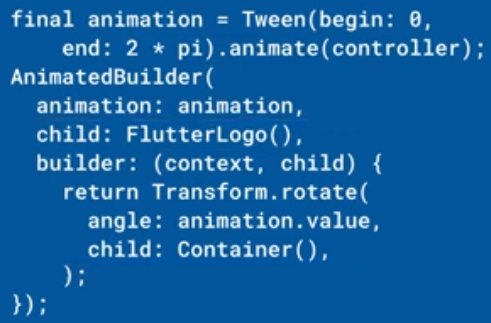
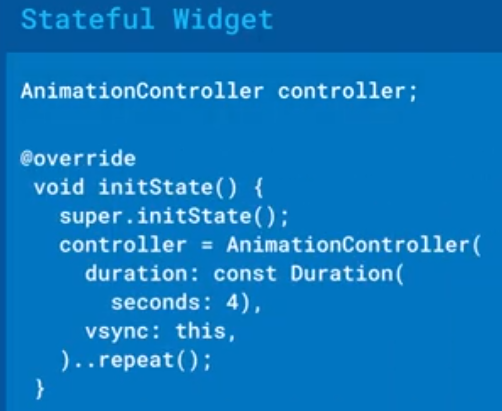
**Animated Builder**

* Definimos la animación (animation con un Tween)
* En el builder ponemos el widget a animar
* En child se puede poner un widget opcional
* El controlador se inicializa en el constructor de un StatefulWidget

|  |  |
| --- | --- |
| **Código** | **Comentarios** |
| import 'dart:math';  import 'package:flutter/material.dart';  class AnimatedBuilderScreen extends StatelessWidget {    const AnimatedBuilderScreen({Key? key}) : super(key: key);    @override    Widget build(BuildContext context) {      return const \_MyHome();    }  }  //----------------------------------------------------------------------  class \_MyHome extends StatefulWidget {    const \_MyHome({      Key? key,    }) : super(key: key);    @override    State<\_MyHome> createState() => \_MyHomeState();  }  class \_MyHomeState extends State<\_MyHome> with SingleTickerProviderStateMixin {    late final AnimationController controller;    late final Animation<double> movingTop;    late final Animation<double> rotation;    late final Animation<double> scale;    @override    void initState() {      super.initState();      controller = AnimationController(          vsync: this, duration: const Duration(milliseconds: 2000));      movingTop = Tween<double>(begin: 0.0, end: 100.0)          .animate(CurvedAnimation(parent: controller, curve: Curves.bounceOut));      rotation = Tween<double>(begin: 0.0, end: 2 \* pi)          .animate(CurvedAnimation(parent: controller, curve: Curves.bounceOut));      scale = Tween<double>(begin: 1.0, end: 2.0)          .animate(CurvedAnimation(parent: controller, curve: Curves.bounceOut));    }    @override    Widget build(BuildContext context) {      return Scaffold(          appBar: AppBar(            title: const Text('AnimatedBuilder'),            centerTitle: true,          ),          body: Center(            child: AnimatedBuilder(              animation: controller,              child: Rectangulo(),              builder: (BuildContext context, Widget? child) {                return Transform.scale(                  scale: scale.value,                  child: Transform.rotate(                    angle: rotation.value,                    child: Transform.translate(                        offset: Offset(movingTop.value \* 0.1, movingTop.value),                        child: child),                  ),                );              },            ),          ),          floatingActionButton: FloatingActionButton(            onPressed: () {              controller.reset();              controller.forward();            },            child: const Icon(Icons.play\_arrow),          ));    }  }  //----------------------------------------------------------------------  class Rectangulo extends StatelessWidget {    @override    Widget build(BuildContext context) {      return Container(        width: 50,        height: 50,        decoration: const BoxDecoration(          gradient: LinearGradient(            colors: [              Color.fromARGB(255, 5, 1, 128),              Color.fromARGB(255, 23, 109, 221)            ],          ),        ),        child: const Align(          alignment: Alignment.center,          child: Text("T",              style: TextStyle(                  color: Colors.white,                  fontSize: 28,                  fontWeight: FontWeight.bold)),        ),      );    }  } | Mezclar con SingleTickerProviderStateMixin  Definir el controller  Definir el las animaciones  Inicializar controller  Inicializar animaciones  AnimatedBuilder  Lleva el controller  Uso de las animaciones  Acciones sobre el controller |

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -