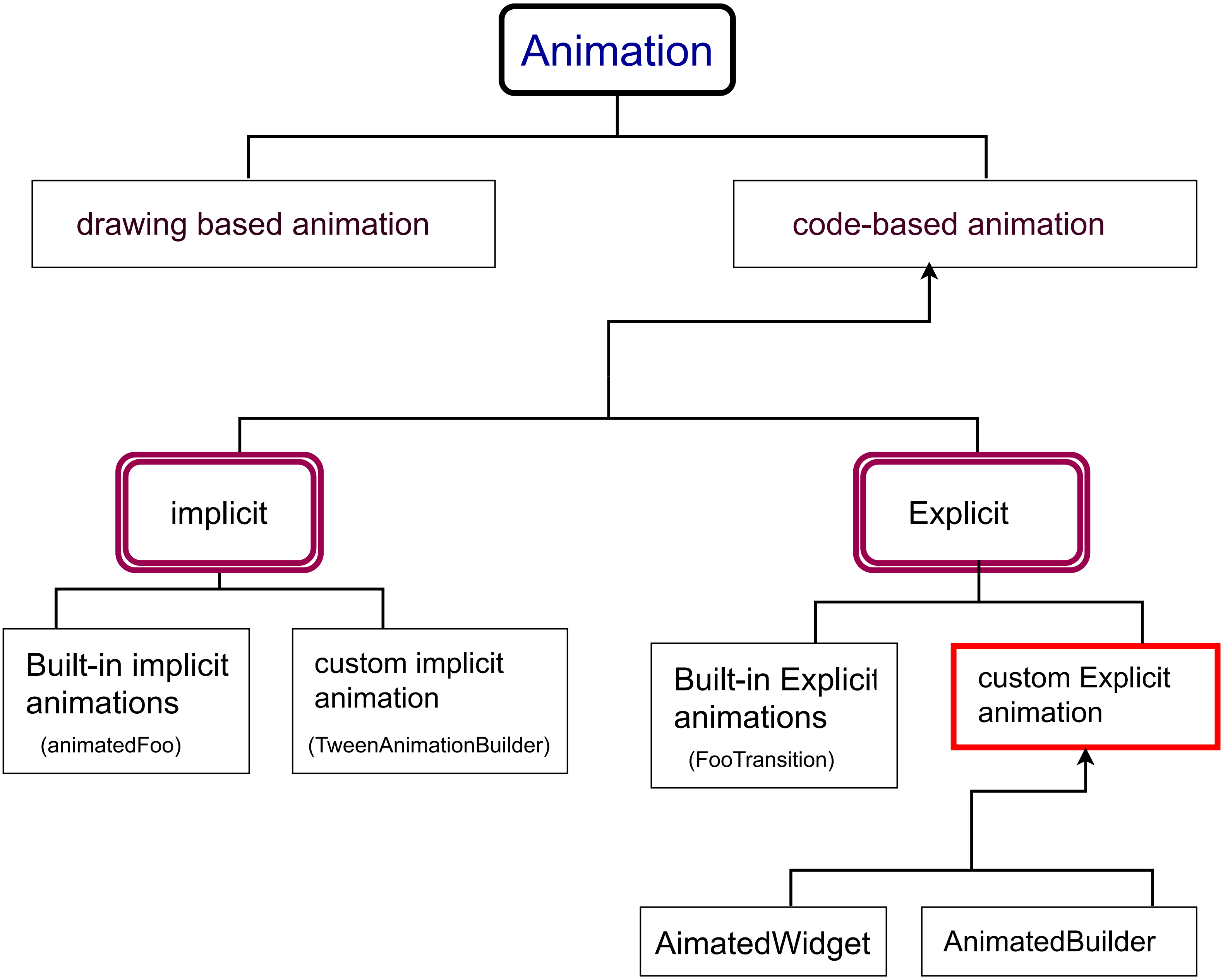


شرح الشكل التخطيطي

[1] drawing based animation => يعتمد علي رسم اشكال او انميشن لاشكال محددة ممكن نستخدمها من ال مكتبات الخارجيه package

[2] code based animation => يعتمد علي تغير قيم لمتغيرات ومنها بيعمل حركه في الويدجت المعروفة متل الكونتينر container

[2-1] built in implicit animation => بيعتمد على ويدجت جاهزه وبيكون سهل في المعامله زي AnimatedContainer

بتمثل الويتجد الذي نريد عمل الانميشن عليه FOO => AnimatedFoo بيطلق عليه ==> AnimatedPosition ,animatedPadding,

[2-2] custom implicit animation => يستخدم عند عدم وجود AnimatedFOO ونستخدم معه tween

[2-1] built in implicit animation => بيعتمد على ويدجت جاهزه وبيكون سهل في المعامله زي AnimatedContainer

Example on Implicit animation :-
used :-
1]AnimatedContainer()
2]AnimatedOpecity()
4]Curve   بيحدد نوع الحركه سواء كانت خطيه او كيرف معين له معادله
5]Duration()   بيحدد المدة الزمنيه التي يعمل حلالها الانميشن
6] Random() ==> عبباره عن كلاس بيولد ارقام عشوائيه

Built-in

```
import 'package:flutter/animation.dart';
import 'package:flutter/material.dart';
import 'dart:math';
class ImplicitAnimationPage extends StatefulWidget {
  @override
  _ImplicitAnimationPageState createState() => _ImplicitAnimationPageState();
}
class _ImplicitAnimationPageState extends State<ImplicitAnimationPage> {
  //////       *****          ////////
  bool visible = true;
  double _height =30.0;
  double _width  =30.0;
  Color _color =Color(0xff123697);
  Random _random = Random();
  /////// *********           ************///
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Implicit animation"),centerTitle: true,),
      body: Container(
        alignment: Alignment.center,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            /////           AnimatedContainer      ////////////
            AnimatedContainer(
              duration: Duration(seconds: 2),
                child:Center(child: Text("Go"),
              ),
              curve: Curves.easeInOutCirc,
              color: _color,
              height: _height,

              width:_width,
            ),
            //////////////////      AnimatedOpacity         /////////
            // An opacity of 1.0 is fully opaque. An opacity
            // of 0.0 is fully transparent
            AnimatedOpacity(
                opacity: visible==true?1:0, duration: Duration(seconds: 2),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                children: [
                  Text("Mahney Mohsen Mohamed Ahmed"),
                  Text("I am engineer"),
                ],
              ),
            ),
            ///   FloatingActionButton
            FloatingActionButton(
              child: Icon(Icons.flip),
              onPressed: (){
                setState(() {
                  visible =!visible;
                  _height =_random.nextInt(200).toDouble() ;
                  _width=_random.nextInt(200).toDouble() ;
                  _color=Color.fromRGBO(Random().nextInt(256),
                  _random.nextInt(256), _random.nextInt(256), visible?0:1) ;
                });},
                ),
          ],
        ),),);}}
```

Example on built in implicit animation
AnimatedDefaultTextStyle => بيضف نفس ال ستيل لاي نص اسفله

```
AnimatedDefaultTextStyle(
  style: TextStyle(color: _color ,fontSize: _fontSize),
  duration: Duration(seconds: 4),
  curve:Curves.bounceInOut,
  child: Column(
  children: [
    Text("Mahney Mohsen"), Text("elbana"),
    Row(children: [
      Text("I am "), Text("engineer"),],),],),
),
/////   TextButton      ////
TextButton(onPressed: (){
  setState(() {
  _color = Color.fromRGBO(_random.nextInt(256),
      _random.nextInt(256),_random.nextInt(256), 1);
  /// in the range from 0.0  to 1.0.
  _fontSize = _random.nextDouble()*40;
});}, child: Text("change style"),
),
```

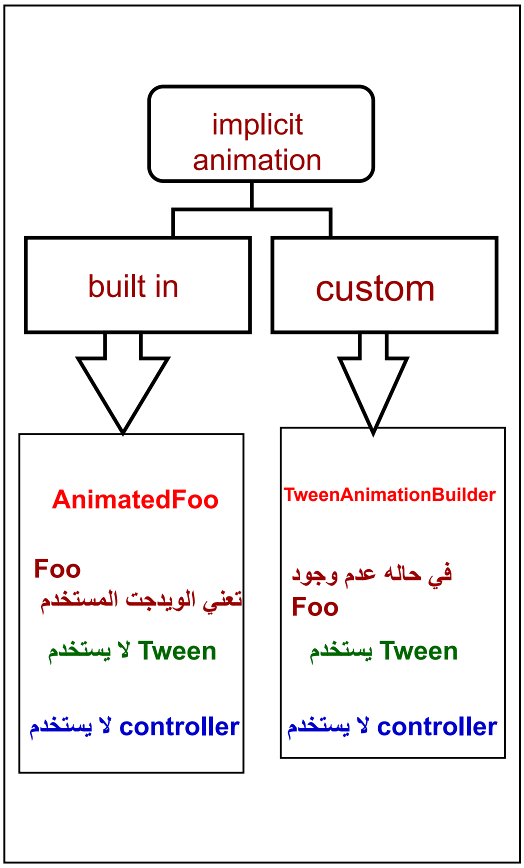
هذة النوع لايستخدم Tween

```
Color _color ;
double _fontSize;
```

Example on Custom implicit animation
== > used Tween == > means between two values

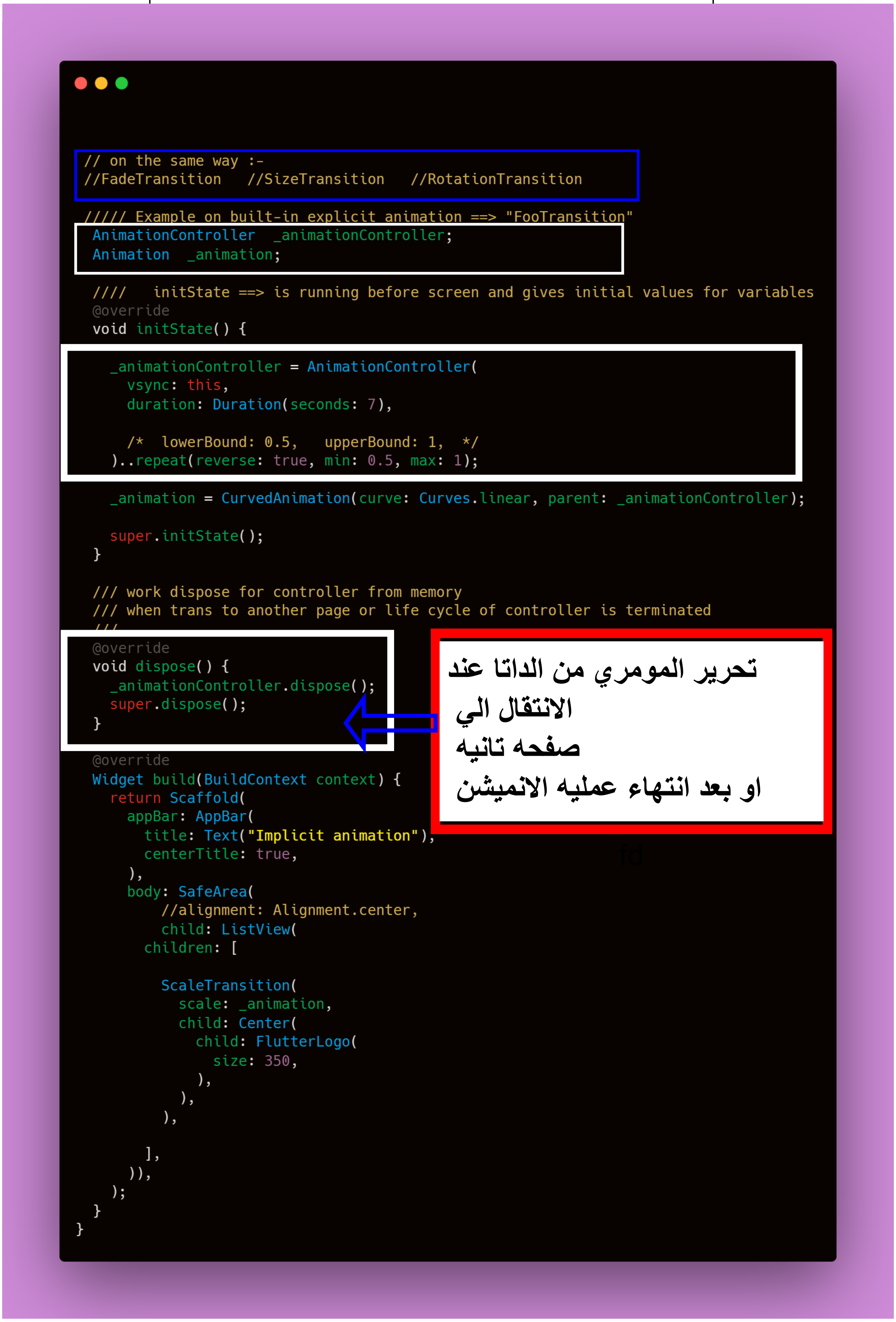
```
///// Example on custom implicit animation =>transform.rotate
     ///* // ==> there are no {AnimatedTransform.rotate}

                TweenAnimationBuilder<double>(
                      //moving is between two values
           tween: Tween<double>(begin: 0.0, end: _angle),
                  duration: Duration(seconds: 3),
                        child: Container(
                       color: Colors.green,
                           height: 100,
                           width: 100,
                               ),
     // angle is taken from Tween.value ==> is a double value
        // we can use Transform.rotate without Builder()
          builder: (context, angle, child) => Builder(
                        builder: (context) {
                       return Transform.rotate(
                              angle: angle,
                              child: child
                                 );
                                },
                               ),
                              ),
                    Text("${(_angle*(180/pi)).round()}"),
                           Slider(
                        value: _angle,
                         max: pi * 2,
                           min: 0,
                        divisions: 4,
                     onChanged: (value) {
                         setState(() {
                         _angle = value;
                             });
                            },
                           ),
         ///////////// Example on custom implicit /////
```

```
double _angle = 0.0;
```



FOO ==> as Container or position or align or padding

Example on :- built in Explicit animation



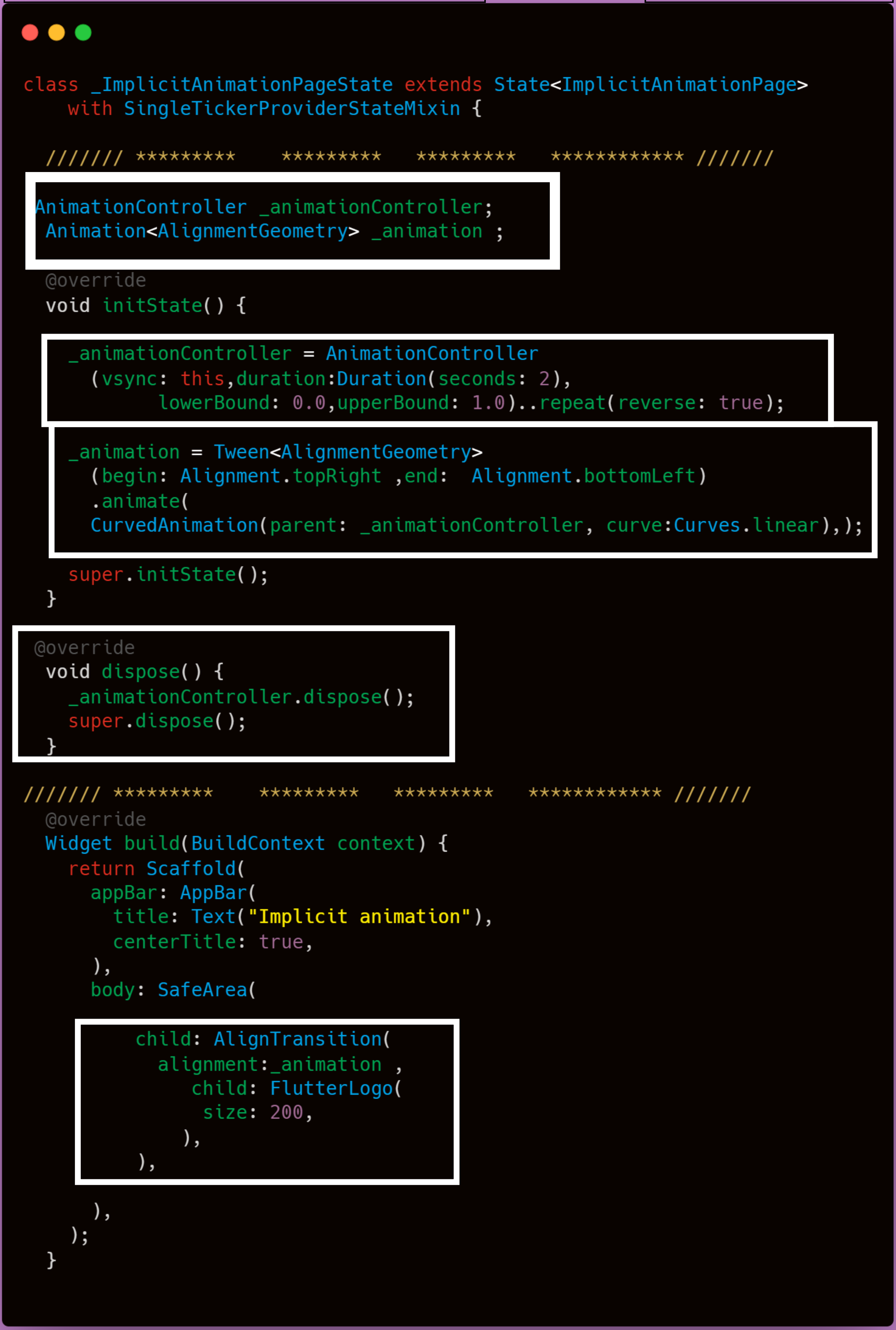
```
// on the same way :-
//FadeTransition   //SizeTransition   //RotationTransition

///// Example on built-in explicit animation ==> "FooTransition"
 AnimationController  _animationController;
 Animation  _animation;

 ////   initState ==> is running before screen and gives initial values for variables
 @override
 void initState() {

   _animationController = AnimationController(
     vsync: this,
     duration: Duration(seconds: 7),

     /*  lowerBound: 0.5,   upperBound: 1,  */
   )..repeat(reverse: true, min: 0.5, max: 1);

   _animation = CurvedAnimation(curve: Curves.linear, parent: _animationController);

   super.initState();
 }

 /// work dispose for controller from memory
 /// when trans to another page or life cycle of controller is terminated
 ///
 @override
 void dispose() {
   _animationController.dispose();
   super.dispose();
 }
 @override
 Widget build(BuildContext context) {
   return Scaffold(
     appBar: AppBar(
       title: Text("Implicit animation"),
       centerTitle: true,
     ),
     body: SafeArea(
         //alignment: Alignment.center,
         child: ListView(
       children: [

         ScaleTransition(
           scale: _animation,
           child: Center(
             child: FlutterLogo(
               size: 350,
             ),
           ),
         ),

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

تحرير المومري من الداتا عند
الانتقال الي
صفحه تانيه
او بعد انتهاء عمليه الانميشن

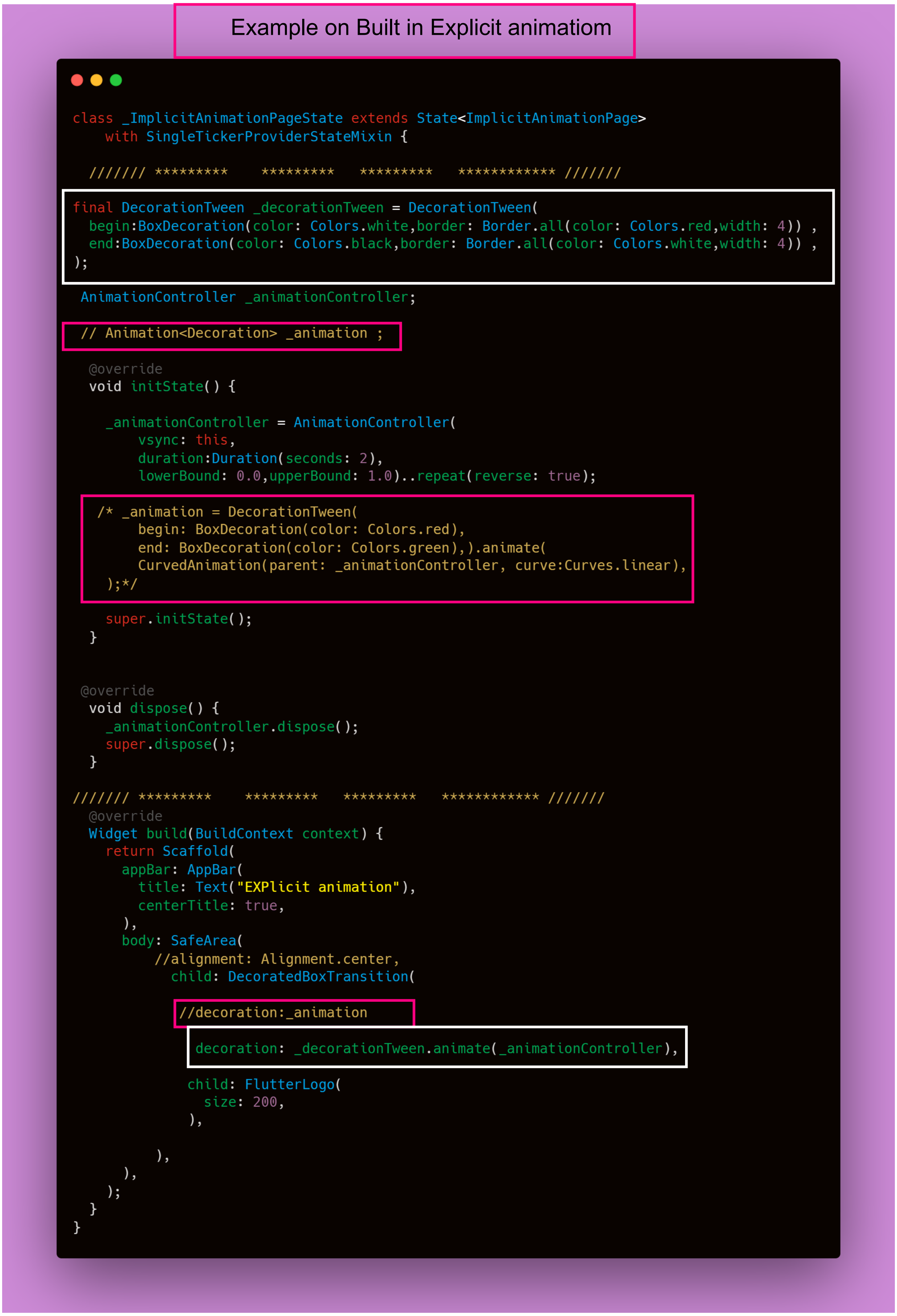
fd

Example on Built-in Explicit animation

using Tween<T>



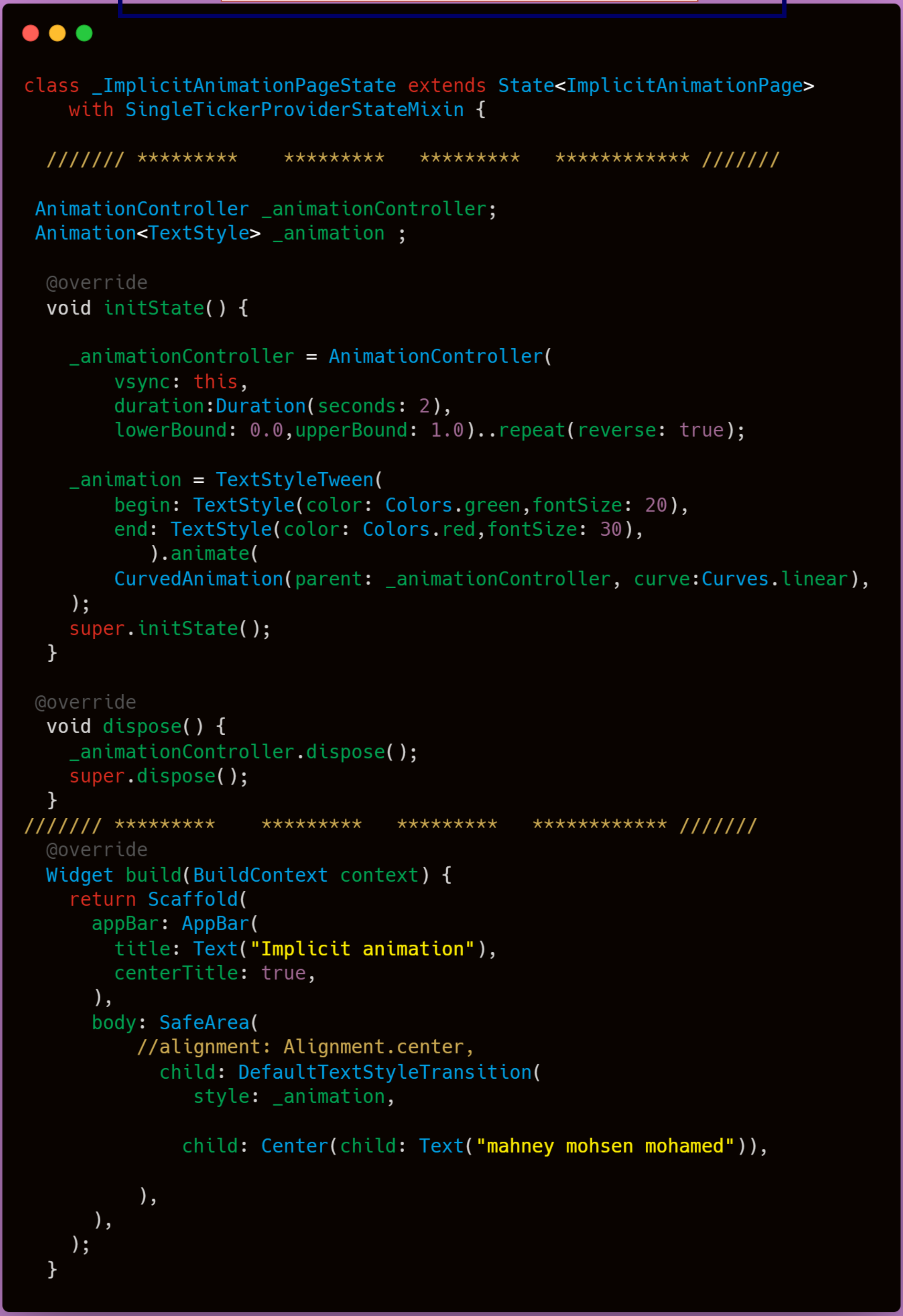
```
class _ImplicitAnimationPageState extends State<ImplicitAnimationPage>
    with SingleTickerProviderStateMixin {

  /////// *********    *********   *********   ************ ///////

 AnimationController _animationController;
  Animation<AlignmentGeometry> _animation ;

  @override
  void initState() {

    _animationController = AnimationController
      (vsync: this,duration:Duration(seconds: 2),
            lowerBound: 0.0,upperBound: 1.0)..repeat(reverse: true);

    _animation = Tween<AlignmentGeometry>
      (begin: Alignment.topRight ,end:  Alignment.bottomLeft)
      .animate(
      CurvedAnimation(parent: _animationController, curve:Curves.linear),);

    super.initState();
  }

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

/////// *********    *********   *********   ************ ///////
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Implicit animation"),
        centerTitle: true,
      ),
      body: SafeArea(

          child: AlignTransition(
            alignment:_animation ,
               child: FlutterLogo(
                size: 200,
              ),
          ),

      ),
    );
  }
```



Example on Built in Explicit animatiom

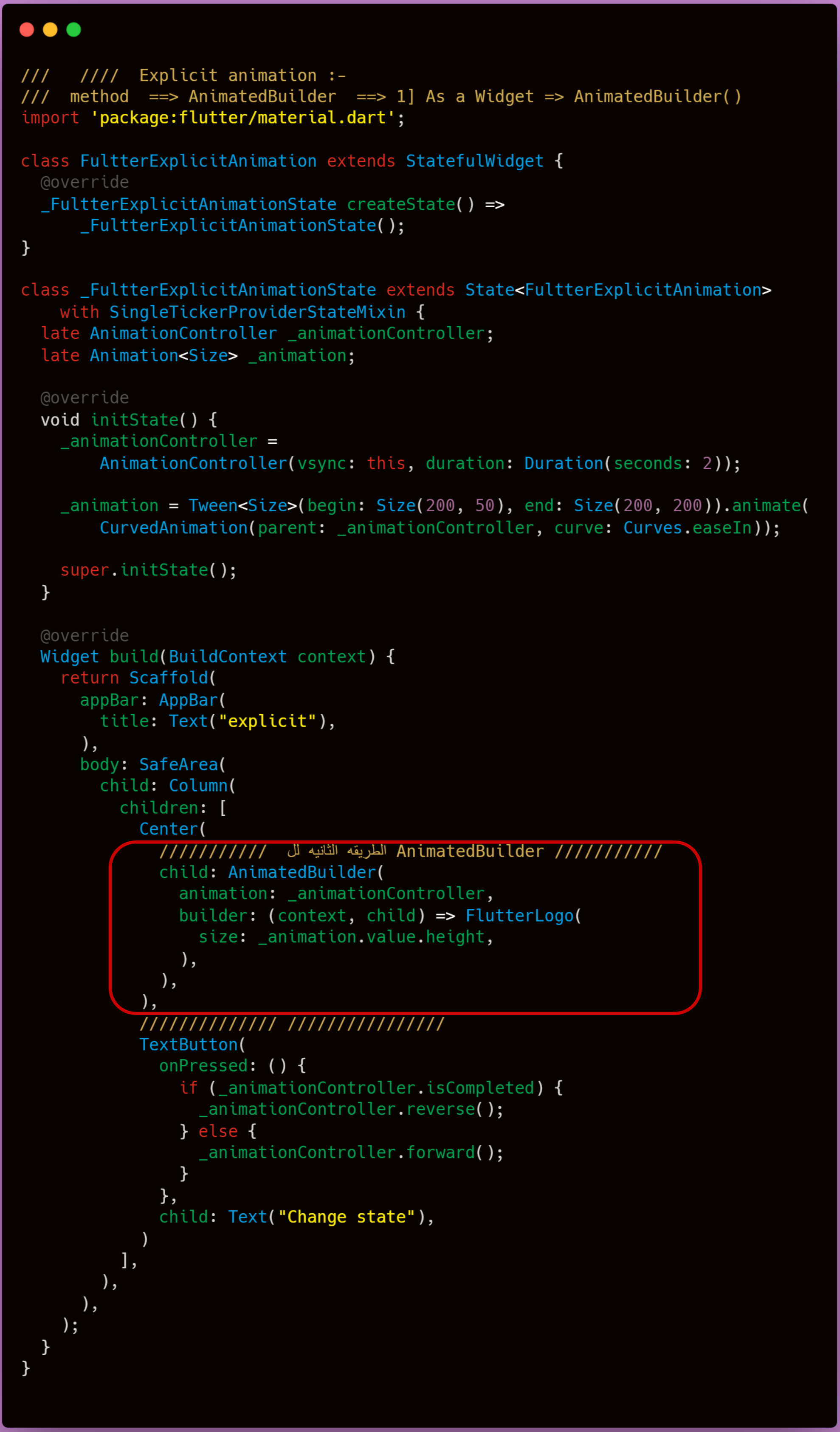
```
class _ImplicitAnimationPageState extends State<ImplicitAnimationPage>
  with SingleTickerProviderStateMixin {

 /////// *********    *********   *********   ************ ///////

final DecorationTween _decorationTween = DecorationTween(
 begin:BoxDecoration(color: Colors.white,border: Border.all(color: Colors.red,width: 4)) ,
 end:BoxDecoration(color: Colors.black,border: Border.all(color: Colors.white,width: 4)) ,
);
 AnimationController _animationController;

 // Animation<Decoration> _animation ;

 @override
 void initState() {

  _animationController = AnimationController(
    vsync: this,
    duration:Duration(seconds: 2),
    lowerBound: 0.0,upperBound: 1.0)..repeat(reverse: true);

  /* _animation = DecorationTween(
    begin: BoxDecoration(color: Colors.red),
    end: BoxDecoration(color: Colors.green),).animate(
    CurvedAnimation(parent: _animationController, curve:Curves.linear),
  );*/

  super.initState();
 }


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

/////// *********    *********   *********   ************ ///////
 @override
 Widget build(BuildContext context) {
  return Scaffold(
   appBar: AppBar(
    title: Text("EXPlicit animation"),
    centerTitle: true,
   ),
   body: SafeArea(
     //alignment: Alignment.center,
      child: DecoratedBoxTransition(

       //decoration:_animation

        decoration: _decorationTween.animate(_animationController),

       child: FlutterLogo(
        size: 200,
       ),

     ),
   ),
  );
 }
}
```

Example on built in explicit animation



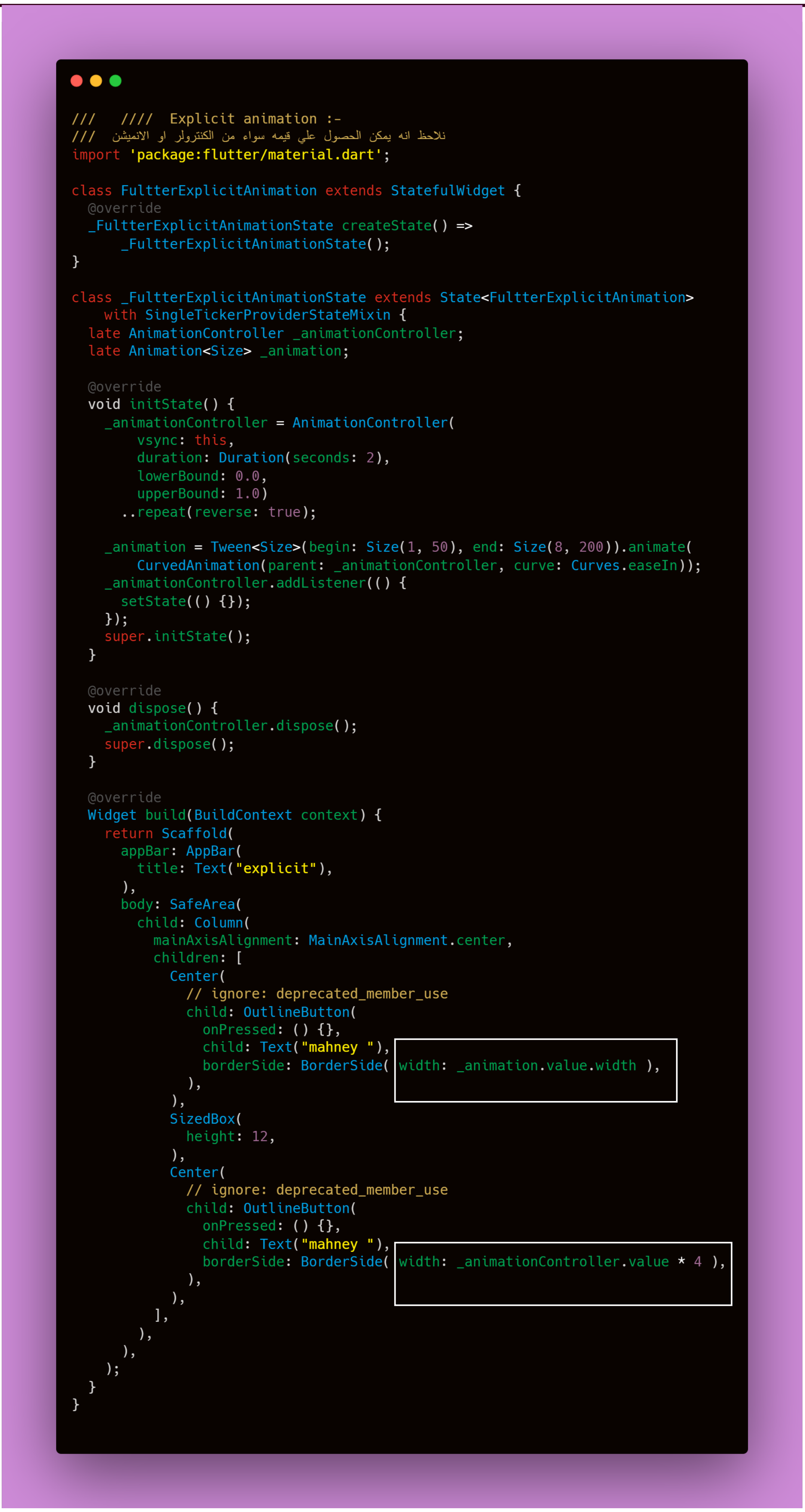
```dart
class _ImplicitAnimationPageState extends State<ImplicitAnimationPage>
    with SingleTickerProviderStateMixin {

  /////// *********    *********    *********    ************ ///////

 AnimationController _animationController;
 Animation<TextStyle> _animation ;

  @override
  void initState() {

    _animationController = AnimationController(
        vsync: this,
        duration:Duration(seconds: 2),
        lowerBound: 0.0,upperBound: 1.0)..repeat(reverse: true);

    _animation = TextStyleTween(
        begin: TextStyle(color: Colors.green,fontSize: 20),
        end: TextStyle(color: Colors.red,fontSize: 30),
           ).animate(
        CurvedAnimation(parent: _animationController, curve:Curves.linear),
    );
    super.initState();
  }

 @override
  void dispose() {
    _animationController.dispose();
    super.dispose();
  }
/////// *********    *********    *********    ************ ///////
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Implicit animation"),
        centerTitle: true,
      ),
      body: SafeArea(
          //alignment: Alignment.center,
            child: DefaultTextStyleTransition(
               style: _animation,

              child: Center(child: Text("mahney mohsen mohamed")),

          ),
      ),
    );
  }
```

Explicit animation :- method ==> AnimatedBuilder ==> 1] As a Widget => **AnimatedBuilder()**



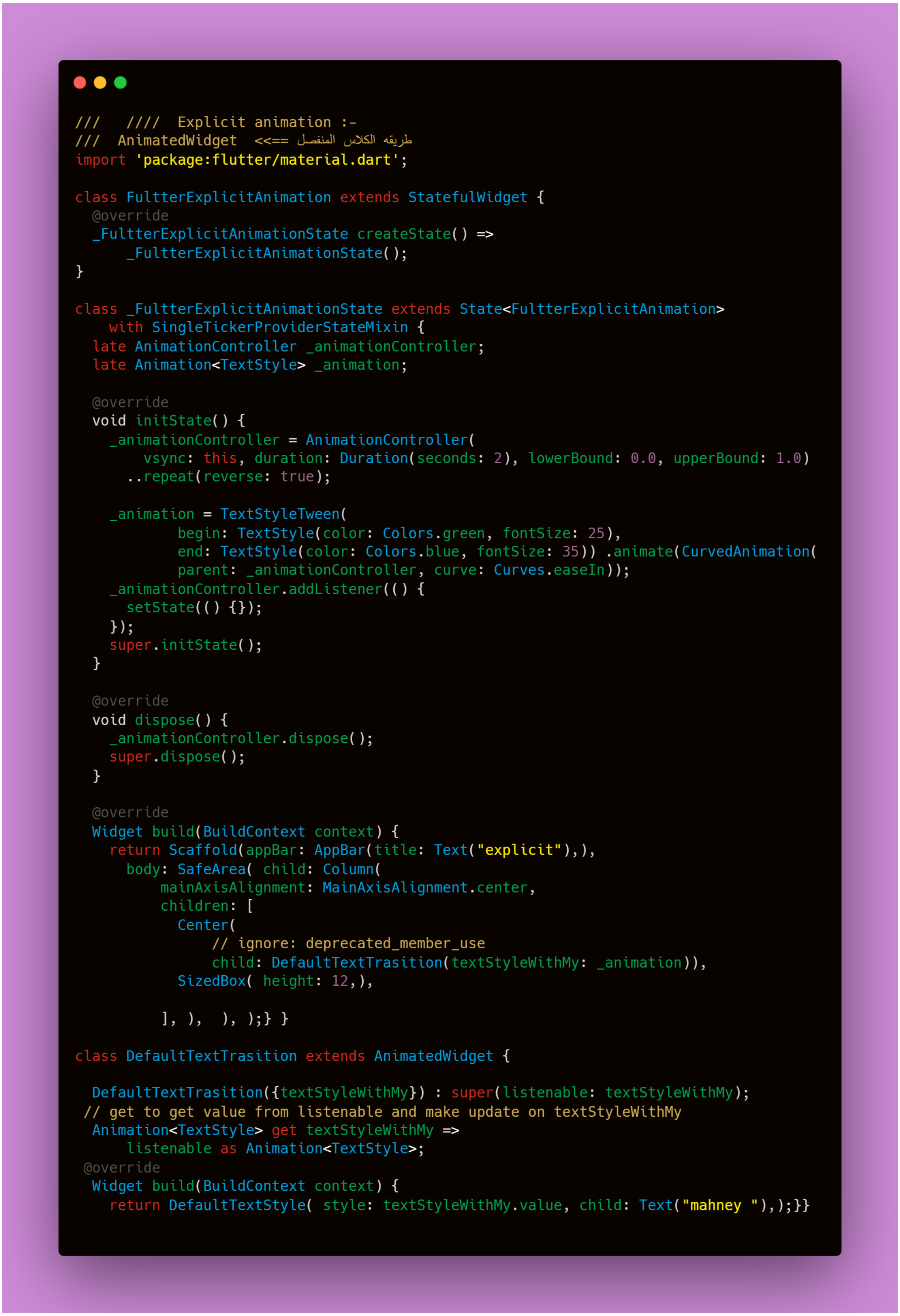
```dart
///   ////  Explicit animation :-
///  method  ==> AnimatedBuilder  ==> 1] As a Widget => AnimatedBuilder()
import 'package:flutter/material.dart';

class FultterExplicitAnimation extends StatefulWidget {
  @override
  _FultterExplicitAnimationState createState() =>
      _FultterExplicitAnimationState();
}

class _FultterExplicitAnimationState extends State<FultterExplicitAnimation>
    with SingleTickerProviderStateMixin {
  late AnimationController _animationController;
  late Animation<Size> _animation;

  @override
  void initState() {
    _animationController =
        AnimationController(vsync: this, duration: Duration(seconds: 2));

    _animation = Tween<Size>(begin: Size(200, 50), end: Size(200, 200)).animate(
        CurvedAnimation(parent: _animationController, curve: Curves.easeIn));

    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("explicit"),
      ),
      body: SafeArea(
        child: Column(
          children: [
            Center(
              ///////////  الطريقه الثانيه لل AnimatedBuilder ///////////
              child: AnimatedBuilder(
                animation: _animationController,
                builder: (context, child) => FlutterLogo(
                  size: _animation.value.height,
                ),
              ),
            ),
            ////////////// ///////////////
            TextButton(
              onPressed: () {
                if (_animationController.isCompleted) {
                  _animationController.reverse();
                } else {
                  _animationController.forward();
                }
              },
              child: Text("Change state"),
            )
          ],
        ),
      ),
    );
  }
}
```

لاحظ انه ممكن نستخدم الكنترولر بدون الانميشن الان كلا منهم
يعطي قيمه مع الملاحظه انه لايتم استخدام الانميشن بدون كنترولر



```dart
///   ////  Explicit animation :-
///  نلاحظ انه يمكن الحصول علي قيمه سواء من الكنترولر او الانميشن
import 'package:flutter/material.dart';

class FultterExplicitAnimation extends StatefulWidget {
  @override
  _FultterExplicitAnimationState createState() =>
      _FultterExplicitAnimationState();
}

class _FultterExplicitAnimationState extends State<FultterExplicitAnimation>
    with SingleTickerProviderStateMixin {
  late AnimationController _animationController;
  late Animation<Size> _animation;

  @override
  void initState() {
    _animationController = AnimationController(
        vsync: this,
        duration: Duration(seconds: 2),
        lowerBound: 0.0,
        upperBound: 1.0)
      ..repeat(reverse: true);

    _animation = Tween<Size>(begin: Size(1, 50), end: Size(8, 200)).animate(
        CurvedAnimation(parent: _animationController, curve: Curves.easeIn));
    _animationController.addListener(() {
      setState(() {});
    });
    super.initState();
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("explicit"),
      ),
      body: SafeArea(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Center(
              // ignore: deprecated_member_use
              child: OutlineButton(
                onPressed: () {},
                child: Text("mahney "),
                borderSide: BorderSide( width: _animation.value.width ),
              ),
            ),
            SizedBox(
              height: 12,
            ),
            Center(
              // ignore: deprecated_member_use
              child: OutlineButton(
                onPressed: () {},
                child: Text("mahney "),
                borderSide: BorderSide( width: _animationController.value * 4 ),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

اخر طريقه في ال اكسبلست انميشن و هي|
طريقه الانميشن ببناء كلاس منفرد



```
///   ////  Explicit animation :-
///  AnimatedWidget  <<== طريقه الكلاس المنفصل
import 'package:flutter/material.dart';

class FultterExplicitAnimation extends StatefulWidget {
  @override
  _FultterExplicitAnimationState createState() =>
      _FultterExplicitAnimationState();
}

class _FultterExplicitAnimationState extends State<FultterExplicitAnimation>
    with SingleTickerProviderStateMixin {
  late AnimationController _animationController;
  late Animation<TextStyle> _animation;

  @override
  void initState() {
    _animationController = AnimationController(
        vsync: this, duration: Duration(seconds: 2), lowerBound: 0.0, upperBound: 1.0)
      ..repeat(reverse: true);

    _animation = TextStyleTween(
            begin: TextStyle(color: Colors.green, fontSize: 25),
            end: TextStyle(color: Colors.blue, fontSize: 35)) .animate(CurvedAnimation(
            parent: _animationController, curve: Curves.easeIn));
    _animationController.addListener(() {
      setState(() {});
    });
    super.initState();
  }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(appBar: AppBar(title: Text("explicit"),),
      body: SafeArea( child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Center(
                // ignore: deprecated_member_use
                child: DefaultTextTrasition(textStyleWithMy: _animation)),
            SizedBox( height: 12,),

          ], ),  ), );} }

class DefaultTextTrasition extends AnimatedWidget {

  DefaultTextTrasition({textStyleWithMy}) : super(listenable: textStyleWithMy);
 // get to get value from listenable and make update on textStyleWithMy
  Animation<TextStyle> get textStyleWithMy =>
      listenable as Animation<TextStyle>;
 @override
  Widget build(BuildContext context) {
    return DefaultTextStyle( style: textStyleWithMy.value, child: Text("mahney "),);}}
```