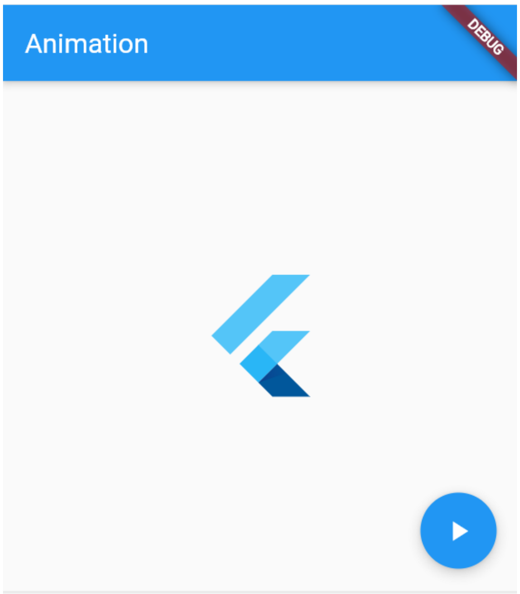
8 a) Add animations to UI elements using Flutter's animation framework.

```
import 'package:flutter/material.dart';
void main() {
 runApp(MyApp());
}
class MyApp extends StatelessWidget {
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'Simple Animation',
     home: SimpleAnimation(),
   );
 }
}
class SimpleAnimation extends StatefulWidget {
 @override
 _SimpleAnimationState createState() => _SimpleAnimationState();
}
class _SimpleAnimationState extends State<SimpleAnimation> with
TickerProviderStateMixin {
 late AnimationController _controller;
 late Animation<double> _animation;
 @override
 void initState() {
   super.initState();
   _controller = AnimationController(duration: Duration(seconds: 2), vsync: this);
   _animation = Tween(begin: 0.0, end: 100.0).animate(_controller);
 }
 @override
 void dispose() {
   _controller.dispose();
   super.dispose();
  }
  @override
  Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: Text('Animation'),
     ),
     body: Center(
       child: AnimatedBuilder(
         animation: _animation,
         builder: (context, child) {
           return Container(
             width: _animation.value,
             height: _animation.value,
             child: FlutterLogo(),
           );
         },
       ),
     ),
     floatingActionButton: FloatingActionButton(
       onPressed: () {
         _controller.forward();
       },
       child: Icon(Icons.play_arrow),
     ),
   );
  }
}
```

Output:



8 b) Experiment with different types of animations (fade, slide, etc.).

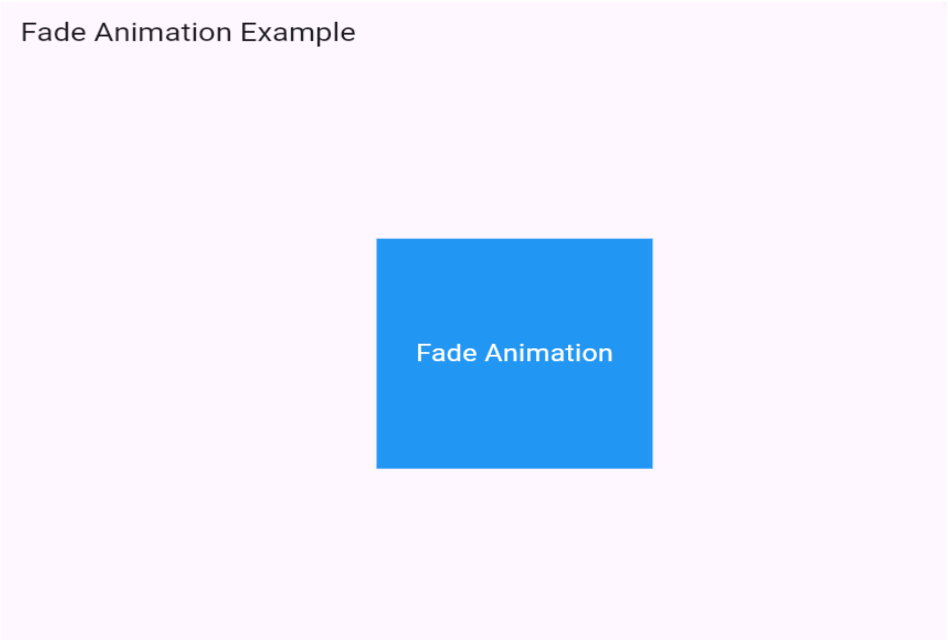
Fade Animation

```
import 'package:flutter/material.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'Fade Animation Example',
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: FadeAnimationWidget(),
);
```

```
}
}
class FadeAnimationWidget extends StatefulWidget {
@override
_FadeAnimationWidgetState createState() => _FadeAnimationWidgetState();
}
class _FadeAnimationWidgetState extends State<FadeAnimationWidget>
with SingleTickerProviderStateMixin {
late AnimationController _animationController;
late Animation<double> _opacityAnimation;
@override
void initState() {
super.initState();
_animationController = AnimationController(
vsync: this,
duration: Duration(seconds: 10),
);
_opacityAnimation = Tween<double>(begin: 0.0, end: 1.0).animate(
CurvedAnimation(
parent: _animationController,
curve: Curves.easeInOut,
),
);
_animationController.forward();
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text('Fade Animation Example'),
),
body: Center(
child: FadeTransition(
opacity: _opacityAnimation,
child: Container(
width: 200,
height: 200,
color: Colors.blue,
child: Center(
child: Text(
'Fade Animation',
style: TextStyle(
color: Colors.white,
fontSize: 20,
),
),
),
),
),
),
);
}
@override
void dispose() {
_animationController.dispose();
super.dispose();
}}
```

Output:

```
}
```

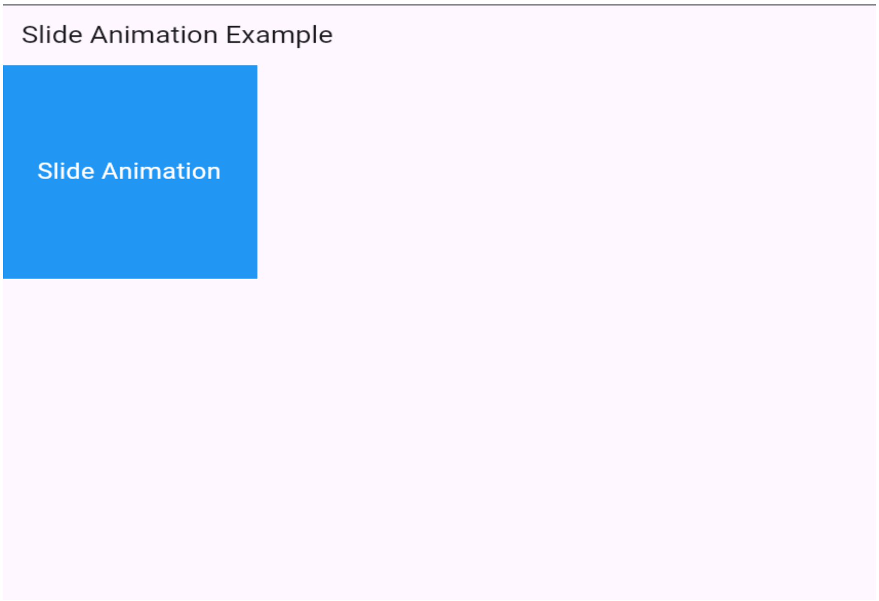


Slide Animation:

```
import 'package:flutter/material.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'Slide Animation Example',
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: SlideAnimationWidget(),
);
}
}
class SlideAnimationWidget extends StatefulWidget {
```

```
@override
_SlideAnimationWidgetState createState() => _SlideAnimationWidgetState();
}
class _SlideAnimationWidgetState extends State<SlideAnimationWidget>
with SingleTickerProviderStateMixin {
late AnimationController _animationController;
late Animation<Offset> _slideAnimation;
@override
void initState() {
super.initState();
_animationController = AnimationController(
vsync: this,
duration: Duration(seconds: 2),
);
_slideAnimation = Tween<Offset>(
begin: Offset(-1.0, 0.0),
end: Offset(0.0, 0.0),
).animate(
CurvedAnimation(
parent: _animationController,
curve: Curves.easeInOut,
),
);
_animationController.forward();
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text('Slide Animation Example'),
),
body: SlideTransition(
position: _slideAnimation,
child: Container(
width: 200,
height: 200,
color: Colors.blue,
child: Center(
child: Text(
'Slide Animation',
style: TextStyle(
color: Colors.white,
fontSize: 20,
),
),
),
),
),
);
}
@override
void dispose() {
_animationController.dispose();
super.dispose();
}
}
```

Output:



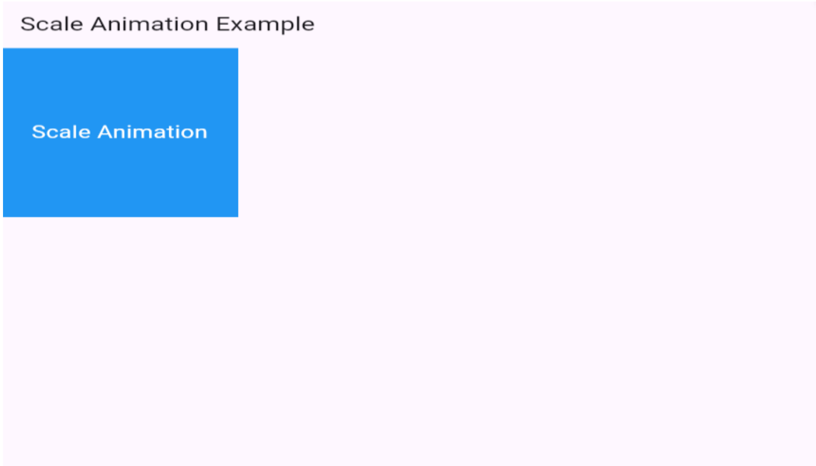
Slide Animation Example

Scale Animation:

```
import 'package:flutter/material.dart';
void main() {
runApp(MyApp());
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
title: 'Scale Animation Example',
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: ScaleAnimationWidget(),
);
}
}
class ScaleAnimationWidget extends StatefulWidget {
@override
```

```
_ScaleAnimationWidgetState createState() => _ScaleAnimationWidgetState();
}
class _ScaleAnimationWidgetState extends State<ScaleAnimationWidget>
with SingleTickerProviderStateMixin {
late AnimationController _animationController;
late Animation<double> _scaleAnimation;
@override
void initState() {
super.initState();
_animationController = AnimationController(
vsync: this,
duration: Duration(seconds: 2),
);
_scaleAnimation = Tween<double>(begin: 0.5, end: 1.0).animate(
CurvedAnimation(
parent: _animationController,
curve: Curves.easeInOut,
),
);
_animationController.forward();
}
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text('Scale Animation Example'),
),
body: ScaleTransition(
scale: _scaleAnimation,
child: Container(
width: 200,
height: 200,
color: Colors.blue,
child: Center(
child: Text(
'Scale Animation',
style: TextStyle(
color: Colors.white,
fontSize: 20,
),
),
),
),
),
);
}
@override
void dispose() {
_animationController.dispose();
super.dispose();
}
}
```

Output:



Scale Animation Example