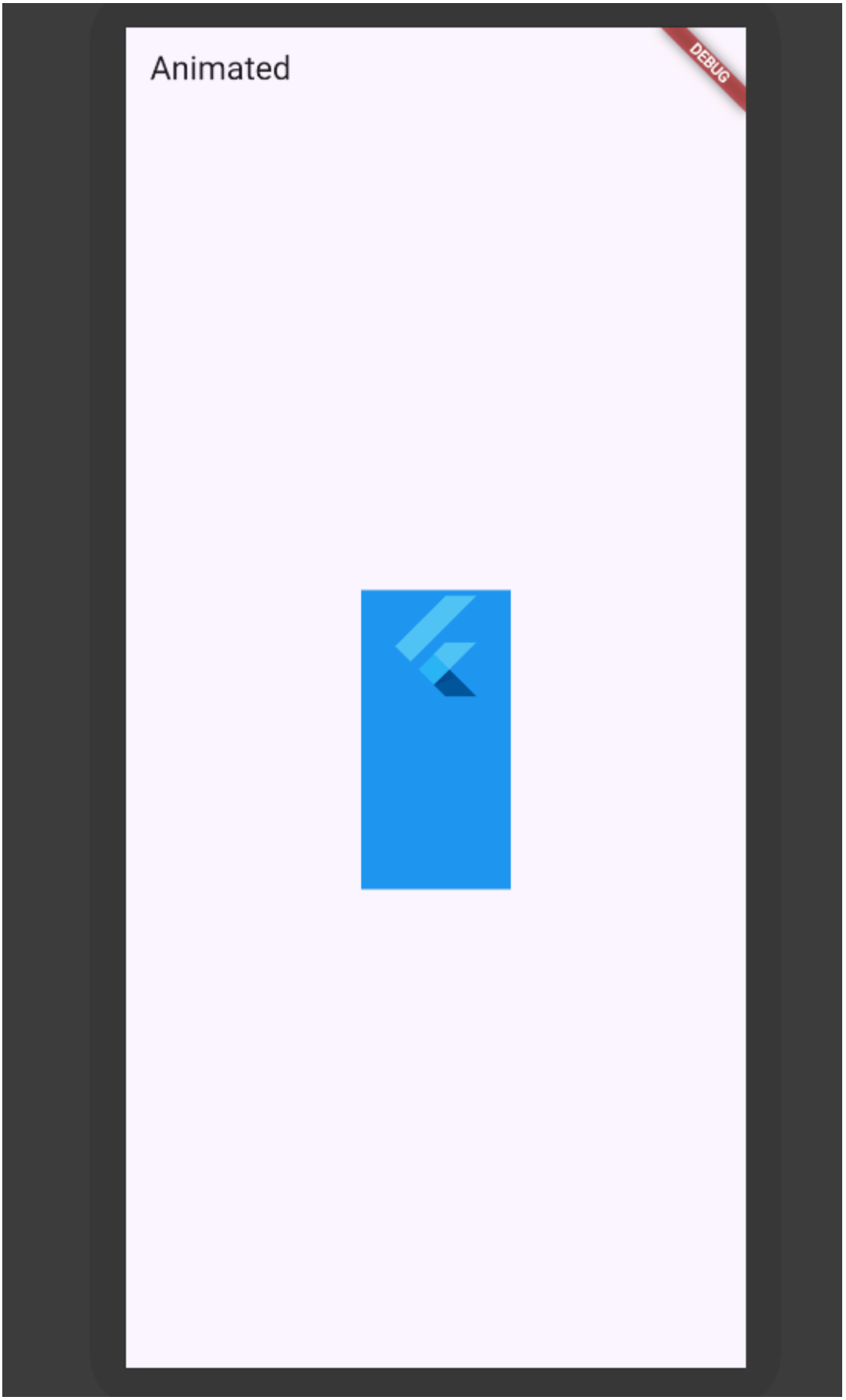
1.

```
import 'package:flutter/material.dart';
/// Flutter code sample for [AnimatedContainer].
void main() => runApp(const
AnimatedContainerExampleApp());
class AnimatedContainerExampleApp extends
StatelessWidget {
const AnimatedContainerExampleApp({super.key});
@override
Widget build(BuildContext context) {
return MaterialApp(
home: Scaffold(
appBar: AppBar(title: const Text('AnimatedContainer
Sample')),
body: const AnimatedContainerExample(),
),
);
}
}
class AnimatedContainerExample extends
StatefulWidget {
const AnimatedContainerExample({super.key});
@override
State<AnimatedContainerExample> createState()
=> _AnimatedContainerExampleState();
}
class _AnimatedContainerExampleState extends
State<AnimatedContainerExample> {

bool selected = false;
@override
Widget build(BuildContext context) {
return GestureDetector(
onTap: () {
setState(() {
selected = !selected;
});
},
child: Center(
child: AnimatedContainer(
width: selected ? 200.0 : 100.0,
height: selected ? 100.0 : 200.0,
color: selected ? Colors.red : Colors.blue,
alignment: selected ? Alignment.center :
AlignmentDirectional.topCenter,
duration: const Duration(seconds: 2),
```

```
curve: Curves.fastOutSlowIn,
child: const FlutterLogo(size: 75),
),
),
);
}
}
```

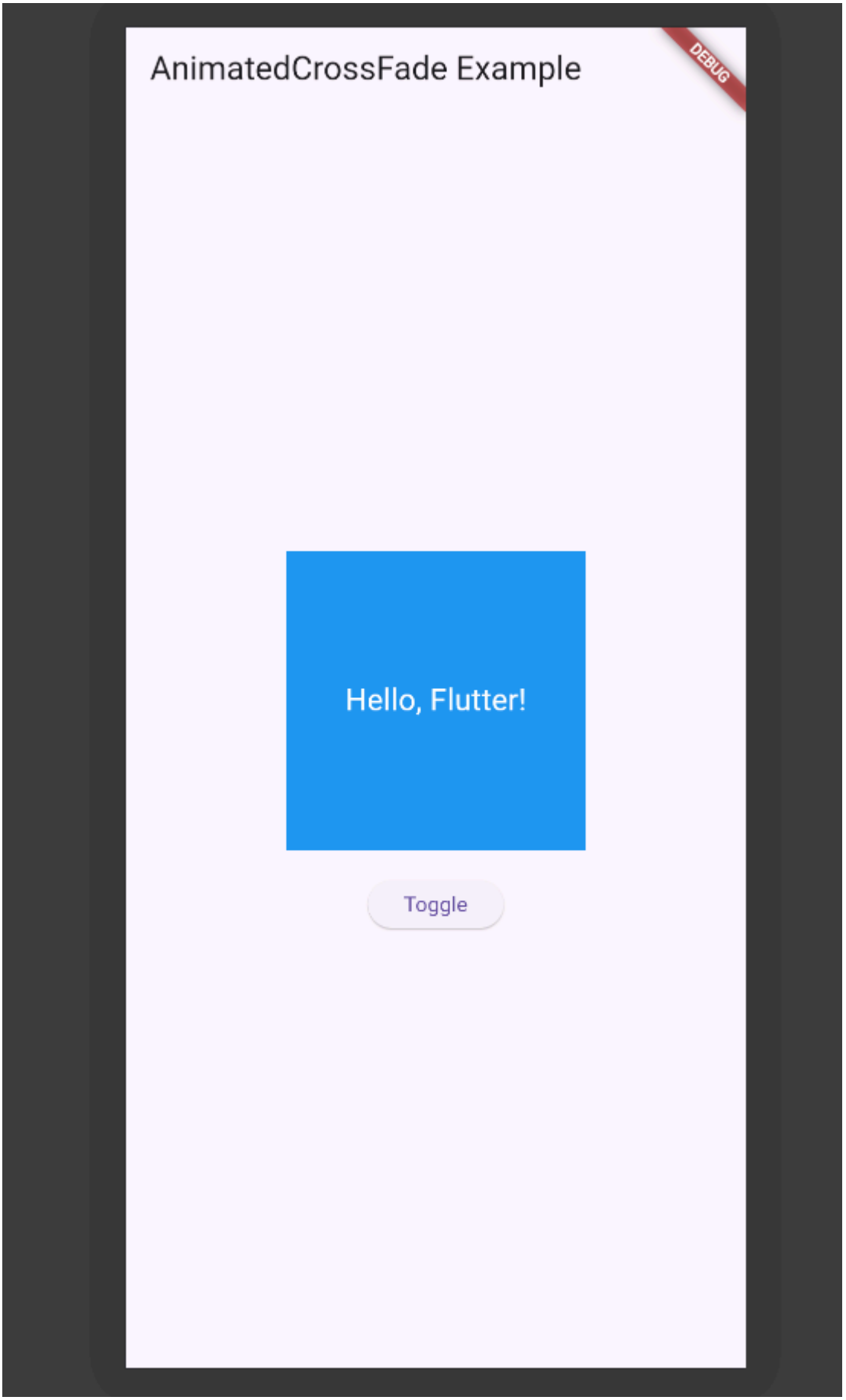

Animated
DEBUG

2.

```
import 'package:flutter/material.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'AnimatedCrossFade Example'
,

theme: ThemeData(
primarySwatch: Colors.blue,
),
home: AnimatedCrossFadePage(),
);
}
}
class AnimatedCrossFadePage extends StatefulWidget {
@override
_AnimatedCrossFadePageState createState() =>
_AnimatedCrossFadePageState();
}
class _AnimatedCrossFadePageState extends
State<AnimatedCrossFadePage> {
bool _showText = true; // Variable to toggle between

@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(

title: Text('AnimatedCrossFade Example'),
),
body: Center(
child: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
AnimatedCrossFade(
firstChild: Container(
color: Colors.blue,
```

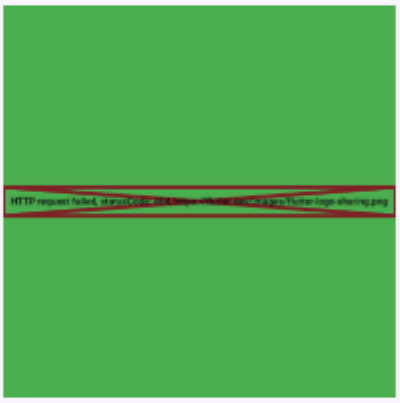
```
width: 200,
height: 200,
child: Center(
child: Text(
'Hello, Flutter!'
,

style: TextStyle(color: Colors.white,

fontSize: 20),
),
),
),
secondChild: Container(
color: Colors.green,
width: 200,
height: 200,
child: Center(
child: Image.network(

'https://flutter.dev/images/flutter-logo-sharing.png'


,
width: 100,
height: 100,
),
),
),
crossFadeState: _showText
? CrossFadeState.showFirst
: CrossFadeState.showSecond,

duration: Duration(seconds: 1),
),
SizedBox(height: 20),
ElevatedButton(
onPressed: () {
setState(() {
_showText = !_showText;
});
},
child: Text('Toggle'),
),
```

```
],
),
),
);
}
}
```


AnimatedCrossFade Example
DEBUG
Hello, Flutter!
Toggle

AnimatedCrossFade Example

DEBUG



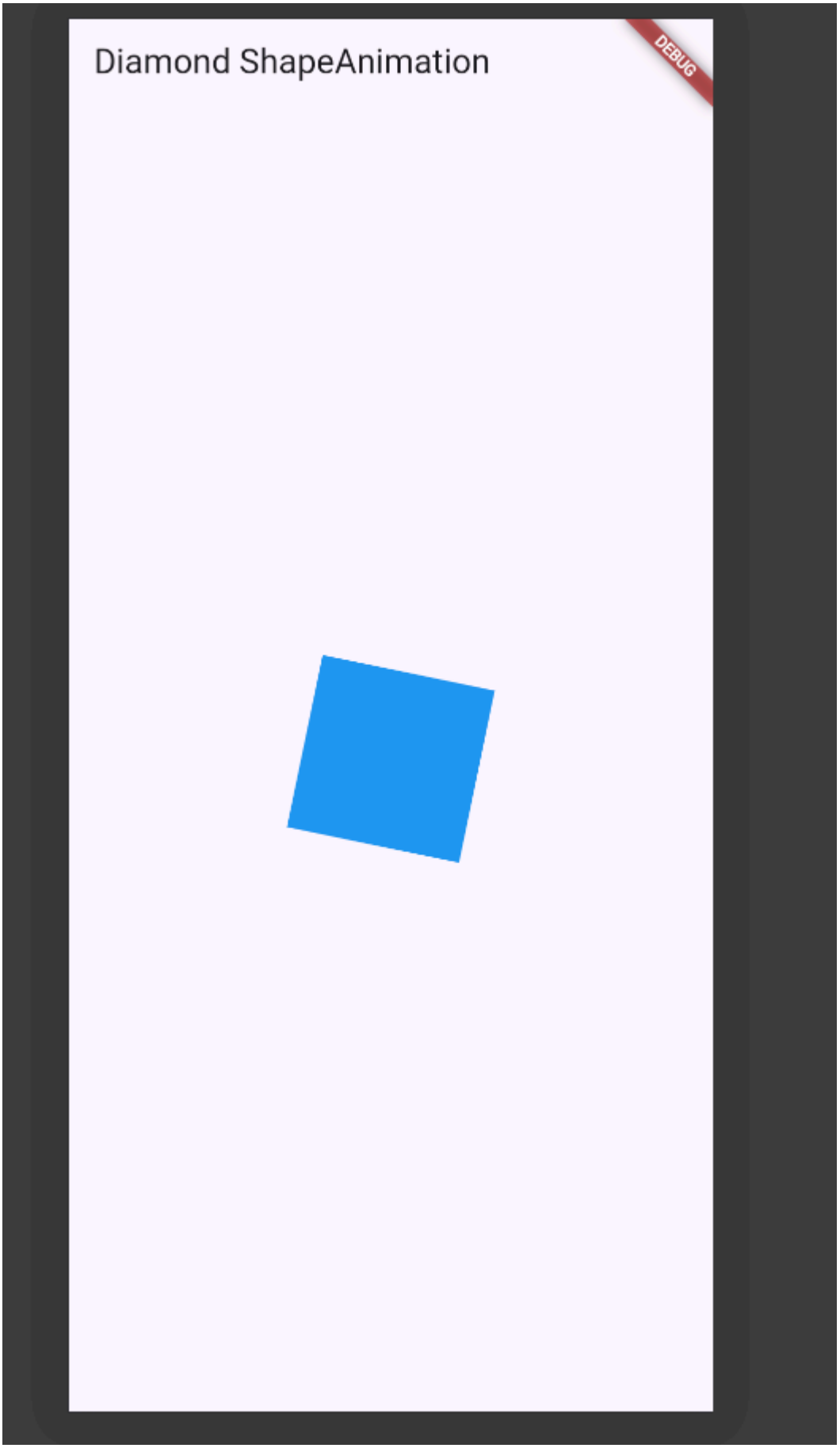
Toggle

3.

```
import 'package:flutter/material.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'Diamond Shape Tween Animation'
,

theme: ThemeData(
primarySwatch: Colors.blue,
),
home: DiamondAnimationPage(title: 'Diamond ShapeAnimation'),
);
}
}
class DiamondAnimationPage extends StatefulWidget {
DiamondAnimationPage({Key? key, required this.title}) :
super(key: key);
final String title;
@override
_DiamondAnimationPageState createState() =>
_DiamondAnimationPageState();
}
class _DiamondAnimationPageState extends
State<DiamondAnimationPage> with
TickerProviderStateMixin {
late AnimationController _controller;
late Animation<double> _sizeAnimation;
late Animation<double> _rotationAnimation;
@override

void initState() {
super.initState();
// Initialize the animation controller with a duration of 2seconds
_controller = AnimationController(
vsync: this,
duration: Duration(seconds: 2),
)
```

```
..addListener(() => setState(() {}))
..addStatusListener((status) {
if (status == AnimationStatus.completed) {
_controller.reverse();
} else if (status == AnimationStatus.dismissed) {
_controller.forward();
}
});
// Tween for scaling (size) from 1.0 to 1.5 for the diamond shape
effect
_sizeAnimation = Tween<double>(begin: 1.0, end:
1.5).animate(CurvedAnimation(
parent: _controller,
curve: Curves.easeInOut,
));
// Tween for rotating (angle) from 0 to 45 degrees

_rotationAnimation = Tween<double>(begin: 0, end:
0.7854).animate(CurvedAnimation(
parent: _controller,
curve: Curves.easeInOut,
));
// Start the animation in forward direction
_controller.forward();
}
@override

void dispose() {
_controller.dispose();
super.dispose();
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text(widget.title),
),
body: Center(
child: Transform.rotate(
angle: _rotationAnimation.value, // Apply rotation based

child: Transform.scale(
scale: _sizeAnimation.value, // Apply scaling based on
```

```
child: Container(
width: 100,
height: 100,
color: Colors.blue,
),
),
),
),
);
}
}
```


Diamond ShapeAnimation
DEBUG

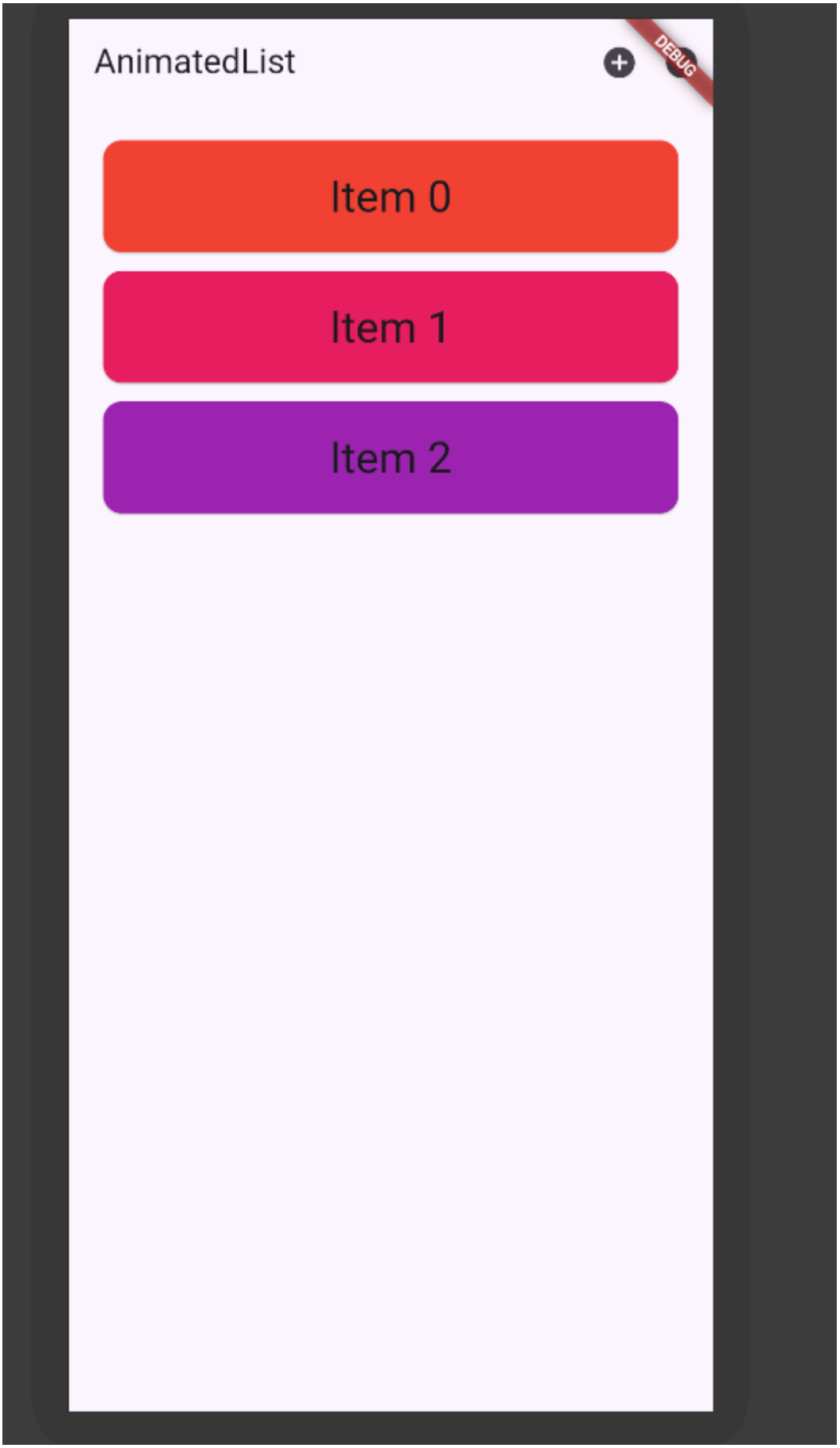
4.

```
import 'package:flutter/material.dart';

/// Flutter code sample for [AnimatedList].
void main() {
  runApp(const AnimatedListSample());
}

class AnimatedListSample extends StatefulWidget {
  const AnimatedListSample({super.key});

  @override
  State<AnimatedListSample> createState() => _AnimatedListSampleState();
}

class _AnimatedListSampleState extends State<AnimatedListSample> {
  final GlobalKey<AnimatedListState> _listKey = GlobalKey<AnimatedListState>();
  late ListModel<int> _list;
  int? _selectedItem;
  late int _nextItem; // The next item inserted when the user presses '+'.

  @override
  void initState() {
    super.initState();
    _list = ListModel<int>(
      listKey: _listKey,
      initialItems: <int>[0, 1, 2],
      removedItemBuilder: _buildRemovedItem,
    );
    _nextItem = 3;
  }

  // Used to build list items that haven't been removed.
  Widget _buildItem(BuildContext context, int index, Animation<double> animation) {
    return CardItem(
      animation: animation,
      item: _list[index],
      selected: _selectedItem == _list[index],
      onTap: () {
        setState(() {
          _selectedItem = _selectedItem == _list[index] ? null : _list[index];
        });
      },
    );
  }

  /// Builder function for removed items.
  Widget _buildRemovedItem(int item, BuildContext context, Animation<double> animation) {
```

```
  return CardItem(
    animation: animation,
    item: item,
  );
}

// Insert the "next item".
void _insert() {
  final int index = _selectedItem == null ? _list.length : _list.indexOf(_selectedItem!);
  _list.insert(index, _nextItem);
  _nextItem++;
}

// Remove the selected item.
void _remove() {
  if (_selectedItem != null) {
    _list.removeAt(_list.indexOf(_selectedItem!));
    setState(() {
      _selectedItem = null;
    });
  }
}

@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: const Text('AnimatedList'),
        actions: <Widget>[
          IconButton(
            icon: const Icon(Icons.add_circle),
            onPressed: _insert,
            tooltip: 'Insert a new item',
          ),
          IconButton(
            icon: const Icon(Icons.remove_circle),
            onPressed: _remove,
            tooltip: 'Remove the selected item',
          ),
        ],
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: AnimatedList(
          key: _listKey,
          initialItemCount: _list.length,
          itemBuilder: _buildItem,
```

```
          ),
        ),
      ),
    );
  }
}

typedef RemovedItemBuilder<T> = Widget Function(T item, BuildContext context,
Animation<double> animation);

/// Keeps a Dart [List] in sync with an [AnimatedList].
class ListModel<E> {
  ListModel({required this.listKey, required this.removedItemBuilder, Iterable<E>?
initialItems})
      : _items = List<E>.from(initialItems ?? <E>[]);

  final GlobalKey<AnimatedListState> listKey;
  final RemovedItemBuilder<E> removedItemBuilder;
  final List<E> _items;

  AnimatedListState? get _animatedList => listKey.currentState;

  void insert(int index, E item) {
    _items.insert(index, item);
    _animatedList!.insertItem(index);
  }

  E removeAt(int index) {
    final E removedItem = _items.removeAt(index);
    _animatedList!.removeItem(
      index,
      (BuildContext context, Animation<double> animation) {
        return removedItemBuilder(removedItem, context, animation);
      },
    );
    return removedItem;
  }

  int get length => _items.length;

  E operator [](int index) => _items[index];

  int indexOf(E item) => _items.indexOf(item);
}

/// Displays integer item as a Card whose color is based on its value.
class CardItem extends StatelessWidget {
  const CardItem({
```

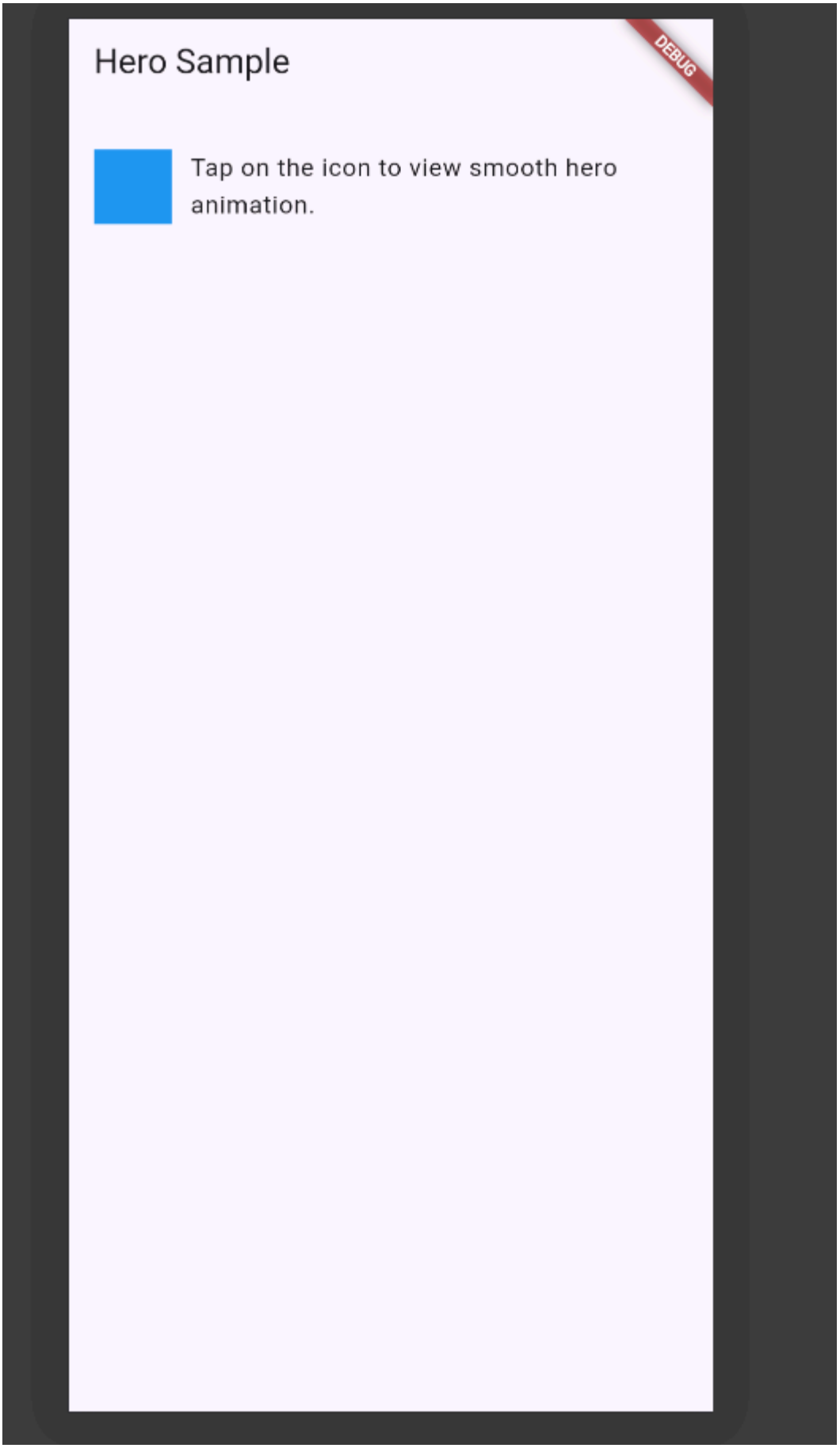
```
    super.key,
    this.onTap,
    this.selected = false,
    required this.animation,
    required this.item,
  }) : assert(item >= 0);

  final Animation<double> animation;
  final VoidCallback? onTap;
  final int item;
  final bool selected;

  @override
  Widget build(BuildContext context) {
    TextStyle textStyle = Theme.of(context).textTheme.headlineMedium!;
    if (selected) {
      textStyle = textStyle.copyWith(color: Colors.lightGreenAccent[400]);
    }

    return Padding(
      padding: const EdgeInsets.all(2.0),
      child: SizeTransition(
        sizeFactor: animation,
        child: GestureDetector(
          behavior: HitTestBehavior.opaque,
          onTap: onTap,
          child: SizedBox(
            height: 80.0,
            child: Card(
              color: Colors.primaries[item % Colors.primaries.length],
              child: Center(
                child: Text('Item $item', style: textStyle),
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

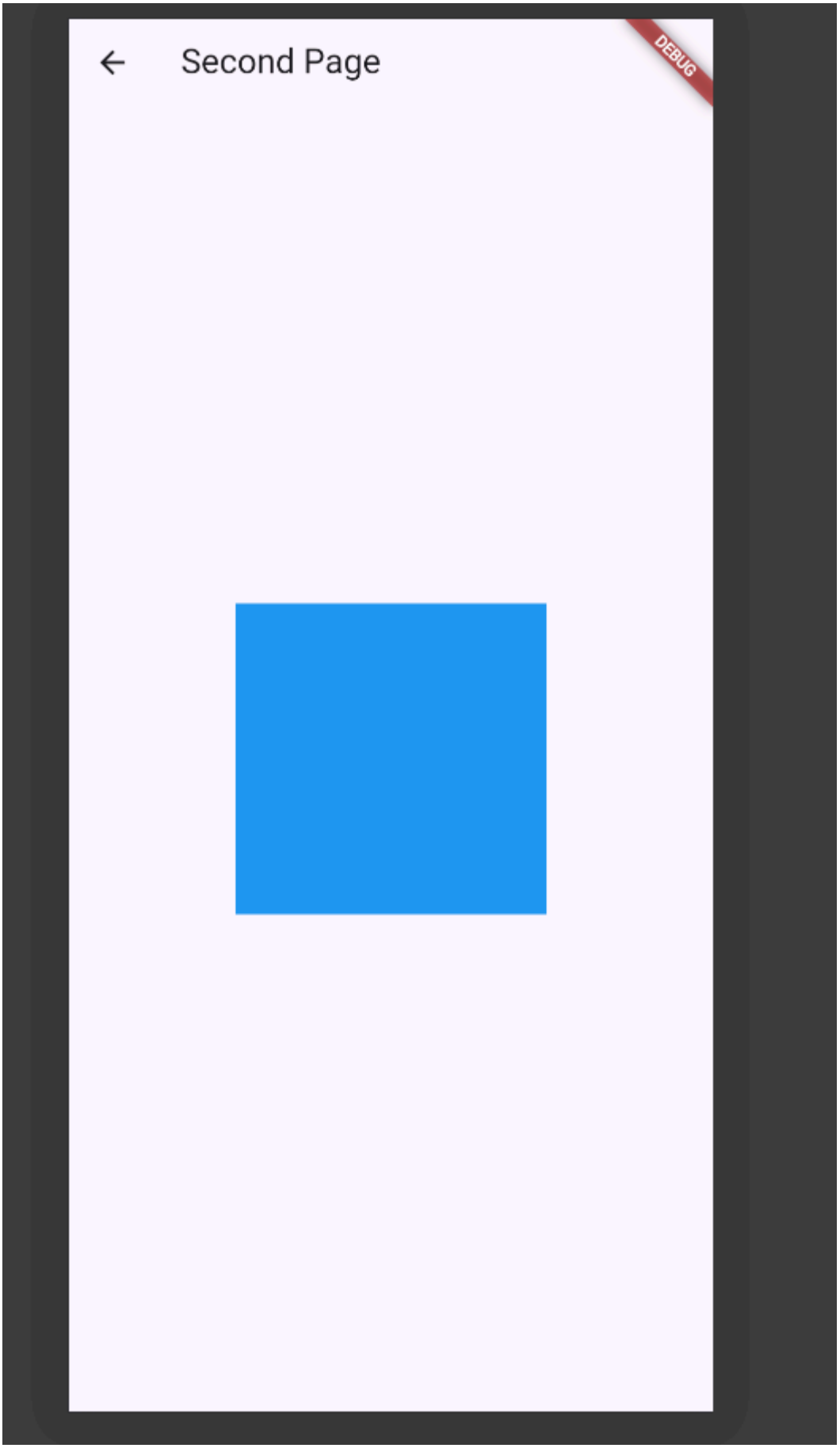

AnimatedList
DEBUG
Item 0
Item 1
Item 2

5.

```
import 'package:flutter/material.dart';

void main() => runApp(const HeroApp());

class HeroApp extends StatelessWidget {
  const HeroApp({super.key});

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(home: HeroExample());
  }
}

class HeroExample extends StatelessWidget {
  const HeroExample({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Hero Sample')),
      body: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          const SizedBox(height: 20.0),
          ListTile(
            leading: const Hero(
              tag: 'hero-rectangle',
              child: BoxWidget(size: Size(50.0, 50.0)),
            ),
            onTap: () => _gotoDetailsPage(context),
            title: const Text('Tap on the icon to view smooth hero animation.'),
          ),
        ],
      ),
    );
  }

  void _gotoDetailsPage(BuildContext context) {
    Navigator.of(context).push(
      PageRouteBuilder(
        transitionDuration: const Duration(seconds: 2), // Slower animation
        pageBuilder: (context, animation, secondaryAnimation) {
          return Scaffold(
            appBar: AppBar(title: const Text('Second Page')),
            body: const Center(
              child: Hero(
                tag: 'hero-rectangle',
```

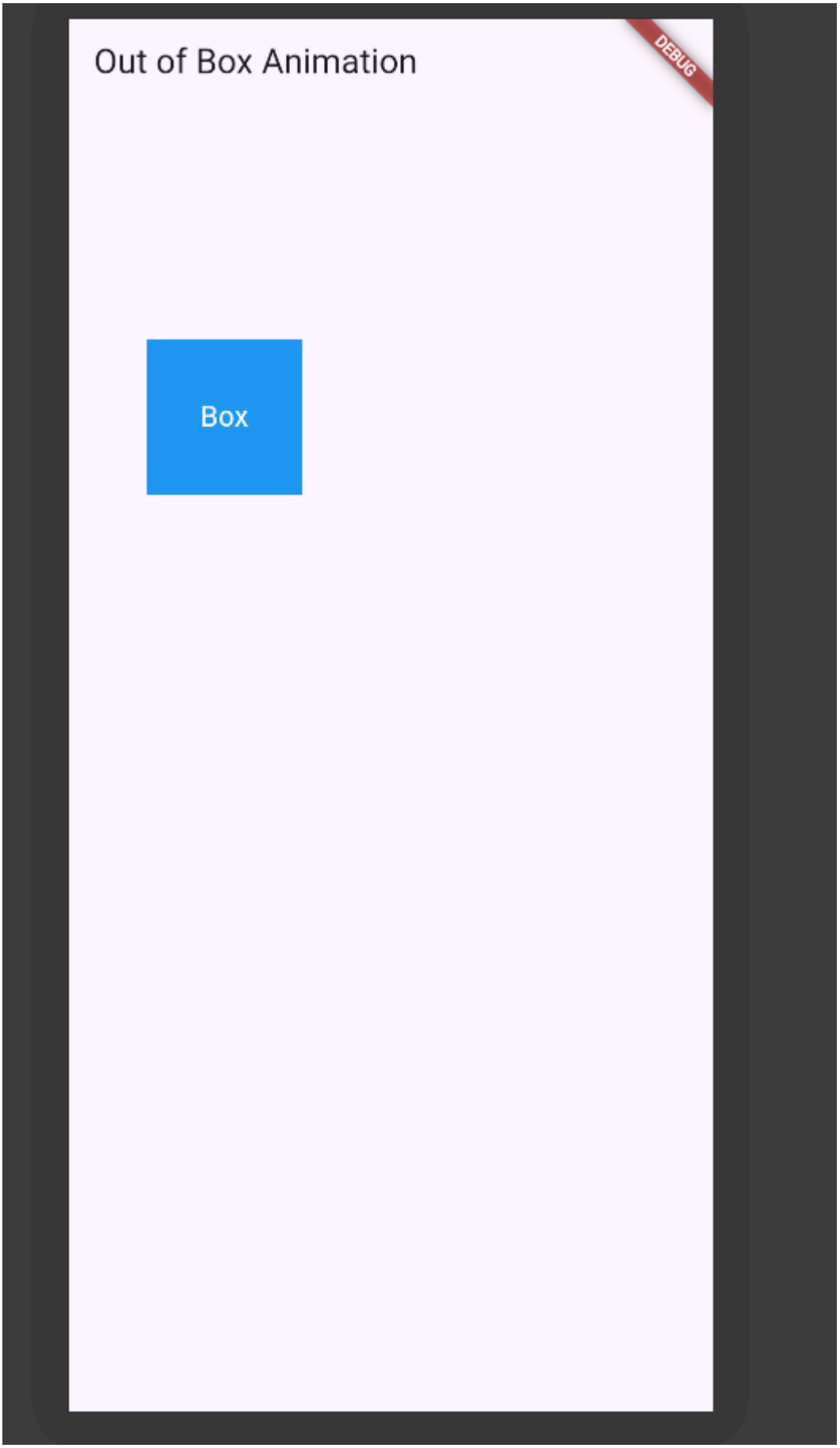
```
              child: BoxWidget(size: Size(200.0, 200.0)),
            ),
          ),
        );
      },
    ),
  );
 }
}

class BoxWidget extends StatelessWidget {
 const BoxWidget({super.key, required this.size});

 final Size size;

 @override
 Widget build(BuildContext context) {
  return Container(
   width: size.width,
   height: size.height,
   color: Colors.blue,
  );
 }
}
```


Hero Sample
DEBUG
Tap on the icon to view smooth hero animation.


Second Page
DEBUG

```
6.import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: OutOfBoxAnimationPage(),
    );
  }
}

class OutOfBoxAnimationPage extends StatefulWidget {
  @override
  _OutOfBoxAnimationPageState createState() => _OutOfBoxAnimationPageState();
}

class _OutOfBoxAnimationPageState extends State<OutOfBoxAnimationPage> {
  double _leftPosition = 50.0;
  bool _isOutOfBox = false;

  void _moveOutOfBox() {
    setState(() {
      _isOutOfBox = !_isOutOfBox;
      if (_isOutOfBox) {
        _leftPosition = MediaQuery.of(context).size.width; // move right
      } else {
        _leftPosition = 50.0; // reset
      }
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Out of Box Animation'),
      ),
      body: GestureDetector(
        onTap: _moveOutOfBox,
        child: Stack(
          children: [
            AnimatedPositioned(
              duration: Duration(seconds: 2),
              curve: Curves.easeInOut,
```

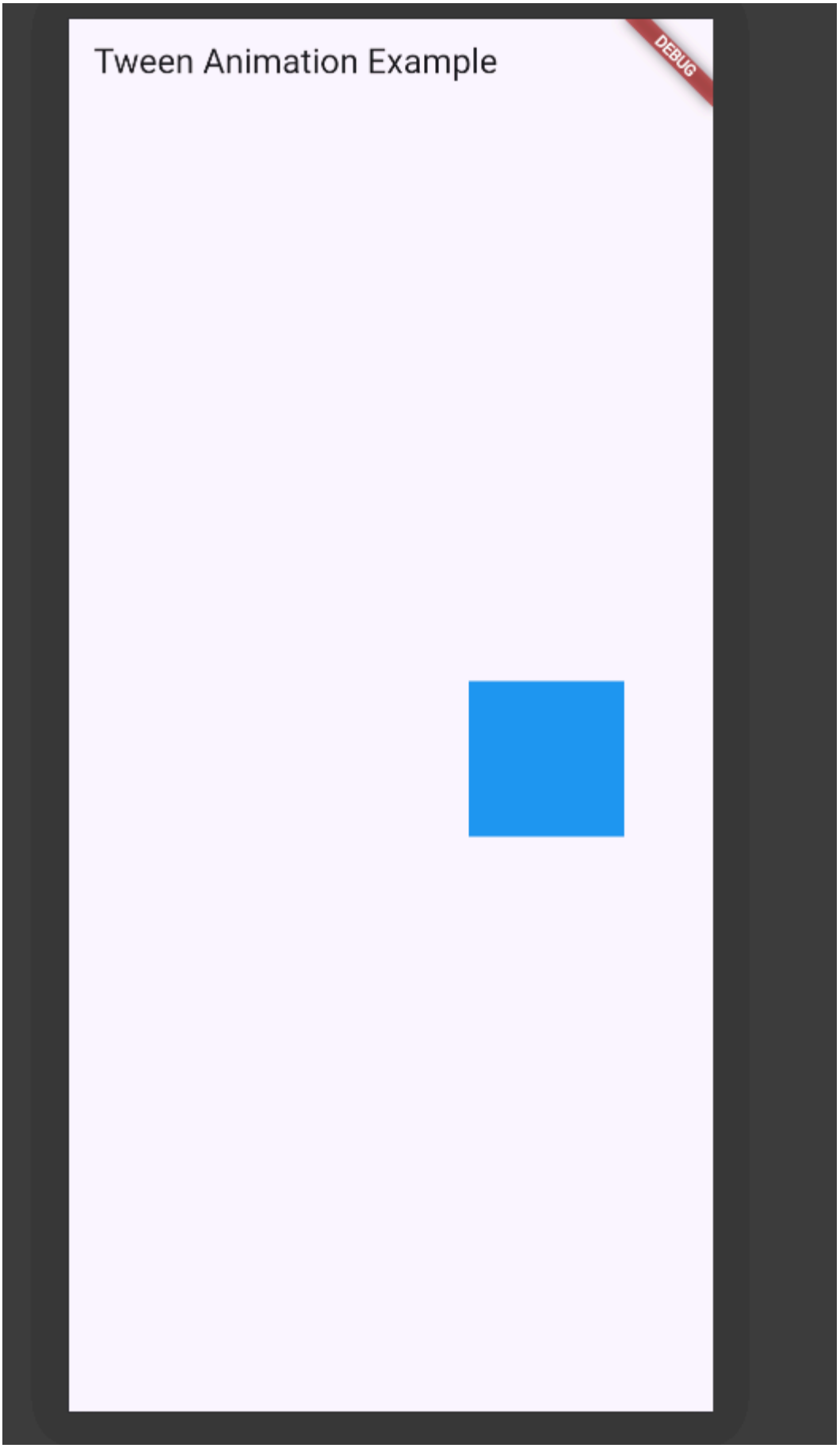
```
            left: _leftPosition,
            top: 150.0,
            child: Container(
              width: 100,
              height: 100,
              color: Colors.blue,
              child: Center(
                child: Text(
                  'Box',
                  style: TextStyle(color: Colors.white, fontSize: 18),
                ),
              ),
            ),
          ),
        ],
      ),
    ),
  );
 }
}
```


Out of Box Animation
DEBUG
Box

7.

```
import 'package:flutter/material.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
home: TweenAnimationExample(),
);
}
}
class TweenAnimationExample extends StatefulWidget {
@override
_TweenAnimationExampleState createState() => _TweenAnimationExampleState();
}
class _TweenAnimationExampleState extends State<TweenAnimationExample>
with SingleTickerProviderStateMixin {
late AnimationController _controller;
late Animation<Offset> _animation;
@override
void initState() {
super.initState();
// Initialize AnimationController
_controller = AnimationController(
duration: Duration(seconds: 3),
vsync: this,
);
// Create a Tween for Offset (move along x-axis)
_animation = Tween<Offset>(
begin: Offset(0, 0), // start position
end: Offset(1, 0), // end position (move to the right)
).animate(CurvedAnimation(
parent: _controller,
curve: Curves.easeInOut, // smooth transition curve
));
// Start the animation
_controller.forward();
}
@override
void dispose() {
_controller.dispose();
super.dispose();
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text('Tween Animation Example'),
),
body: Center(
child: SlideTransition(
position: _animation, // Use the Tween animation here
child: Container(
width: 100,
```

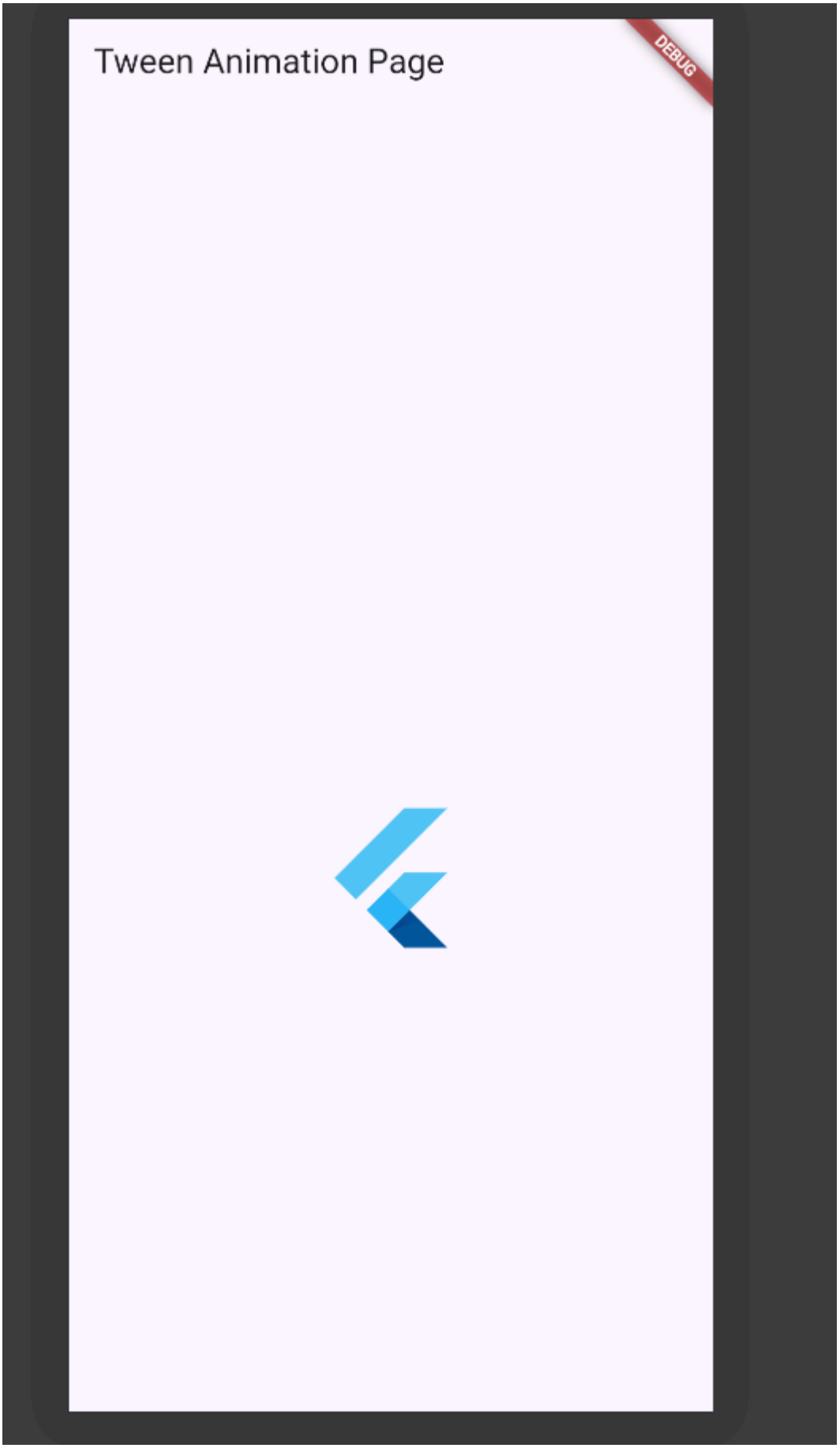
```
height: 100,
color: Colors.blue,
),
),
),
);
}
}
```


Tween Animation Example
DEBUG

8.

```
import 'package:flutter/material.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'Tween Animation Example',
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: TweenAnimationPage(title: 'Tween Animation
Page'),
);
}
}
class TweenAnimationPage extends StatefulWidget {
TweenAnimationPage({Key? key, required
this.title}) : super(key: key);
final String title;
@override
_TweenAnimationPageState createState() =>
_TweenAnimationPageState();
}
class _TweenAnimationPageState extends State<TweenAnimationPage> with
SingleTickerProviderStateMixin {
late AnimationController _simpleAnimController;
late Animation<double> _simpleAnim;
@override
void initState() {
super.initState();
_simpleAnimController = AnimationController(duration: Duration(milliseconds: 1000), vsync:
this)
..addListener(() => setState(() {}))
..addStatusListener((animStatus) {
if (animStatus == AnimationStatus.completed)
_simpleAnimController.reverse();
if (animStatus == AnimationStatus.dismissed)
_simpleAnimController.forward();
});
_simpleAnim = Tween<double>(
begin: 50,
end: 300,
).animate(_simpleAnimController);
_simpleAnimController.forward();
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text(widget.title),
),
body: Stack(children: <Widget>[
Center(
child: Padding(
padding: EdgeInsets.only(top: _simpleAnim.value),
child: Container(
width: 100,
height: 100,
child: FlutterLogo(),
),
```

```
),
),
]),
);
}
}
```


Tween Animation Page
DEBUG