

---

# Flutter Animations: Creating Smooth UI Effects

**Subtitle:** Learn how to add animations to Flutter apps for engaging and interactive UI.

**Website Name:** haas.dev

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

---

## Introduction

Animations make apps **visually appealing and interactive**. Flutter provides built-in widgets and controllers to create smooth effects. This guide covers **implicit and explicit animations** for beginners.

---

## Step 1: Implicit Animations

- Flutter's implicit animations automatically animate changes in widget properties.

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

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

---

## Step 2: AnimatedOpacity

```
AnimatedOpacity(  
  opacity: _visible ? 1.0 : 0.0,  
  duration: Duration(seconds: 1),  
  child: Text("Hello Flutter"),  
)
```

**Exercise:** Add a button to toggle visibility of a text widget with fade effect.

---

## Step 3: Explicit Animations with AnimationController

```
class MyHomePage extends StatefulWidget { ... }
```

```

class _MyHomePageState extends State<MyHomePage>
  with SingleTickerProviderStateMixin {

  late AnimationController _controller;

  @override
  void initState() {
    super.initState();
    _controller = AnimationController(
      duration: Duration(seconds: 2),
      vsync: this,
    )..repeat(reverse: true);
  }

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

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

```

**Exercise:** Animate a Flutter logo to grow and shrink continuously.

---

## Step 4: Tween Animations

```

Animation<double> animation = Tween<double>(begin: 0, end: 300).animate(_control

```

- Use `Tween` to define start and end values for animations
- Combine with `AnimatedBuilder` to update widgets

**Exercise:** Animate a box moving from left to right using Tween + AnimationController.

---

## Step 5: Mini Project

## Build a **Simple Animated Card App**:

- Use `AnimatedContainer` to animate color and size on tap
  - Add `AnimatedOpacity` to fade in text
  - Use `AnimationController` to create bouncing Flutter logo
  - Combine multiple animations for a smooth interactive UI
- 

## Key Takeaways

- Implicit animations are **easy and automatic**
  - Explicit animations with controllers provide **full control**
  - Tween + `AnimatedBuilder` help **animate any property**
  - Animations make apps **engaging and professional**
  - Mastering animations improves **UX and app polish**
- 

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

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

Website Name: haas.dev

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

---