Introduction to animations

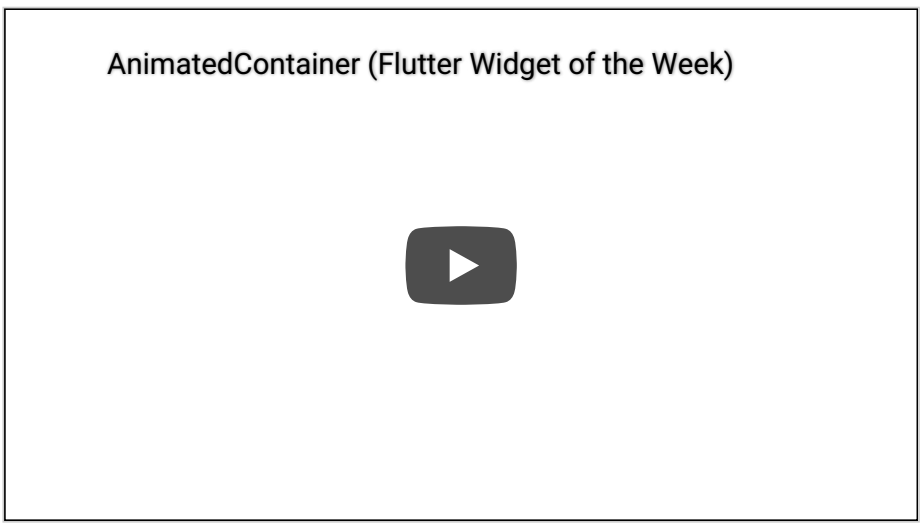
Contents

Well-designed animations make a UI feel more intuitive, contribute to the slick look and feel of a polished app, and improve the us experience. Flutter's animation support makes it easy to implement a variety of animation types. Many widgets, especially Materi widgets, come with the standard motion effects defined in their design spec, but it's also possible to customize these effects.

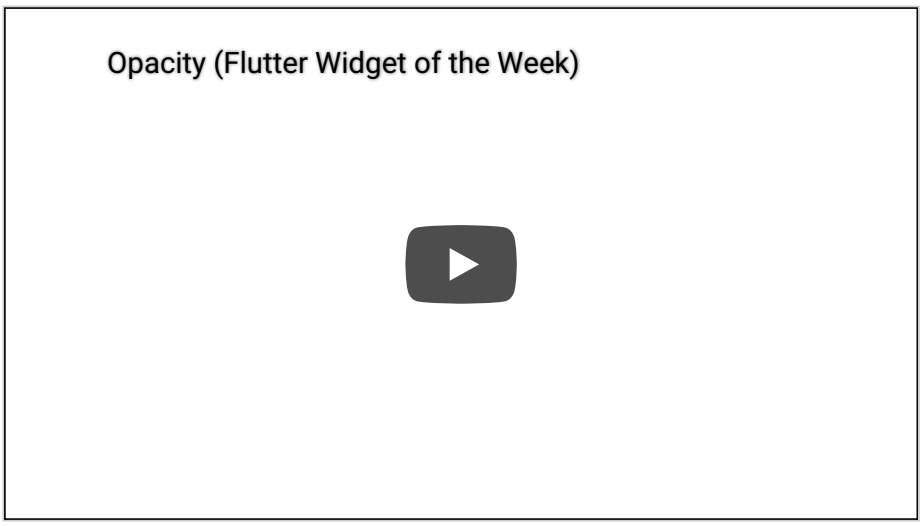
The following resources are a good place to start learning the Flutter animation framework. Each of these documents shows, ste step, how to write animation code.

- Implicit animations codelab
  Covers how to use implicit animations using step-by-step instructions and interactive examples.
- Animations tutorial
  Explains the fundamental classes in the Flutter animation package (controllers, Animatable, curves, listeners, builders), as it guides you through a progression of tween animations using different aspects of the animation APIs.
- Zero to One with Flutter, part 1 and part 2
  Medium articles showing how to create an animated chart using tweening.
- Building Beautiful UIs with Flutter
  Codelab demonstrating how to build a simple chat app. Step 7 (Animate your app) shows how to animate the new message sliding it from the input area up to the message list.

We also have some videos that discuss aspects of Flutter animation.

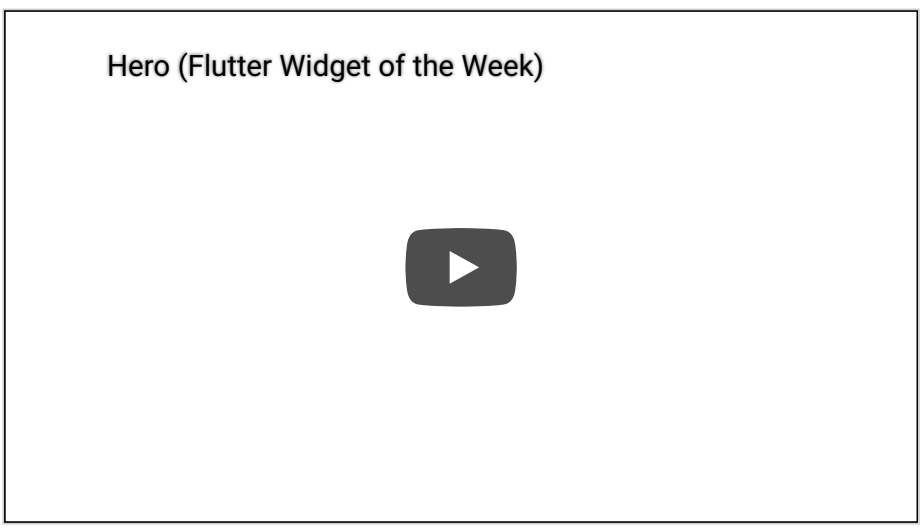


AnimatedContainer

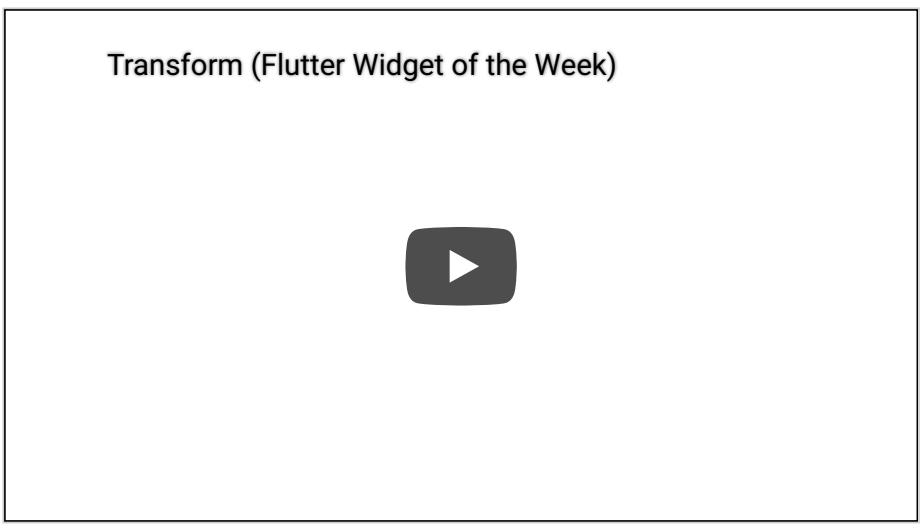


Opacity, including the implicit AnimatedOpacity widget

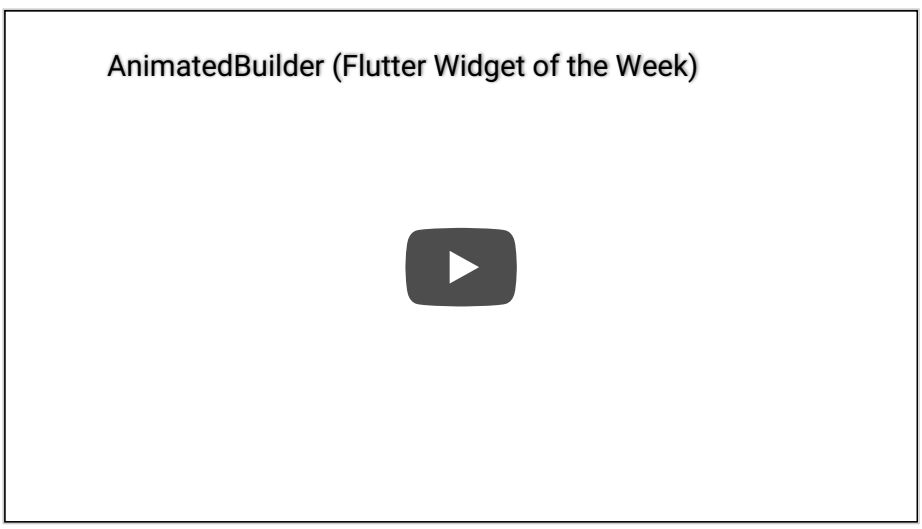
FadeInImage



Hero



Transform



AnimatedBuilder

Animation types

Animations fall into one of two categories: tween- or physics-based. The following sections explain what these terms mean, and point you to resources where you can learn more. In some cases, the best documentation we currently have is example code in th Flutter gallery.

Tween animation

Short for *in-betweening*. In a tween animation, the beginning and ending points are defined, as well as a timeline, and a curve that defines the timing and speed of the transition. The framework calculates how to transition from the beginning point to the end po

The documents listed above, such as the animations tutorial are not about tweening, specifically, but they use tweens in their examples.

Physics-based animation

In physics-based animation, motion is modeled to resemble real-world behavior. When you toss a ball, for example, where and wh lands depends on how fast it was tossed and how far it was from the ground. Similarly, dropping a ball attached to a spring falls (bounces) differently than dropping a ball attached to a string.

- Flutter Gallery
  Under **Material Components**, the Grid example demonstrates a fling animation. Select one of the images from the grid and zoom in. You can pan the image with flinging or dragging gestures.
- Also see the API documentation for AnimationController.animateWith and SpringSimulation.

Common animation patterns

Most UX or motion designers find that certain animation patterns are used repeatedly when designing a UI. This section lists som the commonly used animation patterns, and tells you where you can learn more.

Animated list or grid

This pattern involves animating the addition or removal of elements from a list or grid.

- AnimatedList example
  This demo, from the Sample App Catalog, shows how to animate adding an element to a list, or removing a selected elemer The internal Dart list is synced as the user modifies the list using the plus (+) and minus (-) buttons.

Shared element transition

In this pattern, the user selects an element—often an image—from the page, and the UI animates the selected element to a new p with more detail. In Flutter, you can easily implement shared element transitions between routes (pages) using the Hero widget.

- Hero Animations How to create two styles of Hero animations:
  - The hero flies from one page to another while changing position and size.
  - The hero's boundary changes shape, from a circle to a square, as its flies from one page to another.
- Flutter Gallery
  You can build the Gallery app yourself, or download it from the Play Store. The Shrine demo includes an example of a Hero animation.
- Also see the API documentation for the Hero, Navigator, and PageRoute classes.

Staggered animation

Animations that are broken into smaller motions, where some of the motion is delayed. The smaller animations might be sequent or might partially or completely overlap.

- Staggered Animations

Other resources

Learn more about Flutter animations at the following links:

- Animations: Technical Overview
  A look at some of the major classes in the animations library, and Flutter's animation architecture.
- Animation and Motion Widgets
  A catalog of some of the animation widgets provided in the Flutter APIs.

If there is specific animation documentation you'd like to see, file an issue.