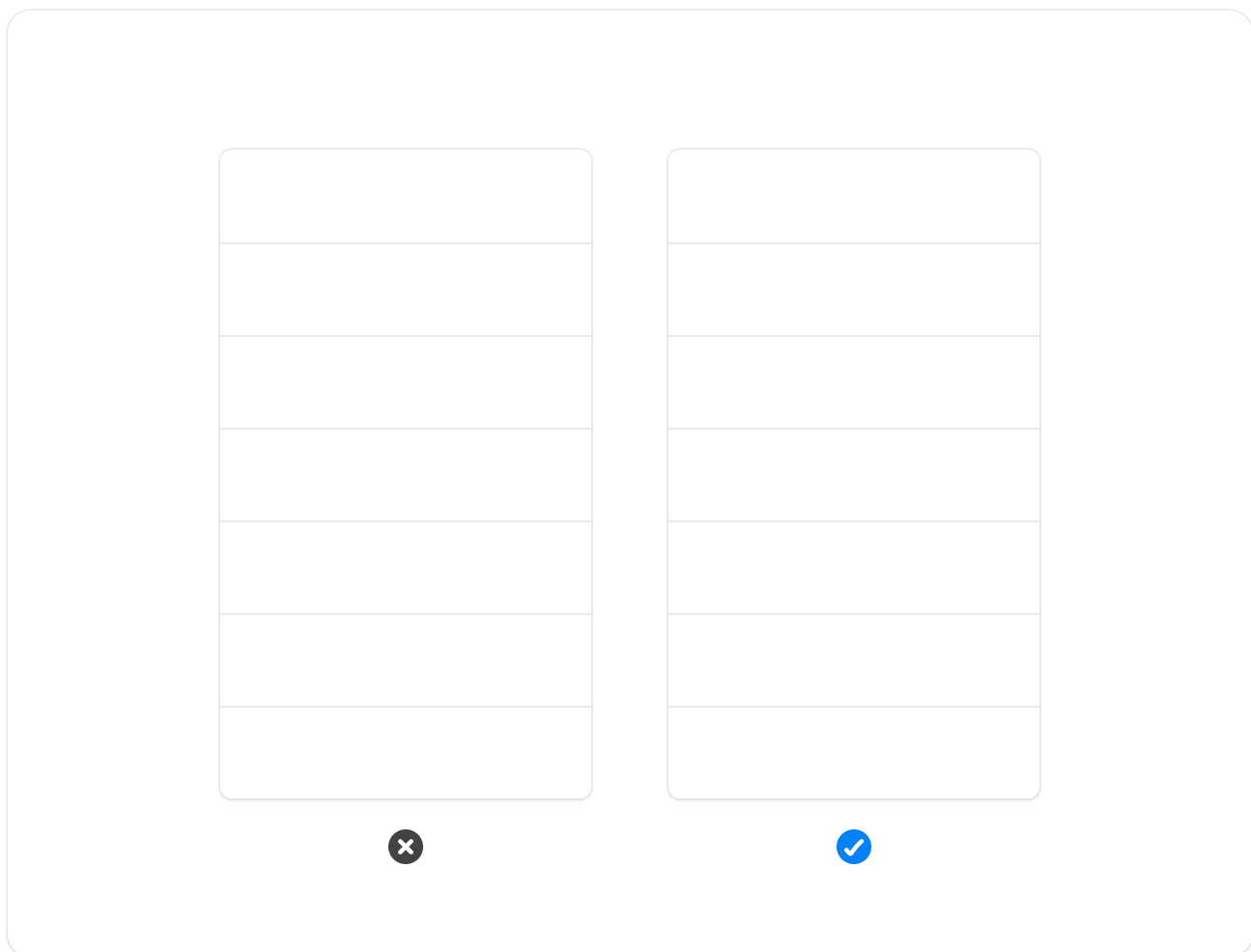
To see it for yourself, try to toggle the open state of the menu below by pressing **J** and then **K**. Which one feels better if used hundreds of times a day?



 Cursor	Application
 Figma	Application
 Obsidian	Application
Commands	
Clipboard History	Command
Emoji Picker	Command
	Open Application ⌘

When I open Raycast, I have a clear goal in mind. I don't expect to be "delighted", I don't need to be. I just want to do my work with no unnecessary friction.

Think about what the user wants to achieve and how often they will see an animation. A hover effect is nice, but if used multiple times a day, it would likely benefit the most from having no animation at all.



Imagine you interact with this list often during the day.

The same goes for keyboard-initiated actions. These actions may be repeated hundreds of times a day, an animation would make them feel slow, delayed, and disconnected from the user's actions. You should *never* animate them.

To see it for yourself, focus on the input below and use arrow keys to navigate through the list. Notice how the highlight feels delayed compared to the keys you press. Now press  (shift) and see how this interaction feels without animation.

Search for apps and commands...

Suggestions

 Linear

Application

 ChatGPT

Application

 Cursor

Application

 Figma

Application

 Obsidian

Application

Commands

Clipboard History

Command

Emoji Picker

Command



Open Application 

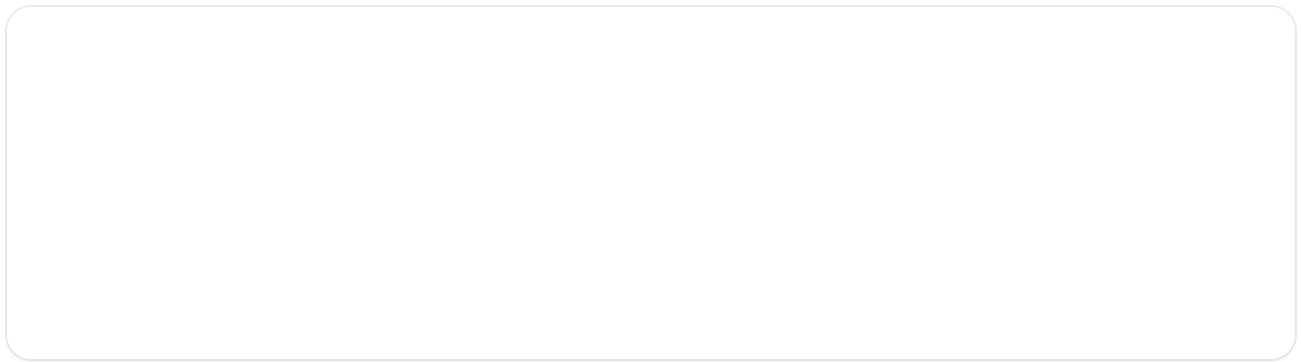
Press shift to toggle the animation

But even if your animation won't be used too often and it fulfills a clear purpose, you still have to think about its speed...

Perception of speed

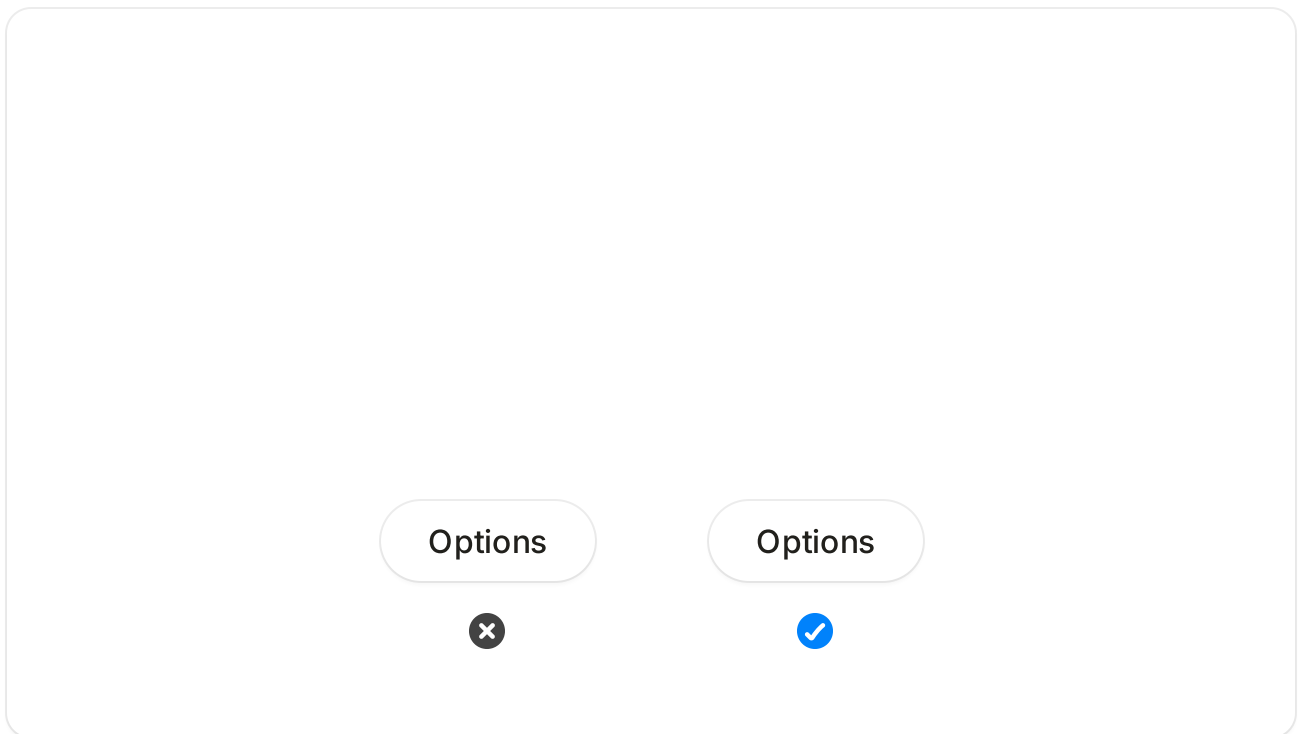
Unless you are working on marketing sites, your animations *have* to be fast. They improve the perceived performance of your app, stay connected to user's actions, and make the interface feel as if it's truly listening to the user.

To give you an example, a faster-spinning spinner makes the app seem to load faster, even though the load time is the same. This improves perceived performance.



Which one works harder to load the data?

A **180ms** dropdown animation feels more responsive than a **400ms** one:



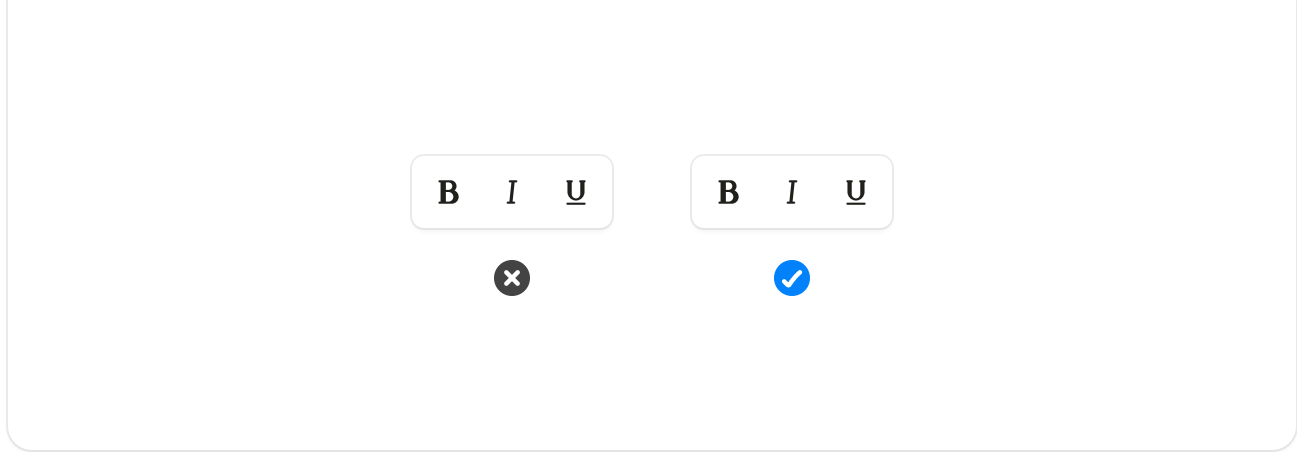
Click on the buttons to compare the speed.

As a rule of thumb, UI animations should generally stay under **300ms**.

Another example of the importance of speed: tooltips should have a slight delay before appearing to prevent accidental activation. Once a tooltip is open however, hovering over other tooltips should open them with no delay and no animation.

This feels faster without defeating the purpose of the initial delay.





Radix UI and Base UI skip the delay once a tooltip is shown.

Building great interfaces

The goal is not to animate for animation's sake, it's to build great user interfaces. The ones that users will happily use, even on a daily basis. Sometimes this requires animations, but sometimes the best animation is no animation.

Knowing when to animate is just one of many things you need to know in order to craft great animations. If you'd like to dive deeper into the theory and practice of it, I've created a course that covers everything you need to know:

→ [Check out "Animations on the Web"](#)