Flutter resumen documentación

# Animation and MOTION WIDGETS

## animated align

Animated version of [Align](https://api.flutter.dev/flutter/widgets/Align-class.html) which automatically transitions the child's position over a given duration whenever the given [alignment](https://api.flutter.dev/flutter/widgets/AnimatedAlign/alignment.html) changes.

Here's an illustration of what this can look like, using a [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) of [Curves.fastOutSlowIn](https://api.flutter.dev/flutter/animation/Curves/fastOutSlowIn-constant.html).

For the animation, you can choose a [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) as well as a [duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) and the widget will automatically animate to the new target [alignment](https://api.flutter.dev/flutter/widgets/AnimatedAlign/alignment.html). If you require more control over the animation (e.g. if you want to stop it mid-animation), consider using an [AlignTransition](https://api.flutter.dev/flutter/widgets/AlignTransition-class.html) instead, which takes a provided [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) as argument. While that allows you to fine-tune the animation, it also requires more development overhead as you have to manually manage the lifecycle of the underlying [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html).

See also:

* [AnimatedContainer](https://api.flutter.dev/flutter/widgets/AnimatedContainer-class.html), which can transition more values at once.
* [AnimatedPadding](https://api.flutter.dev/flutter/widgets/AnimatedPadding-class.html), which can animate the padding instead of the alignment.
* [AnimatedSlide](https://api.flutter.dev/flutter/widgets/AnimatedSlide-class.html), which can animate the translation of child by a given offset relative to its size.
* [AnimatedPositioned](https://api.flutter.dev/flutter/widgets/AnimatedPositioned-class.html), which, as a child of a [Stack](https://api.flutter.dev/flutter/widgets/Stack-class.html), automatically transitions its child's position over a given duration whenever the given position changes.

### Constructors

[AnimatedAlign](https://api.flutter.dev/flutter/widgets/AnimatedAlign/AnimatedAlign.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html) alignment, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? heightFactor, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? widthFactor, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onEnd})

Creates a widget that positions its child by an alignment that animates implicitly.

const

### Properties

[alignment](https://api.flutter.dev/flutter/widgets/AnimatedAlign/alignment.html) → [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html)

How to align the child.

final

[child](https://api.flutter.dev/flutter/widgets/AnimatedAlign/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

final

[curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The curve to apply when animating the parameters of this container.

finalinherited

[duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration over which to animate the parameters of this container.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[heightFactor](https://api.flutter.dev/flutter/widgets/AnimatedAlign/heightFactor.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

If non-null, sets its height to the child's height multiplied by this factor.

final

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[onEnd](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/onEnd.html) → [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)?

Called every time an animation completes.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[widthFactor](https://api.flutter.dev/flutter/widgets/AnimatedAlign/widthFactor.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

If non-null, sets its width to the child's width multiplied by this factor.

final

## animated builder

A general-purpose widget for building animations.

[AnimatedBuilder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder-class.html) is useful for more complex widgets that wish to include an animation as part of a larger build function. To use [AnimatedBuilder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder-class.html), construct the widget and pass it a builder function.

For simple cases without additional state, consider using [AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html).

If the [builder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder/builder.html) function contains a subtree that does not depend on the animation passed to the constructor, it's more efficient to build that subtree once instead of rebuilding it on every animation tick.

If a pre-built subtree is passed as the [child](https://api.flutter.dev/flutter/widgets/ListenableBuilder/child.html) parameter, the [AnimatedBuilder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder-class.html) will pass it back to the [builder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder/builder.html) function so that it can be incorporated into the build.

Using this pre-built child is entirely optional, but can improve performance significantly in some cases and is therefore a good practice.

See also:

* [ListenableBuilder](https://api.flutter.dev/flutter/widgets/ListenableBuilder-class.html), a widget with similar functionality, but named more appropriately for a builder triggered on changes in [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)s that aren't [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)s.
* [TweenAnimationBuilder](https://api.flutter.dev/flutter/widgets/TweenAnimationBuilder-class.html), which animates a property to a target value without requiring manual management of an [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html).

Inheritance

* [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)
* [DiagnosticableTree](https://api.flutter.dev/flutter/foundation/DiagnosticableTree-class.html)
* [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)
* [StatefulWidget](https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html)
* [AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html)
* [ListenableBuilder](https://api.flutter.dev/flutter/widgets/ListenableBuilder-class.html)
* AnimatedBuilder

### Constructors

[AnimatedBuilder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder/AnimatedBuilder.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) animation, required [TransitionBuilder](https://api.flutter.dev/flutter/widgets/TransitionBuilder.html) builder, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child})

Creates an animated builder.

const

### Properties

[animation](https://api.flutter.dev/flutter/widgets/AnimatedBuilder/animation.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) supplied to the constructor (typically an [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)).

read-only

[builder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder/builder.html) → [TransitionBuilder](https://api.flutter.dev/flutter/widgets/TransitionBuilder.html)

Called every time the [animation](https://api.flutter.dev/flutter/widgets/AnimatedBuilder/animation.html) notifies about a change.

read-onlyoverride

[child](https://api.flutter.dev/flutter/widgets/ListenableBuilder/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The child widget to pass to the [builder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder/builder.html).

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedBuilder/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) supplied to the constructor (typically an [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)).

read-onlyoverride

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## animated container

Animated version of [Container](https://api.flutter.dev/flutter/widgets/Container-class.html) that gradually changes its values over a period of time.

The [AnimatedContainer](https://api.flutter.dev/flutter/widgets/AnimatedContainer-class.html) will automatically animate between the old and new values of properties when they change using the provided curve and duration. Properties that are null are not animated. Its child and descendants are not animated.

This class is useful for generating simple implicit transitions between different parameters to [Container](https://api.flutter.dev/flutter/widgets/Container-class.html) with its internal [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html). For more complex animations, you'll likely want to use a subclass of [AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html) such as the [DecoratedBoxTransition](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition-class.html) or use your own [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html).

ee also:

* [AnimatedPadding](https://api.flutter.dev/flutter/widgets/AnimatedPadding-class.html), which is a subset of this widget that only supports animating the [padding](https://api.flutter.dev/flutter/widgets/AnimatedContainer/padding.html).
* The [catalog of layout widgets](https://flutter.dev/widgets/layout/).
* [AnimatedPositioned](https://api.flutter.dev/flutter/widgets/AnimatedPositioned-class.html), which, as a child of a [Stack](https://api.flutter.dev/flutter/widgets/Stack-class.html), automatically transitions its child's position over a given duration whenever the given position changes.
* [AnimatedAlign](https://api.flutter.dev/flutter/widgets/AnimatedAlign-class.html), which automatically transitions its child's position over a given duration whenever the given [AnimatedAlign.alignment](https://api.flutter.dev/flutter/widgets/AnimatedAlign/alignment.html) changes.
* [AnimatedSwitcher](https://api.flutter.dev/flutter/widgets/AnimatedSwitcher-class.html), which switches out a child for a new one with a customizable transition.
* [AnimatedCrossFade](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade-class.html), which fades between two children and interpolates their sizes.

Inheritance

* [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)
* [DiagnosticableTree](https://api.flutter.dev/flutter/foundation/DiagnosticableTree-class.html)
* [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)
* [StatefulWidget](https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html)
* [ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget-class.html)
* AnimatedContainer

### Constructors

[AnimatedContainer](https://api.flutter.dev/flutter/widgets/AnimatedContainer/AnimatedContainer.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html)? alignment, [EdgeInsetsGeometry](https://api.flutter.dev/flutter/painting/EdgeInsetsGeometry-class.html)? padding, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? color, [Decoration](https://api.flutter.dev/flutter/painting/Decoration-class.html)? decoration, [Decoration](https://api.flutter.dev/flutter/painting/Decoration-class.html)? foregroundDecoration, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? width, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? height, [BoxConstraints](https://api.flutter.dev/flutter/rendering/BoxConstraints-class.html)? constraints, [EdgeInsetsGeometry](https://api.flutter.dev/flutter/painting/EdgeInsetsGeometry-class.html)? margin, [Matrix4](https://api.flutter.dev/flutter/vector_math_64/Matrix4-class.html)? transform, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html)? transformAlignment, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child, [Clip](https://api.flutter.dev/flutter/dart-ui/Clip.html) clipBehavior = Clip.none, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onEnd})

Creates a container that animates its parameters implicitly.

### Properties

[alignment](https://api.flutter.dev/flutter/widgets/AnimatedContainer/alignment.html) → [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html)?

Align the [child](https://api.flutter.dev/flutter/widgets/AnimatedContainer/child.html) within the container.

final

[child](https://api.flutter.dev/flutter/widgets/AnimatedContainer/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The [child](https://api.flutter.dev/flutter/widgets/AnimatedContainer/child.html) contained by the container.

final

[clipBehavior](https://api.flutter.dev/flutter/widgets/AnimatedContainer/clipBehavior.html) → [Clip](https://api.flutter.dev/flutter/dart-ui/Clip.html)

The clip behavior when [AnimatedContainer.decoration](https://api.flutter.dev/flutter/widgets/AnimatedContainer/decoration.html) is not null.

final

[constraints](https://api.flutter.dev/flutter/widgets/AnimatedContainer/constraints.html) → [BoxConstraints](https://api.flutter.dev/flutter/rendering/BoxConstraints-class.html)?

Additional constraints to apply to the child.

final

[curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The curve to apply when animating the parameters of this container.

finalinherited

[decoration](https://api.flutter.dev/flutter/widgets/AnimatedContainer/decoration.html) → [Decoration](https://api.flutter.dev/flutter/painting/Decoration-class.html)?

The decoration to paint behind the [child](https://api.flutter.dev/flutter/widgets/AnimatedContainer/child.html).

final

[duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration over which to animate the parameters of this container.

finalinherited

[foregroundDecoration](https://api.flutter.dev/flutter/widgets/AnimatedContainer/foregroundDecoration.html) → [Decoration](https://api.flutter.dev/flutter/painting/Decoration-class.html)?

The decoration to paint in front of the child.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[margin](https://api.flutter.dev/flutter/widgets/AnimatedContainer/margin.html) → [EdgeInsetsGeometry](https://api.flutter.dev/flutter/painting/EdgeInsetsGeometry-class.html)?

Empty space to surround the [decoration](https://api.flutter.dev/flutter/widgets/AnimatedContainer/decoration.html) and [child](https://api.flutter.dev/flutter/widgets/AnimatedContainer/child.html).

final

[onEnd](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/onEnd.html) → [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)?

Called every time an animation completes.

finalinherited

[padding](https://api.flutter.dev/flutter/widgets/AnimatedContainer/padding.html) → [EdgeInsetsGeometry](https://api.flutter.dev/flutter/painting/EdgeInsetsGeometry-class.html)?

Empty space to inscribe inside the [decoration](https://api.flutter.dev/flutter/widgets/AnimatedContainer/decoration.html). The [child](https://api.flutter.dev/flutter/widgets/AnimatedContainer/child.html), if any, is placed inside this padding.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[transform](https://api.flutter.dev/flutter/widgets/AnimatedContainer/transform.html) → [Matrix4](https://api.flutter.dev/flutter/vector_math_64/Matrix4-class.html)?

The transformation matrix to apply before painting the container.

final

[transformAlignment](https://api.flutter.dev/flutter/widgets/AnimatedContainer/transformAlignment.html) → [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html)?

The alignment of the origin, relative to the size of the container, if [transform](https://api.flutter.dev/flutter/widgets/AnimatedContainer/transform.html) is specified.

final

## animated cross fade

A widget that cross-fades between two given children and animates itself between their sizes.

The animation is controlled through the [crossFadeState](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/crossFadeState.html) parameter. [firstCurve](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/firstCurve.html) and [secondCurve](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/secondCurve.html) represent the opacity curves of the two children. The [firstCurve](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/firstCurve.html) is inverted, i.e. it fades out when providing a growing curve like [Curves.linear](https://api.flutter.dev/flutter/animation/Curves/linear-constant.html). The [sizeCurve](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/sizeCurve.html) is the curve used to animate between the size of the fading-out child and the size of the fading-in child.

This widget is intended to be used to fade a pair of widgets with the same width. In the case where the two children have different heights, the animation crops overflowing children during the animation by aligning their top edge, which means that the bottom will be clipped.

The animation is automatically triggered when an existing [AnimatedCrossFade](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade-class.html) is rebuilt with a different value for the [crossFadeState](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/crossFadeState.html) property.

### Constructors

[AnimatedCrossFade](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/AnimatedCrossFade.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) firstChild, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) secondChild, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) firstCurve = Curves.linear, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) secondCurve = Curves.linear, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) sizeCurve = Curves.linear, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html) alignment = Alignment.topCenter, required [CrossFadeState](https://api.flutter.dev/flutter/widgets/CrossFadeState.html) crossFadeState, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)? reverseDuration, [AnimatedCrossFadeBuilder](https://api.flutter.dev/flutter/widgets/AnimatedCrossFadeBuilder.html) layoutBuilder = defaultLayoutBuilder, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excludeBottomFocus = true})

Creates a cross-fade animation widget.

const

### Properties

[alignment](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/alignment.html) → [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html)

How the children should be aligned while the size is animating.

final

[crossFadeState](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/crossFadeState.html) → [CrossFadeState](https://api.flutter.dev/flutter/widgets/CrossFadeState.html)

The child that will be shown when the animation has completed.

final

[duration](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration of the whole orchestrated animation.

final

[excludeBottomFocus](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/excludeBottomFocus.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

When true, this is equivalent to wrapping the bottom widget with an [ExcludeFocus](https://api.flutter.dev/flutter/widgets/ExcludeFocus-class.html) widget while it is at the bottom of the cross-fade stack.

final

[firstChild](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/firstChild.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The child that is visible when [crossFadeState](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/crossFadeState.html) is [CrossFadeState.showFirst](https://api.flutter.dev/flutter/widgets/CrossFadeState.html). It fades out when transitioning [crossFadeState](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/crossFadeState.html) from [CrossFadeState.showFirst](https://api.flutter.dev/flutter/widgets/CrossFadeState.html) to [CrossFadeState.showSecond](https://api.flutter.dev/flutter/widgets/CrossFadeState.html) and vice versa.

final

[firstCurve](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/firstCurve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The fade curve of the first child.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[layoutBuilder](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/layoutBuilder.html) → [AnimatedCrossFadeBuilder](https://api.flutter.dev/flutter/widgets/AnimatedCrossFadeBuilder.html)

A builder that positions the [firstChild](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/firstChild.html) and [secondChild](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/secondChild.html) widgets.

final

[reverseDuration](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/reverseDuration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)?

The duration of the whole orchestrated animation when running in reverse.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[secondChild](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/secondChild.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The child that is visible when [crossFadeState](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/crossFadeState.html) is [CrossFadeState.showSecond](https://api.flutter.dev/flutter/widgets/CrossFadeState.html). It fades in when transitioning [crossFadeState](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/crossFadeState.html) from [CrossFadeState.showFirst](https://api.flutter.dev/flutter/widgets/CrossFadeState.html) to [CrossFadeState.showSecond](https://api.flutter.dev/flutter/widgets/CrossFadeState.html) and vice versa.

final

[secondCurve](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/secondCurve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The fade curve of the second child.

final

[sizeCurve](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade/sizeCurve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The curve of the animation between the two children's sizes.

final

## animated default text style

Animated version of [DefaultTextStyle](https://api.flutter.dev/flutter/widgets/DefaultTextStyle-class.html) which automatically transitions the default text style (the text style to apply to descendant [Text](https://api.flutter.dev/flutter/widgets/Text-class.html) widgets without explicit style) over a given duration whenever the given style changes.

The [textAlign](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/textAlign.html), [softWrap](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/softWrap.html), [overflow](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/overflow.html), [maxLines](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/maxLines.html), [textWidthBasis](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/textWidthBasis.html) and [textHeightBehavior](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/textHeightBehavior.html) properties are not animated and take effect immediately when changed.

Here's an illustration of what using this widget looks like, using a [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) of [Curves.elasticInOut](https://api.flutter.dev/flutter/animation/Curves/elasticInOut-constant.html).

For the animation, you can choose a [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) as well as a [duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) and the widget will automatically animate to the new default text style. If you require more control over the animation (e.g. if you want to stop it mid-animation), consider using a [DefaultTextStyleTransition](https://api.flutter.dev/flutter/widgets/DefaultTextStyleTransition-class.html) instead, which takes a provided [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) as argument. While that allows you to fine-tune the animation, it also requires more development overhead as you have to manually manage the lifecycle of the underlying [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html).

Inheritance

* [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)
* [DiagnosticableTree](https://api.flutter.dev/flutter/foundation/DiagnosticableTree-class.html)
* [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)
* [StatefulWidget](https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html)
* [ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget-class.html)
* AnimatedDefaultTextStyle

### Constructors

[AnimatedDefaultTextStyle](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/AnimatedDefaultTextStyle.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) child, required [TextStyle](https://api.flutter.dev/flutter/painting/TextStyle-class.html) style, [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html)? textAlign, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) softWrap = true, [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html) overflow = TextOverflow.clip, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? maxLines, [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html) textWidthBasis = TextWidthBasis.parent, [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)? textHeightBehavior, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onEnd})

Creates a widget that animates the default text style implicitly.

const

### Properties

[child](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The widget below this widget in the tree.

final

[curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The curve to apply when animating the parameters of this container.

finalinherited

[duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration over which to animate the parameters of this container.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[maxLines](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/maxLines.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)?

An optional maximum number of lines for the text to span, wrapping if necessary.

final

[onEnd](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/onEnd.html) → [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)?

Called every time an animation completes.

finalinherited

[overflow](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/overflow.html) → [TextOverflow](https://api.flutter.dev/flutter/painting/TextOverflow.html)

How visual overflow should be handled.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[softWrap](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/softWrap.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the text should break at soft line breaks.

final

[style](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/style.html) → [TextStyle](https://api.flutter.dev/flutter/painting/TextStyle-class.html)

The target text style.

final

[textAlign](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/textAlign.html) → [TextAlign](https://api.flutter.dev/flutter/dart-ui/TextAlign.html)?

How the text should be aligned horizontally.

final

[textHeightBehavior](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/textHeightBehavior.html) → [TextHeightBehavior](https://api.flutter.dev/flutter/dart-ui/TextHeightBehavior-class.html)?

Defines how to apply [TextStyle.height](https://api.flutter.dev/flutter/painting/TextStyle/height.html) over and under text.

final

[textWidthBasis](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle/textWidthBasis.html) → [TextWidthBasis](https://api.flutter.dev/flutter/painting/TextWidthBasis.html)

The strategy to use when calculating the width of the Text.

final

## animated list

A scrolling container that animates items when they are inserted or removed.

This widget's [AnimatedListState](https://api.flutter.dev/flutter/widgets/AnimatedListState-class.html) can be used to dynamically insert or remove items. To refer to the [AnimatedListState](https://api.flutter.dev/flutter/widgets/AnimatedListState-class.html) either provide a [GlobalKey](https://api.flutter.dev/flutter/widgets/GlobalKey-class.html) or use the static [of](https://api.flutter.dev/flutter/widgets/AnimatedList/of.html) method from an item's input callback.

This widget is similar to one created by [ListView.builder](https://api.flutter.dev/flutter/widgets/ListView/ListView.builder.html).

### Constructors

[AnimatedList](https://api.flutter.dev/flutter/widgets/AnimatedList/AnimatedList.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [AnimatedItemBuilder](https://api.flutter.dev/flutter/widgets/AnimatedItemBuilder.html) itemBuilder, [int](https://api.flutter.dev/flutter/dart-core/int-class.html) initialItemCount = 0, [Axis](https://api.flutter.dev/flutter/painting/Axis.html) scrollDirection = Axis.vertical, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) reverse = false, [ScrollController](https://api.flutter.dev/flutter/widgets/ScrollController-class.html)? controller, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? primary, [ScrollPhysics](https://api.flutter.dev/flutter/widgets/ScrollPhysics-class.html)? physics, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) shrinkWrap = false, [EdgeInsetsGeometry](https://api.flutter.dev/flutter/painting/EdgeInsetsGeometry-class.html)? padding, [Clip](https://api.flutter.dev/flutter/dart-ui/Clip.html) clipBehavior = Clip.hardEdge})

Creates a scrolling container that animates items when they are inserted or removed.

const

### Properties

[clipBehavior](https://api.flutter.dev/flutter/widgets/AnimatedList/clipBehavior.html) → [Clip](https://api.flutter.dev/flutter/dart-ui/Clip.html)

The content will be clipped (or not) according to this option.

finalinherited

[controller](https://api.flutter.dev/flutter/widgets/AnimatedList/controller.html) → [ScrollController](https://api.flutter.dev/flutter/widgets/ScrollController-class.html)?

An object that can be used to control the position to which this scroll view is scrolled.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[initialItemCount](https://api.flutter.dev/flutter/widgets/AnimatedList/initialItemCount.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The number of items the [AnimatedList](https://api.flutter.dev/flutter/widgets/AnimatedList-class.html) or [AnimatedGrid](https://api.flutter.dev/flutter/widgets/AnimatedGrid-class.html) will start with.

finalinherited

[itemBuilder](https://api.flutter.dev/flutter/widgets/AnimatedList/itemBuilder.html) → [AnimatedItemBuilder](https://api.flutter.dev/flutter/widgets/AnimatedItemBuilder.html)

Called, as needed, to build children widgets.

finalinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[padding](https://api.flutter.dev/flutter/widgets/AnimatedList/padding.html) → [EdgeInsetsGeometry](https://api.flutter.dev/flutter/painting/EdgeInsetsGeometry-class.html)?

The amount of space by which to inset the children.

finalinherited

[physics](https://api.flutter.dev/flutter/widgets/AnimatedList/physics.html) → [ScrollPhysics](https://api.flutter.dev/flutter/widgets/ScrollPhysics-class.html)?

How the scroll view should respond to user input.

finalinherited

[primary](https://api.flutter.dev/flutter/widgets/AnimatedList/primary.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)?

Whether this is the primary scroll view associated with the parent [PrimaryScrollController](https://api.flutter.dev/flutter/widgets/PrimaryScrollController-class.html).

finalinherited

[reverse](https://api.flutter.dev/flutter/widgets/AnimatedList/reverse.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the scroll view scrolls in the reading direction.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[scrollDirection](https://api.flutter.dev/flutter/widgets/AnimatedList/scrollDirection.html) → [Axis](https://api.flutter.dev/flutter/painting/Axis.html)

The [Axis](https://api.flutter.dev/flutter/painting/Axis.html) along which the scroll view's offset increases.

finalinherited

[shrinkWrap](https://api.flutter.dev/flutter/widgets/AnimatedList/shrinkWrap.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the extent of the scroll view in the [scrollDirection](https://api.flutter.dev/flutter/widgets/AnimatedList/scrollDirection.html) should be determined by the contents being viewed.

finalinherited

## animated list state

The [AnimatedListState](https://api.flutter.dev/flutter/widgets/AnimatedListState-class.html) for [AnimatedList](https://api.flutter.dev/flutter/widgets/AnimatedList-class.html), a scrolling list container that animates items when they are inserted or removed.

When an item is inserted with [insertItem](https://api.flutter.dev/flutter/widgets/AnimatedListState/insertItem.html) an animation begins running. The animation is passed to [AnimatedList.itemBuilder](https://api.flutter.dev/flutter/widgets/AnimatedList/itemBuilder.html) whenever the item's widget is needed.

When multiple items are inserted with [insertAllItems](https://api.flutter.dev/flutter/widgets/AnimatedListState/insertAllItems.html) an animation begins running. The animation is passed to [AnimatedList.itemBuilder](https://api.flutter.dev/flutter/widgets/AnimatedList/itemBuilder.html) whenever the item's widget is needed.

When an item is removed with [removeItem](https://api.flutter.dev/flutter/widgets/AnimatedListState/removeItem.html) its animation is reversed. The removed item's animation is passed to the [removeItem](https://api.flutter.dev/flutter/widgets/AnimatedListState/removeItem.html) builder parameter.

An app that needs to insert or remove items in response to an event can refer to the [AnimatedList](https://api.flutter.dev/flutter/widgets/AnimatedList-class.html)'s state with a global key:

[AnimatedList](https://api.flutter.dev/flutter/widgets/AnimatedList-class.html) item input handlers can also refer to their [AnimatedListState](https://api.flutter.dev/flutter/widgets/AnimatedListState-class.html) with the static [AnimatedList.of](https://api.flutter.dev/flutter/widgets/AnimatedList/of.html) method.

Inheritance

* [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)
* [State](https://api.flutter.dev/flutter/widgets/State-class.html)<[AnimatedList](https://api.flutter.dev/flutter/widgets/AnimatedList-class.html)>
* AnimatedListState

### Constructor

[AnimatedListState](https://api.flutter.dev/flutter/widgets/AnimatedListState/AnimatedListState.html)()

### Properties

[context](https://api.flutter.dev/flutter/widgets/State/context.html) → [BuildContext](https://api.flutter.dev/flutter/widgets/BuildContext-class.html)

The location in the tree where this widget builds.

read-onlyinherited

[hashCode](https://api.flutter.dev/flutter/dart-core/Object/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[mounted](https://api.flutter.dev/flutter/widgets/State/mounted.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether this [State](https://api.flutter.dev/flutter/widgets/State-class.html) object is currently in a tree.

read-onlyinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[widget](https://api.flutter.dev/flutter/widgets/State/widget.html) → [AnimatedList](https://api.flutter.dev/flutter/widgets/AnimatedList-class.html)

The current configuration.

read-onlyinherited

## animated modal barrier

A widget that prevents the user from interacting with widgets behind itself, and can be configured with an animated color value.

The modal barrier is the scrim that is rendered behind each route, which generally prevents the user from interacting with the route below the current route, and normally partially obscures such routes.

For example, when a dialog is on the screen, the page below the dialog is usually darkened by the modal barrier.

This widget is similar to [ModalBarrier](https://api.flutter.dev/flutter/widgets/ModalBarrier-class.html) except that it takes an animated [color](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/color.html) instead of a single color.

See also:

* [ModalRoute](https://api.flutter.dev/flutter/widgets/ModalRoute-class.html), which uses this widget.

Inheritance

* [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)
* [DiagnosticableTree](https://api.flutter.dev/flutter/foundation/DiagnosticableTree-class.html)
* [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)
* [StatefulWidget](https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html)
* [AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html)
* AnimatedModalBarrier

### Constructors

[AnimatedModalBarrier](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/AnimatedModalBarrier.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)?> color, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) dismissible = true, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticsLabel, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)? barrierSemanticsDismissible, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onDismiss, [ValueNotifier](https://api.flutter.dev/flutter/foundation/ValueNotifier-class.html)<[EdgeInsets](https://api.flutter.dev/flutter/painting/EdgeInsets-class.html)>? clipDetailsNotifier, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticsOnTapHint})

Creates a widget that blocks user interaction.

const

### Properties

[barrierSemanticsDismissible](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/barrierSemanticsDismissible.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)?

Whether the modal barrier semantics are included in the semantics tree.

final

[clipDetailsNotifier](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/clipDetailsNotifier.html) → [ValueNotifier](https://api.flutter.dev/flutter/foundation/ValueNotifier-class.html)<[EdgeInsets](https://api.flutter.dev/flutter/painting/EdgeInsets-class.html)>?

Contains a value of type EdgeInsets that specifies how the [SemanticsNode.rect](https://api.flutter.dev/flutter/semantics/SemanticsNode/rect.html) of the widget should be clipped.

final

[color](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/color.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)?>

If non-null, fill the barrier with this color.

read-only

[dismissible](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/dismissible.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether touching the barrier will pop the current route off the [Navigator](https://api.flutter.dev/flutter/widgets/Navigator-class.html).

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedWidget/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) to which this widget is listening.

finalinherited

[onDismiss](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/onDismiss.html) → [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)?

Called when the barrier is being dismissed.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[semanticsLabel](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/semanticsLabel.html) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)?

Semantics label used for the barrier if it is [dismissible](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/dismissible.html).

final

[semanticsOnTapHint](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier/semanticsOnTapHint.html) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)?

This hint text instructs users what they are able to do when they tap on the [ModalBarrier](https://api.flutter.dev/flutter/widgets/ModalBarrier-class.html)

final

## animated opacity

Animated version of [Opacity](https://api.flutter.dev/flutter/widgets/Opacity-class.html) which automatically transitions the child's opacity over a given duration whenever the given opacity changes.

Animating an opacity is relatively expensive because it requires painting the child into an intermediate buffer.

Here's an illustration of what using this widget looks like, using a [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) of [Curves.fastOutSlowIn](https://api.flutter.dev/flutter/animation/Curves/fastOutSlowIn-constant.html).

Setting the [opacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity/opacity.html) to zero does not prevent hit testing from being applied to the descendants of the [AnimatedOpacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity-class.html) widget. This can be confusing for the user, who may not see anything, and may believe the area of the interface where the [AnimatedOpacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity-class.html) is hiding a widget to be non-interactive.

With certain widgets, such as [Flow](https://api.flutter.dev/flutter/widgets/Flow-class.html), that compute their positions only when they are painted, this can actually lead to bugs (from unexpected geometry to exceptions), because those widgets are not painted by the [AnimatedOpacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity-class.html) widget at all when the [opacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity/opacity.html) animation reaches zero.

To avoid such problems, it is generally a good idea to use an [IgnorePointer](https://api.flutter.dev/flutter/widgets/IgnorePointer-class.html) widget when setting the [opacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity/opacity.html) to zero. This prevents interactions with any children in the subtree when the [child](https://api.flutter.dev/flutter/widgets/AnimatedOpacity/child.html) is animating away.

### Constructors

[AnimatedOpacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity/AnimatedOpacity.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child, required [double](https://api.flutter.dev/flutter/dart-core/double-class.html) opacity, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onEnd, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) alwaysIncludeSemantics = false})

Creates a widget that animates its opacity implicitly.

const

### Properties

[alwaysIncludeSemantics](https://api.flutter.dev/flutter/widgets/AnimatedOpacity/alwaysIncludeSemantics.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the semantic information of the children is always included.

final

[child](https://api.flutter.dev/flutter/widgets/AnimatedOpacity/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

final

[curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The curve to apply when animating the parameters of this container.

finalinherited

[duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration over which to animate the parameters of this container.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[onEnd](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/onEnd.html) → [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)?

Called every time an animation completes.

finalinherited

[opacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity/opacity.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)

The target opacity.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## animated physical model

Animated version of [PhysicalModel](https://api.flutter.dev/flutter/widgets/PhysicalModel-class.html).

The [borderRadius](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/borderRadius.html) and [elevation](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/elevation.html) are animated.

The [color](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/color.html) is animated if the [animateColor](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/animateColor.html) property is set; otherwise, the color changes immediately at the start of the animation for the other two properties. This allows the color to be animated independently (e.g. because it is being driven by an [AnimatedTheme](https://api.flutter.dev/flutter/material/AnimatedTheme-class.html)).

The [shape](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/shape.html) is not animated.

Here's an illustration of what using this widget looks like, using a [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) of [Curves.fastOutSlowIn](https://api.flutter.dev/flutter/animation/Curves/fastOutSlowIn-constant.html).

Inheritance

* [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)
* [DiagnosticableTree](https://api.flutter.dev/flutter/foundation/DiagnosticableTree-class.html)
* [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)
* [StatefulWidget](https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html)
* [ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget-class.html)
* AnimatedPhysicalModel

### Constructors

[AnimatedPhysicalModel](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/AnimatedPhysicalModel.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) child, required [BoxShape](https://api.flutter.dev/flutter/painting/BoxShape.html) shape, [Clip](https://api.flutter.dev/flutter/dart-ui/Clip.html) clipBehavior = Clip.none, [BorderRadius](https://api.flutter.dev/flutter/painting/BorderRadius-class.html) borderRadius = BorderRadius.zero, required [double](https://api.flutter.dev/flutter/dart-core/double-class.html) elevation, required [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html) color, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) animateColor = true, required [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html) shadowColor, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) animateShadowColor = true, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onEnd})

Creates a widget that animates the properties of a [PhysicalModel](https://api.flutter.dev/flutter/widgets/PhysicalModel-class.html).

const

### Properties

[animateColor](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/animateColor.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the color should be animated.

final

[animateShadowColor](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/animateShadowColor.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the shadow color should be animated.

final

[borderRadius](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/borderRadius.html) → [BorderRadius](https://api.flutter.dev/flutter/painting/BorderRadius-class.html)

The target border radius of the rounded corners for a rectangle shape.

final

[child](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The widget below this widget in the tree.

final

[clipBehavior](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/clipBehavior.html) → [Clip](https://api.flutter.dev/flutter/dart-ui/Clip.html)

The content will be clipped (or not) according to this option.

final

[color](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/color.html) → [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)

The target background color.

final

[curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The curve to apply when animating the parameters of this container.

finalinherited

[duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration over which to animate the parameters of this container.

finalinherited

[elevation](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/elevation.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)

The target z-coordinate relative to the parent at which to place this physical object.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[onEnd](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/onEnd.html) → [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)?

Called every time an animation completes.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[shadowColor](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/shadowColor.html) → [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)

The target shadow color.

final

[shape](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel/shape.html) → [BoxShape](https://api.flutter.dev/flutter/painting/BoxShape.html)

The type of shape.

final

## animated positioned

Animated version of [Positioned](https://api.flutter.dev/flutter/widgets/Positioned-class.html) which automatically transitions the child's position over a given duration whenever the given position changes.

Only works if it's the child of a [Stack](https://api.flutter.dev/flutter/widgets/Stack-class.html).

This widget is a good choice if the *size* of the child would end up changing as a result of this animation. If the size is intended to remain the same, with only the *position* changing over time, then consider [SlideTransition](https://api.flutter.dev/flutter/widgets/SlideTransition-class.html) instead. [SlideTransition](https://api.flutter.dev/flutter/widgets/SlideTransition-class.html) only triggers a repaint each frame of the animation, whereas [AnimatedPositioned](https://api.flutter.dev/flutter/widgets/AnimatedPositioned-class.html) will trigger a relayout as well.

Here's an illustration of what using this widget looks like, using a [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) of [Curves.fastOutSlowIn](https://api.flutter.dev/flutter/animation/Curves/fastOutSlowIn-constant.html).

For the animation, you can choose a [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) as well as a [duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) and the widget will automatically animate to the new target position. If you require more control over the animation (e.g. if you want to stop it mid-animation), consider using a [PositionedTransition](https://api.flutter.dev/flutter/widgets/PositionedTransition-class.html) instead, which takes a provided [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) as an argument. While that allows you to fine-tune the animation, it also requires more development overhead as you have to manually m

### Constructors

[AnimatedPositioned](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/AnimatedPositioned.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) child, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? left, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? top, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? right, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? bottom, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? width, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? height, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onEnd})

Creates a widget that animates its position implicitly.

const

[AnimatedPositioned.fromRect](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/AnimatedPositioned.fromRect.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) child, required [Rect](https://api.flutter.dev/flutter/dart-ui/Rect-class.html) rect, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onEnd})

Creates a widget that animates the rectangle it occupies implicitly.

### Properties

[bottom](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/bottom.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The offset of the child's bottom edge from the bottom of the stack.

final

[child](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The widget below this widget in the tree.

final

[curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The curve to apply when animating the parameters of this container.

finalinherited

[duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration over which to animate the parameters of this container.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[height](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/height.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The child's height.

final

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[left](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/left.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The offset of the child's left edge from the left of the stack.

final

[onEnd](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/onEnd.html) → [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)?

Called every time an animation completes.

finalinherited

[right](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/right.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The offset of the child's right edge from the right of the stack.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[top](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/top.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The offset of the child's top edge from the top of the stack.

final

[width](https://api.flutter.dev/flutter/widgets/AnimatedPositioned/width.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The child's width.

final

## animated size

Animated widget that automatically transitions its size over a given duration whenever the given child's size changes.

### Constructors

[AnimatedSize](https://api.flutter.dev/flutter/widgets/AnimatedSize/AnimatedSize.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html) alignment = Alignment.center, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)? reverseDuration, [Clip](https://api.flutter.dev/flutter/dart-ui/Clip.html) clipBehavior = Clip.hardEdge})

Creates a widget that animates its size to match that of its child.

const

### Properties

[alignment](https://api.flutter.dev/flutter/widgets/AnimatedSize/alignment.html) → [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html)

The alignment of the child within the parent when the parent is not yet the same size as the child.

final

[child](https://api.flutter.dev/flutter/widgets/AnimatedSize/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

final

[clipBehavior](https://api.flutter.dev/flutter/widgets/AnimatedSize/clipBehavior.html) → [Clip](https://api.flutter.dev/flutter/dart-ui/Clip.html)

The content will be clipped (or not) according to this option.

final

[curve](https://api.flutter.dev/flutter/widgets/AnimatedSize/curve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The animation curve when transitioning this widget's size to match the child's size.

final

[duration](https://api.flutter.dev/flutter/widgets/AnimatedSize/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration when transitioning this widget's size to match the child's size.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[reverseDuration](https://api.flutter.dev/flutter/widgets/AnimatedSize/reverseDuration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)?

The duration when transitioning this widget's size to match the child's size when going in reverse.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## animated widget

A widget that rebuilds when the given [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) changes value.

[AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html) is most commonly used with [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) objects, which are [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html), but it can be used with any [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html), including [ChangeNotifier](https://api.flutter.dev/flutter/foundation/ChangeNotifier-class.html) and [ValueNotifier](https://api.flutter.dev/flutter/foundation/ValueNotifier-class.html).

[AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html) is most useful for widgets that are otherwise stateless. To use [AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html), subclass it and implement the build function.

### Relationship to [ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget-class.html)s

[AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html)s (and their subclasses) take an explicit [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) as argument, which is usually an [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) derived from an [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html). In most cases, the lifecycle of that [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html) has to be managed manually by the developer. In contrast to that, [ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget-class.html)s (and their subclasses) automatically manage their own internal [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html) making those classes easier to use as no external [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) has to be provided by the developer. If you only need to set a target value for the animation and configure its duration/curve, consider using (a subclass of) [ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget-class.html)s instead of (a subclass of) this class.

### Common animated widgets

A number of animated widgets ship with the framework. They are usually named FooTransition, where Foo is the name of the non-animated version of that widget. The subclasses of this class should not be confused with subclasses of [ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget-class.html) (see above), which are usually named AnimatedFoo. Commonly used animated widgets include:

* [ListenableBuilder](https://api.flutter.dev/flutter/widgets/ListenableBuilder-class.html), which uses a builder pattern that is useful for complex [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) use cases.
* [AnimatedBuilder](https://api.flutter.dev/flutter/widgets/AnimatedBuilder-class.html), which uses a builder pattern that is useful for complex [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) use cases.
* [AlignTransition](https://api.flutter.dev/flutter/widgets/AlignTransition-class.html), which is an animated version of [Align](https://api.flutter.dev/flutter/widgets/Align-class.html).
* [DecoratedBoxTransition](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition-class.html), which is an animated version of [DecoratedBox](https://api.flutter.dev/flutter/widgets/DecoratedBox-class.html).
* [DefaultTextStyleTransition](https://api.flutter.dev/flutter/widgets/DefaultTextStyleTransition-class.html), which is an animated version of [DefaultTextStyle](https://api.flutter.dev/flutter/widgets/DefaultTextStyle-class.html).
* [PositionedTransition](https://api.flutter.dev/flutter/widgets/PositionedTransition-class.html), which is an animated version of [Positioned](https://api.flutter.dev/flutter/widgets/Positioned-class.html).
* [RelativePositionedTransition](https://api.flutter.dev/flutter/widgets/RelativePositionedTransition-class.html), which is an animated version of [Positioned](https://api.flutter.dev/flutter/widgets/Positioned-class.html).
* [RotationTransition](https://api.flutter.dev/flutter/widgets/RotationTransition-class.html), which animates the rotation of a widget.
* [ScaleTransition](https://api.flutter.dev/flutter/widgets/ScaleTransition-class.html), which animates the scale of a widget.
* [SizeTransition](https://api.flutter.dev/flutter/widgets/SizeTransition-class.html), which animates its own size.
* [SlideTransition](https://api.flutter.dev/flutter/widgets/SlideTransition-class.html), which animates the position of a widget relative to its normal position.
* [FadeTransition](https://api.flutter.dev/flutter/widgets/FadeTransition-class.html), which is an animated version of [Opacity](https://api.flutter.dev/flutter/widgets/Opacity-class.html).
* [AnimatedModalBarrier](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier-class.html), which is an animated version of [ModalBarrier](https://api.flutter.dev/flutter/widgets/ModalBarrier-class.html).

Inheritance

* [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)
* [DiagnosticableTree](https://api.flutter.dev/flutter/foundation/DiagnosticableTree-class.html)
* [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)
* [StatefulWidget](https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html)
* AnimatedWidget

Implementers

* [AlignTransition](https://api.flutter.dev/flutter/widgets/AlignTransition-class.html)
* [AnimatedModalBarrier](https://api.flutter.dev/flutter/widgets/AnimatedModalBarrier-class.html)
* [DecoratedBoxTransition](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition-class.html)
* [DefaultTextStyleTransition](https://api.flutter.dev/flutter/widgets/DefaultTextStyleTransition-class.html)
* [ListenableBuilder](https://api.flutter.dev/flutter/widgets/ListenableBuilder-class.html)
* [MatrixTransition](https://api.flutter.dev/flutter/widgets/MatrixTransition-class.html)
* [PositionedTransition](https://api.flutter.dev/flutter/widgets/PositionedTransition-class.html)
* [RelativePositionedTransition](https://api.flutter.dev/flutter/widgets/RelativePositionedTransition-class.html)
* [SizeTransition](https://api.flutter.dev/flutter/widgets/SizeTransition-class.html)
* [SlideTransition](https://api.flutter.dev/flutter/widgets/SlideTransition-class.html)

### Constructors

[AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget/AnimatedWidget.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) listenable})

Creates a widget that rebuilds when the given listenable changes.

const

### Properties

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedWidget/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) to which this widget is listening.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## implicitly animated widget

An abstract class for building widgets that animate changes to their properties.

Widgets of this type will not animate when they are first added to the widget tree. Rather, when they are rebuilt with different values, they will respond to those changes by animating the changes over a specified [duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html).

Which properties are animated is left up to the subclass. Subclasses' [State](https://api.flutter.dev/flutter/widgets/State-class.html)s must extend [ImplicitlyAnimatedWidgetState](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidgetState-class.html) and provide a way to visit the relevant fields to animate.

### Relationship to [AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html)s

[ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget-class.html)s (and their subclasses) automatically animate changes in their properties whenever they change. For this, they create and manage their own internal [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html)s to power the animation. While these widgets are simple to use and don't require you to manually manage the lifecycle of an [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html), they are also somewhat limited: Besides the target value for the animated property, developers can only choose a [duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) and [curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) for the animation. If you require more control over the animation (e.g. you want to stop it somewhere in the middle), consider using an [AnimatedWidget](https://api.flutter.dev/flutter/widgets/AnimatedWidget-class.html) or one of its subclasses. These widgets take an [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) as an argument to power the animation. This gives the developer full control over the animation at the cost of requiring you to manually manage the underlying [AnimationController](https://api.flutter.dev/flutter/animation/AnimationController-class.html).

### Common implicitly animated widgets

A number of implicitly animated widgets ship with the framework. They are usually named AnimatedFoo, where Foo is the name of the non-animated version of that widget. Commonly used implicitly animated widgets include:

* [TweenAnimationBuilder](https://api.flutter.dev/flutter/widgets/TweenAnimationBuilder-class.html), which animates any property expressed by a [Tween](https://api.flutter.dev/flutter/animation/Tween-class.html) to a specified target value.
* [AnimatedAlign](https://api.flutter.dev/flutter/widgets/AnimatedAlign-class.html), which is an implicitly animated version of [Align](https://api.flutter.dev/flutter/widgets/Align-class.html).
* [AnimatedContainer](https://api.flutter.dev/flutter/widgets/AnimatedContainer-class.html), which is an implicitly animated version of [Container](https://api.flutter.dev/flutter/widgets/Container-class.html).
* [AnimatedDefaultTextStyle](https://api.flutter.dev/flutter/widgets/AnimatedDefaultTextStyle-class.html), which is an implicitly animated version of [DefaultTextStyle](https://api.flutter.dev/flutter/widgets/DefaultTextStyle-class.html).
* [AnimatedScale](https://api.flutter.dev/flutter/widgets/AnimatedScale-class.html), which is an implicitly animated version of [Transform.scale](https://api.flutter.dev/flutter/widgets/Transform/Transform.scale.html).
* [AnimatedRotation](https://api.flutter.dev/flutter/widgets/AnimatedRotation-class.html), which is an implicitly animated version of [Transform.rotate](https://api.flutter.dev/flutter/widgets/Transform/Transform.rotate.html).
* [AnimatedSlide](https://api.flutter.dev/flutter/widgets/AnimatedSlide-class.html), which implicitly animates the position of a widget relative to its normal position.
* [AnimatedOpacity](https://api.flutter.dev/flutter/widgets/AnimatedOpacity-class.html), which is an implicitly animated version of [Opacity](https://api.flutter.dev/flutter/widgets/Opacity-class.html).
* [AnimatedPadding](https://api.flutter.dev/flutter/widgets/AnimatedPadding-class.html), which is an implicitly animated version of [Padding](https://api.flutter.dev/flutter/widgets/Padding-class.html).
* [AnimatedPhysicalModel](https://api.flutter.dev/flutter/widgets/AnimatedPhysicalModel-class.html), which is an implicitly animated version of [PhysicalModel](https://api.flutter.dev/flutter/widgets/PhysicalModel-class.html).
* [AnimatedPositioned](https://api.flutter.dev/flutter/widgets/AnimatedPositioned-class.html), which is an implicitly animated version of [Positioned](https://api.flutter.dev/flutter/widgets/Positioned-class.html).
* [AnimatedPositionedDirectional](https://api.flutter.dev/flutter/widgets/AnimatedPositionedDirectional-class.html), which is an implicitly animated version of [PositionedDirectional](https://api.flutter.dev/flutter/widgets/PositionedDirectional-class.html).
* [AnimatedTheme](https://api.flutter.dev/flutter/material/AnimatedTheme-class.html), which is an implicitly animated version of [Theme](https://api.flutter.dev/flutter/material/Theme-class.html).
* [AnimatedCrossFade](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade-class.html), which cross-fades between two given children and animates itself between their sizes.
* [AnimatedSize](https://api.flutter.dev/flutter/widgets/AnimatedSize-class.html), which automatically transitions its size over a given duration.
* [AnimatedSwitcher](https://api.flutter.dev/flutter/widgets/AnimatedSwitcher-class.html), which fades from one widget to another.

### Constructors

[ImplicitlyAnimatedWidget](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/ImplicitlyAnimatedWidget.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html) curve = Curves.linear, required [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html) duration, [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)? onEnd})

Initializes fields for subclasses.

const

### Properties

[curve](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/curve.html) → [Curve](https://api.flutter.dev/flutter/animation/Curve-class.html)

The curve to apply when animating the parameters of this container.

final

[duration](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/duration.html) → [Duration](https://api.flutter.dev/flutter/dart-core/Duration-class.html)

The duration over which to animate the parameters of this container.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[onEnd](https://api.flutter.dev/flutter/widgets/ImplicitlyAnimatedWidget/onEnd.html) → [VoidCallback](https://api.flutter.dev/flutter/dart-ui/VoidCallback.html)?

Called every time an animation completes.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## decorated box transition

Animated version of a [DecoratedBox](https://api.flutter.dev/flutter/widgets/DecoratedBox-class.html) that animates the different properties of its [Decoration](https://api.flutter.dev/flutter/painting/Decoration-class.html).

Here's an illustration of the [DecoratedBoxTransition](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition-class.html) widget, with it's [decoration](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition/decoration.html) animated by a [CurvedAnimation](https://api.flutter.dev/flutter/animation/CurvedAnimation-class.html) set to [Curves.decelerate](https://api.flutter.dev/flutter/animation/Curves/decelerate-constant.html):

### Constructors

[DecoratedBoxTransition](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition/DecoratedBoxTransition.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[Decoration](https://api.flutter.dev/flutter/painting/Decoration-class.html)> decoration, [DecorationPosition](https://api.flutter.dev/flutter/rendering/DecorationPosition.html) position = DecorationPosition.background, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) child})

Creates an animated [DecoratedBox](https://api.flutter.dev/flutter/widgets/DecoratedBox-class.html) whose [Decoration](https://api.flutter.dev/flutter/painting/Decoration-class.html) animation updates the widget.

const

### Properties

[child](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The widget below this widget in the tree.

final

[decoration](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition/decoration.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[Decoration](https://api.flutter.dev/flutter/painting/Decoration-class.html)>

Animation of the decoration to paint.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedWidget/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) to which this widget is listening.

finalinherited

[position](https://api.flutter.dev/flutter/widgets/DecoratedBoxTransition/position.html) → [DecorationPosition](https://api.flutter.dev/flutter/rendering/DecorationPosition.html)

Whether to paint the box decoration behind or in front of the child.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## fade transition

Animates the opacity of a widget.

For a widget that automatically animates between the sizes of two children, fading between them, see [AnimatedCrossFade](https://api.flutter.dev/flutter/widgets/AnimatedCrossFade-class.html).

### Hit testing

Setting the [opacity](https://api.flutter.dev/flutter/widgets/FadeTransition/opacity.html) to zero does not prevent hit testing from being applied to the descendants of the [FadeTransition](https://api.flutter.dev/flutter/widgets/FadeTransition-class.html) widget. This can be confusing for the user, who may not see anything, and may believe the area of the interface where the [FadeTransition](https://api.flutter.dev/flutter/widgets/FadeTransition-class.html) is hiding a widget to be non-interactive.

With certain widgets, such as [Flow](https://api.flutter.dev/flutter/widgets/Flow-class.html), that compute their positions only when they are painted, this can actually lead to bugs (from unexpected geometry to exceptions), because those widgets are not painted by the [FadeTransition](https://api.flutter.dev/flutter/widgets/FadeTransition-class.html) widget at all when the [opacity](https://api.flutter.dev/flutter/widgets/FadeTransition/opacity.html) animation reaches zero.

To avoid such problems, it is generally a good idea to combine this widget with an [IgnorePointer](https://api.flutter.dev/flutter/widgets/IgnorePointer-class.html) that one enables when the [opacity](https://api.flutter.dev/flutter/widgets/FadeTransition/opacity.html) animation reaches zero. This prevents interactions with any children in the subtree when the [child](https://api.flutter.dev/flutter/widgets/SingleChildRenderObjectWidget/child.html) is not visible. For performance reasons, when implementing this, care should be taken not to rebuild the relevant widget (e.g. by calling [State.setState](https://api.flutter.dev/flutter/widgets/State/setState.html)) except at the transition point.

### Constructors

[FadeTransition](https://api.flutter.dev/flutter/widgets/FadeTransition/FadeTransition.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)> opacity, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) alwaysIncludeSemantics = false, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child})

Creates an opacity transition.

const

### Properties

[alwaysIncludeSemantics](https://api.flutter.dev/flutter/widgets/FadeTransition/alwaysIncludeSemantics.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether the semantic information of the children is always included.

final

[child](https://api.flutter.dev/flutter/widgets/SingleChildRenderObjectWidget/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[opacity](https://api.flutter.dev/flutter/widgets/FadeTransition/opacity.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>

The animation that controls the opacity of the child.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## hero

A widget that marks its child as being a candidate for [hero animations](https://flutter.dev/docs/development/ui/animations/hero-animations).

When a [PageRoute](https://api.flutter.dev/flutter/widgets/PageRoute-class.html) is pushed or popped with the [Navigator](https://api.flutter.dev/flutter/widgets/Navigator-class.html), the entire screen's content is replaced. An old route disappears and a new route appears. If there's a common visual feature on both routes then it can be helpful for orienting the user for the feature to physically move from one page to the other during the routes' transition. Such an animation is called a *hero animation*. The hero widgets "fly" in the Navigator's overlay during the transition and while they're in-flight they're, by default, not shown in their original locations in the old and new routes.

To label a widget as such a feature, wrap it in a [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html) widget. When navigation happens, the [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html) widgets on each route are identified by the [HeroController](https://api.flutter.dev/flutter/widgets/HeroController-class.html). For each pair of [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html) widgets that have the same tag, a hero animation is triggered.

If a [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html) is already in flight when navigation occurs, its flight animation will be redirected to its new destination. The widget shown in-flight during the transition is, by default, the destination route's [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html)'s child.

For a Hero animation to trigger, the Hero has to exist on the very first frame of the new page's animation.

### Discussion

Heroes and the [Navigator](https://api.flutter.dev/flutter/widgets/Navigator-class.html)'s [Overlay](https://api.flutter.dev/flutter/widgets/Overlay-class.html) [Stack](https://api.flutter.dev/flutter/widgets/Stack-class.html) must be axis-aligned for all this to work. The top left and bottom right coordinates of each animated Hero will be converted to global coordinates and then from there converted to that [Stack](https://api.flutter.dev/flutter/widgets/Stack-class.html)'s coordinate space, and the entire Hero subtree will, for the duration of the animation, be lifted out of its original place, and positioned on that stack. If the [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html) isn't axis aligned, this is going to fail in a rather ugly fashion. Don't rotate your heroes!

To make the animations look good, it's critical that the widget tree for the hero in both locations be essentially identical. The widget of the target is, by default, used to do the transition: when going from route A to route B, route B's hero's widget is placed over route A's hero's widget. Additionally, if the [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html) subtree changes appearance based on an [InheritedWidget](https://api.flutter.dev/flutter/widgets/InheritedWidget-class.html) (such as [MediaQuery](https://api.flutter.dev/flutter/widgets/MediaQuery-class.html) or [Theme](https://api.flutter.dev/flutter/material/Theme-class.html)), then the hero animation may have discontinuity at the start or the end of the animation because route A and route B provides different such [InheritedWidget](https://api.flutter.dev/flutter/widgets/InheritedWidget-class.html)s. Consider providing a custom [flightShuttleBuilder](https://api.flutter.dev/flutter/widgets/Hero/flightShuttleBuilder.html) to ensure smooth transitions. The default [flightShuttleBuilder](https://api.flutter.dev/flutter/widgets/Hero/flightShuttleBuilder.html) interpolates [MediaQuery](https://api.flutter.dev/flutter/widgets/MediaQuery-class.html)'s paddings. If your [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html) widget uses custom [InheritedWidget](https://api.flutter.dev/flutter/widgets/InheritedWidget-class.html)s and displays a discontinuity in the animation, try to provide custom in-flight transition using [flightShuttleBuilder](https://api.flutter.dev/flutter/widgets/Hero/flightShuttleBuilder.html).

By default, both route A and route B's heroes are hidden while the transitioning widget is animating in-flight above the 2 routes. [placeholderBuilder](https://api.flutter.dev/flutter/widgets/Hero/placeholderBuilder.html) can be used to show a custom widget in their place instead once the transition has taken flight.

During the transition, the transition widget is animated to route B's hero's position, and then the widget is inserted into route B. When going back from B to A, route A's hero's widget is, by default, placed over where route B's hero's widget was, and then the animation goes the other way.

### Nested Navigators

If either or both routes contain nested [Navigator](https://api.flutter.dev/flutter/widgets/Navigator-class.html)s, only [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html)es contained in the top-most routes (as defined by [Route.isCurrent](https://api.flutter.dev/flutter/widgets/Route/isCurrent.html)) of those nested [*Navigator*](https://api.flutter.dev/flutter/widgets/Navigator-class.html)s are considered for animation. Just like in the non-nested case the top-most routes containing these [Hero](https://api.flutter.dev/flutter/widgets/Hero-class.html)es in the nested [Navigator](https://api.flutter.dev/flutter/widgets/Navigator-class.html)s have to be [PageRoute](https://api.flutter.dev/flutter/widgets/PageRoute-class.html)s.

### Constructors

[Hero](https://api.flutter.dev/flutter/widgets/Hero/Hero.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html) tag, [CreateRectTween](https://api.flutter.dev/flutter/widgets/CreateRectTween.html)? createRectTween, [HeroFlightShuttleBuilder](https://api.flutter.dev/flutter/widgets/HeroFlightShuttleBuilder.html)? flightShuttleBuilder, [HeroPlaceholderBuilder](https://api.flutter.dev/flutter/widgets/HeroPlaceholderBuilder.html)? placeholderBuilder, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) transitionOnUserGestures = false, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) child})

Create a hero.

const

### Properties

[child](https://api.flutter.dev/flutter/widgets/Hero/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The widget subtree that will "fly" from one route to another during a [Navigator](https://api.flutter.dev/flutter/widgets/Navigator-class.html) push or pop transition.

final

[createRectTween](https://api.flutter.dev/flutter/widgets/Hero/createRectTween.html) → [CreateRectTween](https://api.flutter.dev/flutter/widgets/CreateRectTween.html)?

Defines how the destination hero's bounds change as it flies from the starting route to the destination route.

final

[flightShuttleBuilder](https://api.flutter.dev/flutter/widgets/Hero/flightShuttleBuilder.html) → [HeroFlightShuttleBuilder](https://api.flutter.dev/flutter/widgets/HeroFlightShuttleBuilder.html)?

Optional override to supply a widget that's shown during the hero's flight.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[placeholderBuilder](https://api.flutter.dev/flutter/widgets/Hero/placeholderBuilder.html) → [HeroPlaceholderBuilder](https://api.flutter.dev/flutter/widgets/HeroPlaceholderBuilder.html)?

Placeholder widget left in place as the Hero's [child](https://api.flutter.dev/flutter/widgets/Hero/child.html) once the flight takes off.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[tag](https://api.flutter.dev/flutter/widgets/Hero/tag.html) → [Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)

The identifier for this particular hero. If the tag of this hero matches the tag of a hero on a [PageRoute](https://api.flutter.dev/flutter/widgets/PageRoute-class.html) that we're navigating to or from, then a hero animation will be triggered.

final

[transitionOnUserGestures](https://api.flutter.dev/flutter/widgets/Hero/transitionOnUserGestures.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether to perform the hero transition if the [PageRoute](https://api.flutter.dev/flutter/widgets/PageRoute-class.html) transition was triggered by a user gesture, such as a back swipe on iOS.

## positioned transition

Animated version of [Positioned](https://api.flutter.dev/flutter/widgets/Positioned-class.html) which takes a specific [Animation<RelativeRect>](https://api.flutter.dev/flutter/animation/Animation-class.html) to transition the child's position from a start position to an end position over the lifetime of the animation.

Only works if it's the child of a [Stack](https://api.flutter.dev/flutter/widgets/Stack-class.html).

### Constructors

[PositionedTransition](https://api.flutter.dev/flutter/widgets/PositionedTransition/PositionedTransition.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[RelativeRect](https://api.flutter.dev/flutter/rendering/RelativeRect-class.html)> rect, required [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html) child})

Creates a transition for [Positioned](https://api.flutter.dev/flutter/widgets/Positioned-class.html).

const

### Properties

[child](https://api.flutter.dev/flutter/widgets/PositionedTransition/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)

The widget below this widget in the tree.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedWidget/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) to which this widget is listening.

finalinherited

[rect](https://api.flutter.dev/flutter/widgets/PositionedTransition/rect.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[RelativeRect](https://api.flutter.dev/flutter/rendering/RelativeRect-class.html)>

The animation that controls the child's size and position.

read-only

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

## rotation transition

Animates the rotation of a widget.

Here's an illustration of the [RotationTransition](https://api.flutter.dev/flutter/widgets/RotationTransition-class.html) widget, with it's [turns](https://api.flutter.dev/flutter/widgets/RotationTransition/turns.html) animated by a [CurvedAnimation](https://api.flutter.dev/flutter/animation/CurvedAnimation-class.html) set to [Curves.elasticOut](https://api.flutter.dev/flutter/animation/Curves/elasticOut-constant.html):

### Constructors

[RotationTransition](https://api.flutter.dev/flutter/widgets/RotationTransition/RotationTransition.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)> turns, [Alignment](https://api.flutter.dev/flutter/painting/Alignment-class.html) alignment = Alignment.center, [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html)? filterQuality, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child})

Creates a rotation transition.

const

### Properties

[alignment](https://api.flutter.dev/flutter/widgets/MatrixTransition/alignment.html) → [Alignment](https://api.flutter.dev/flutter/painting/Alignment-class.html)

The alignment of the origin of the coordinate system in which the transform takes place, relative to the size of the box.

finalinherited

[animation](https://api.flutter.dev/flutter/widgets/MatrixTransition/animation.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>

The animation that controls the matrix of the child.

read-onlyinherited

[child](https://api.flutter.dev/flutter/widgets/MatrixTransition/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

finalinherited

[filterQuality](https://api.flutter.dev/flutter/widgets/MatrixTransition/filterQuality.html) → [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html)?

The filter quality with which to apply the transform as a bitmap operation.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedWidget/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) to which this widget is listening.

finalinherited

[onTransform](https://api.flutter.dev/flutter/widgets/MatrixTransition/onTransform.html) → [TransformCallback](https://api.flutter.dev/flutter/widgets/TransformCallback.html)

The callback to compute a [Matrix4](https://api.flutter.dev/flutter/vector_math_64/Matrix4-class.html) from the [animation](https://api.flutter.dev/flutter/widgets/MatrixTransition/animation.html). It's called every time [animation](https://api.flutter.dev/flutter/widgets/MatrixTransition/animation.html) changes its value.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[turns](https://api.flutter.dev/flutter/widgets/RotationTransition/turns.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>

The animation that controls the rotation of the child.

read-only

## scale transition

Animates the scale of a transformed widget.

### Constructors

[ScaleTransition](https://api.flutter.dev/flutter/widgets/ScaleTransition/ScaleTransition.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)> scale, [Alignment](https://api.flutter.dev/flutter/painting/Alignment-class.html) alignment = Alignment.center, [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html)? filterQuality, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child})

Creates a scale transition.

const

### Properties

[alignment](https://api.flutter.dev/flutter/widgets/MatrixTransition/alignment.html) → [Alignment](https://api.flutter.dev/flutter/painting/Alignment-class.html)

The alignment of the origin of the coordinate system in which the transform takes place, relative to the size of the box.

finalinherited

[animation](https://api.flutter.dev/flutter/widgets/MatrixTransition/animation.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>

The animation that controls the matrix of the child.

read-onlyinherited

[child](https://api.flutter.dev/flutter/widgets/MatrixTransition/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

finalinherited

[filterQuality](https://api.flutter.dev/flutter/widgets/MatrixTransition/filterQuality.html) → [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html)?

The filter quality with which to apply the transform as a bitmap operation.

finalinherited

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedWidget/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) to which this widget is listening.

finalinherited

[onTransform](https://api.flutter.dev/flutter/widgets/MatrixTransition/onTransform.html) → [TransformCallback](https://api.flutter.dev/flutter/widgets/TransformCallback.html)

The callback to compute a [Matrix4](https://api.flutter.dev/flutter/vector_math_64/Matrix4-class.html) from the [animation](https://api.flutter.dev/flutter/widgets/MatrixTransition/animation.html). It's called every time [animation](https://api.flutter.dev/flutter/widgets/MatrixTransition/animation.html) changes its value.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[scale](https://api.flutter.dev/flutter/widgets/ScaleTransition/scale.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>

The animation that controls the scale of the child.

read-only

## size transition

Animates its own size and clips and aligns its child.

[SizeTransition](https://api.flutter.dev/flutter/widgets/SizeTransition-class.html) acts as a [ClipRect](https://api.flutter.dev/flutter/widgets/ClipRect-class.html) that animates either its width or its height, depending upon the value of [axis](https://api.flutter.dev/flutter/widgets/SizeTransition/axis.html). The alignment of the child along the [axis](https://api.flutter.dev/flutter/widgets/SizeTransition/axis.html) is specified by the [axisAlignment](https://api.flutter.dev/flutter/widgets/SizeTransition/axisAlignment.html).

Like most widgets, [SizeTransition](https://api.flutter.dev/flutter/widgets/SizeTransition-class.html) will conform to the constraints it is given, so be sure to put it in a context where it can change size. For instance, if you place it into a [Container](https://api.flutter.dev/flutter/widgets/Container-class.html) with a fixed size, then the [SizeTransition](https://api.flutter.dev/flutter/widgets/SizeTransition-class.html) will not be able to change size, and will appear to do nothing.

### Constructors

[SizeTransition](https://api.flutter.dev/flutter/widgets/SizeTransition/SizeTransition.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [Axis](https://api.flutter.dev/flutter/painting/Axis.html) axis = Axis.vertical, required [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)> sizeFactor, [double](https://api.flutter.dev/flutter/dart-core/double-class.html) axisAlignment = 0.0, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child})

Creates a size transition.

const

### Properties

[axis](https://api.flutter.dev/flutter/widgets/SizeTransition/axis.html) → [Axis](https://api.flutter.dev/flutter/painting/Axis.html)

[Axis.horizontal](https://api.flutter.dev/flutter/painting/Axis.html) if [sizeFactor](https://api.flutter.dev/flutter/widgets/SizeTransition/sizeFactor.html) modifies the width, otherwise [Axis.vertical](https://api.flutter.dev/flutter/painting/Axis.html).

final

[axisAlignment](https://api.flutter.dev/flutter/widgets/SizeTransition/axisAlignment.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)

Describes how to align the child along the axis that [sizeFactor](https://api.flutter.dev/flutter/widgets/SizeTransition/sizeFactor.html) is modifying.

final

[child](https://api.flutter.dev/flutter/widgets/SizeTransition/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedWidget/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) to which this widget is listening.

finalinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[sizeFactor](https://api.flutter.dev/flutter/widgets/SizeTransition/sizeFactor.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>

The animation that controls the (clipped) size of the child.

read-only

## slide transition

Animates the position of a widget relative to its normal position.

The translation is expressed as an [Offset](https://api.flutter.dev/flutter/dart-ui/Offset-class.html) scaled to the child's size. For example, an [Offset](https://api.flutter.dev/flutter/dart-ui/Offset-class.html) with a dx of 0.25 will result in a horizontal translation of one quarter the width of the child.

By default, the offsets are applied in the coordinate system of the canvas (so positive x offsets move the child towards the right). If a [textDirection](https://api.flutter.dev/flutter/widgets/SlideTransition/textDirection.html) is provided, then the offsets are applied in the reading direction, so in right-to-left text, positive x offsets move towards the left, and in left-to-right text, positive x offsets move towards the right.

### Constructors

[SlideTransition](https://api.flutter.dev/flutter/widgets/SlideTransition/SlideTransition.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[Offset](https://api.flutter.dev/flutter/dart-ui/Offset-class.html)> position, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) transformHitTests = true, [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)? textDirection, [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)? child})

Creates a fractional translation transition.

const

### Properties

[child](https://api.flutter.dev/flutter/widgets/SlideTransition/child.html) → [Widget](https://api.flutter.dev/flutter/widgets/Widget-class.html)?

The widget below this widget in the tree.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[listenable](https://api.flutter.dev/flutter/widgets/AnimatedWidget/listenable.html) → [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html)

The [Listenable](https://api.flutter.dev/flutter/foundation/Listenable-class.html) to which this widget is listening.

finalinherited

[position](https://api.flutter.dev/flutter/widgets/SlideTransition/position.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[Offset](https://api.flutter.dev/flutter/dart-ui/Offset-class.html)>

The animation that controls the position of the child.

read-only

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[textDirection](https://api.flutter.dev/flutter/widgets/SlideTransition/textDirection.html) → [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)?

The direction to use for the x offset described by the [position](https://api.flutter.dev/flutter/widgets/SlideTransition/position.html).

final

[transformHitTests](https://api.flutter.dev/flutter/widgets/SlideTransition/transformHitTests.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether hit testing should be affected by the slide animation.

final