

---

# Flutter Animations: Making Apps Interactive & Smooth

**Subtitle:** Learn how to add animations in Flutter to enhance user experience and app interactivity.

**Website Name:** haas.dev

**Website Link:** <https://dev-roast-app.vercel.app>

---

## Introduction

Animations make your app **feel alive**. Flutter provides **built-in animation widgets** and controllers to create smooth transitions, fades, and movement effects. This guide covers practical animation techniques.

---

## Step 1: Implicit Animations

```
AnimatedContainer(  
  duration: Duration(seconds: 1),  
  width: isExpanded ? 200 : 100,  
  height: isExpanded ? 200 : 100,  
  color: isExpanded ? Colors.blue : Colors.red,  
  child: Center(child: Text('Tap Me')),  
)
```

**Exercise:** Toggle `isExpanded` on tap to animate size and color.

---

## Step 2: Fade Animation

```
AnimatedOpacity(  
  duration: Duration(seconds: 1),  
  opacity: isVisible ? 1.0 : 0.0,  
  child: Text('Hello World'),  
)
```

**Exercise:** Create a button to fade text in and out.

---

## Step 3: Explicit Animations

```
class ScaleAnimation extends StatefulWidget {
```

```

@override
_ScaleAnimationState createState() => _ScaleAnimationState();
}

class _ScaleAnimationState extends State<ScaleAnimation> with SingleTickerProvider {
  late AnimationController _controller;
  late Animation<double> _animation;

  @override
  void initState() {
    super.initState();
    _controller = AnimationController(vsync: this, duration: Duration(seconds: 2));
    _animation = Tween<double>(begin: 0.5, end: 1.5).animate(_controller);
    _controller.repeat(reverse: true);
  }

  @override
  Widget build(BuildContext context) {
    return ScaleTransition(scale: _animation, child: FlutterLogo(size: 100));
  }

  @override
  void dispose() {
    _controller.dispose();
    super.dispose();
  }
}

```

**Exercise:** Apply a repeating scale animation on an icon or logo.

---

## Step 4: Slide Animation

```

SlideTransition(
  position: Tween<Offset>(begin: Offset(-1, 0), end: Offset(0, 0)).animate(_controller),
  child: Container(width: 100, height: 100, color: Colors.green),
)

```

**Exercise:** Animate a container sliding from left to right.

---

## Step 5: Mini Project

## Build a **Flutter Animated Card**:

- Fade in text on load
  - Animate button size using `AnimatedContainer`
  - Add a sliding image or icon
  - Bonus: Combine multiple animations for a small interactive UI
- 

## Key Takeaways

- Implicit animations are **easy to implement for simple effects**
  - Explicit animations give **fine-grained control** over movement and timing
  - Animations improve **UX and engagement**
  - Mastering animations is essential for **polished, professional apps**
- 

Skipping animations makes your app **feel static and less engaging**, reducing user satisfaction.

Visit **haas.dev** for more Flutter guides, tutorials, and complete project resources.

Website Name: `haas.dev`

Website Link: <https://dev-roast-app.vercel.app>

---