**1.AnimatedWidget**

AnimatedWidget là một widget trong Flutter dùng để tạo hiệu ứng hoạt hình cho các thành phần giao diện của ứng dụng. Nó là một lớp con của StatefulWidget và thường được sử dụng kết hợp với các lớp khác như Animation và Tween để tạo ra các hiệu ứng di chuyển, thay đổi kích thước, màu sắc, v.v.

Cấu trúc của một AnimatedWidget bao gồm:

+ Một Animation được cung cấp thông qua constructor.

+ Một phương thức builder được triển khai để xác định trạng thái widget dựa trên giá trị hiện tại của Animation.

**-Ví dụ minh họa:**

import 'package:flutter/material.dart';

void main() => runApp(MaterialApp(

home: Scaffold(

appBar: AppBar(title: Text('AnimatedWidget Example')),

body: ColorChangeDemo(),

),

));

class ColorChangeDemo extends StatefulWidget {

@override

\_ColorChangeDemoState createState() => \_ColorChangeDemoState();

}

class \_ColorChangeDemoState extends State<ColorChangeDemo>

with SingleTickerProviderStateMixin {

AnimationController \_controller;

Animation<Color> \_animation;

@override

void initState() {

super.initState();

\_controller = AnimationController(

duration: const Duration(seconds: 2),

vsync: this,

);

\_animation = ColorTween(begin: Colors.blue, end: Colors.red)

.animate(CurvedAnimation(parent: \_controller, curve: Curves.easeInOut));

\_controller.repeat(reverse: true);

}

@override

void dispose() {

\_controller.dispose();

super.dispose();

}

@override

Widget build(BuildContext context) {

return Center(

child: ColorChangingBox(animation: \_animation),

);

}

}

class ColorChangingBox extends AnimatedWidget {

ColorChangingBox({Key key, Animation<Color> animation})

: super(key: key, listenable: animation);

@override

Widget build(BuildContext context) {

final Animation<Color> animation = listenable;

return Container(

width: 100,

height: 100,

color: animation.value,

);

}

}

Trong ví dụ này, sử dụng AnimatedWidget có tên ColorChangingBox để thay đổi màu sắc của hộp từ màu xanh đến màu đỏ. AnimatedWidget giúp tái sử dụng giá trị Animation \_animation mà không cần phải viết lại mã lệnh cập nhật UI. Khi giá trị animation thay đổi, widget ColorChangingBox sẽ tự động được tái build và cập nhật hiển thị màu sắc mới của hộp.