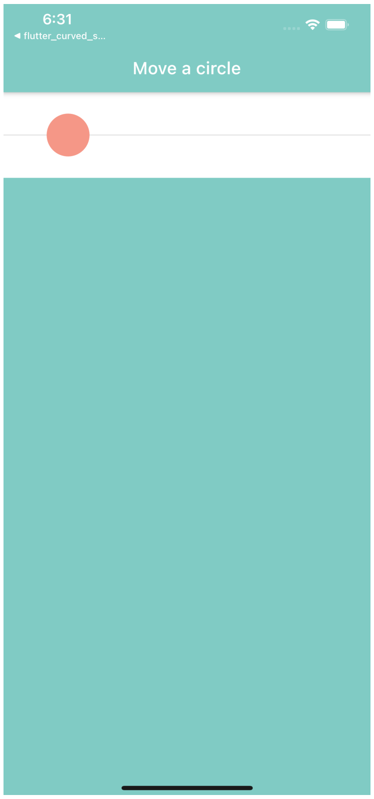
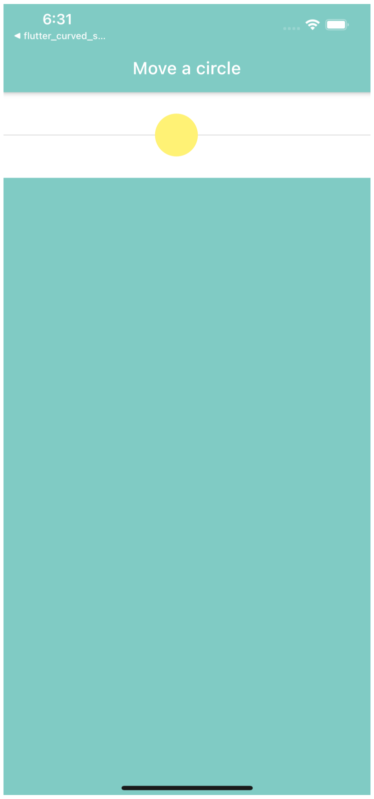
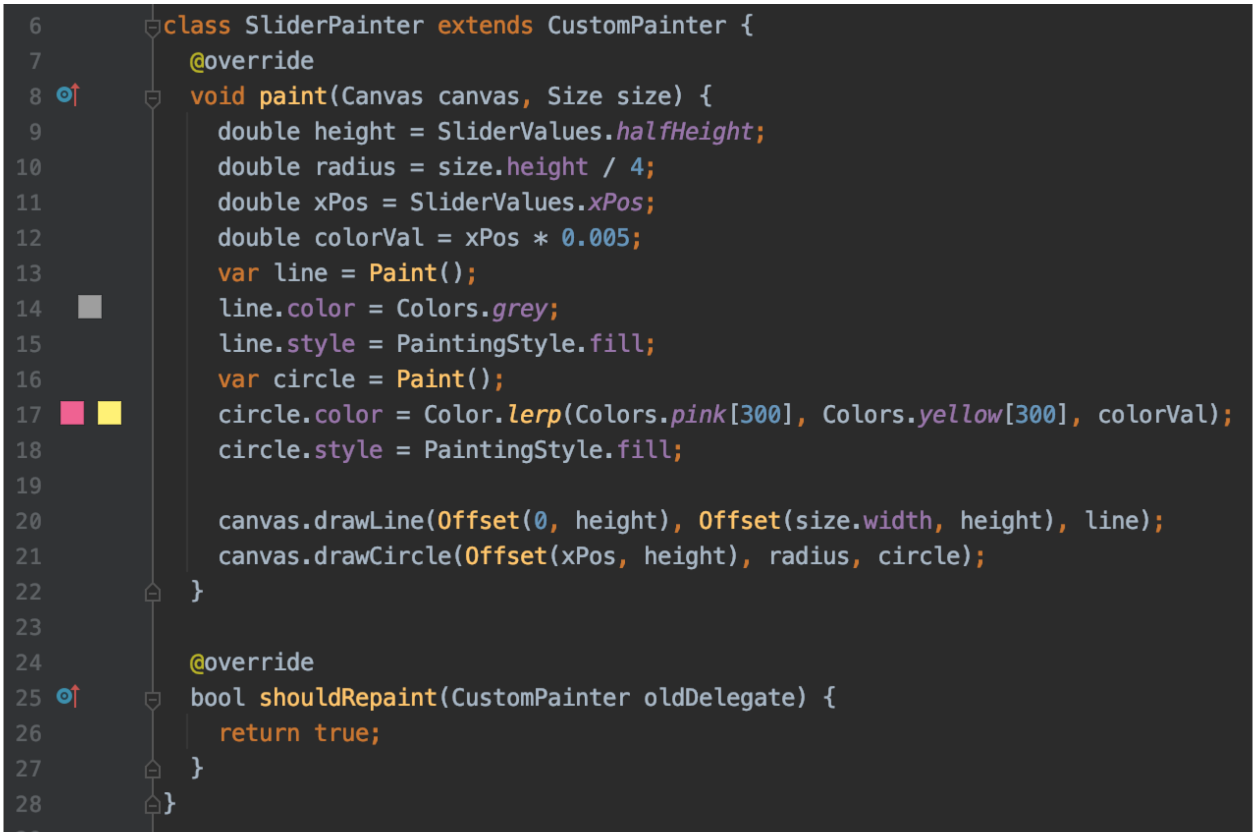


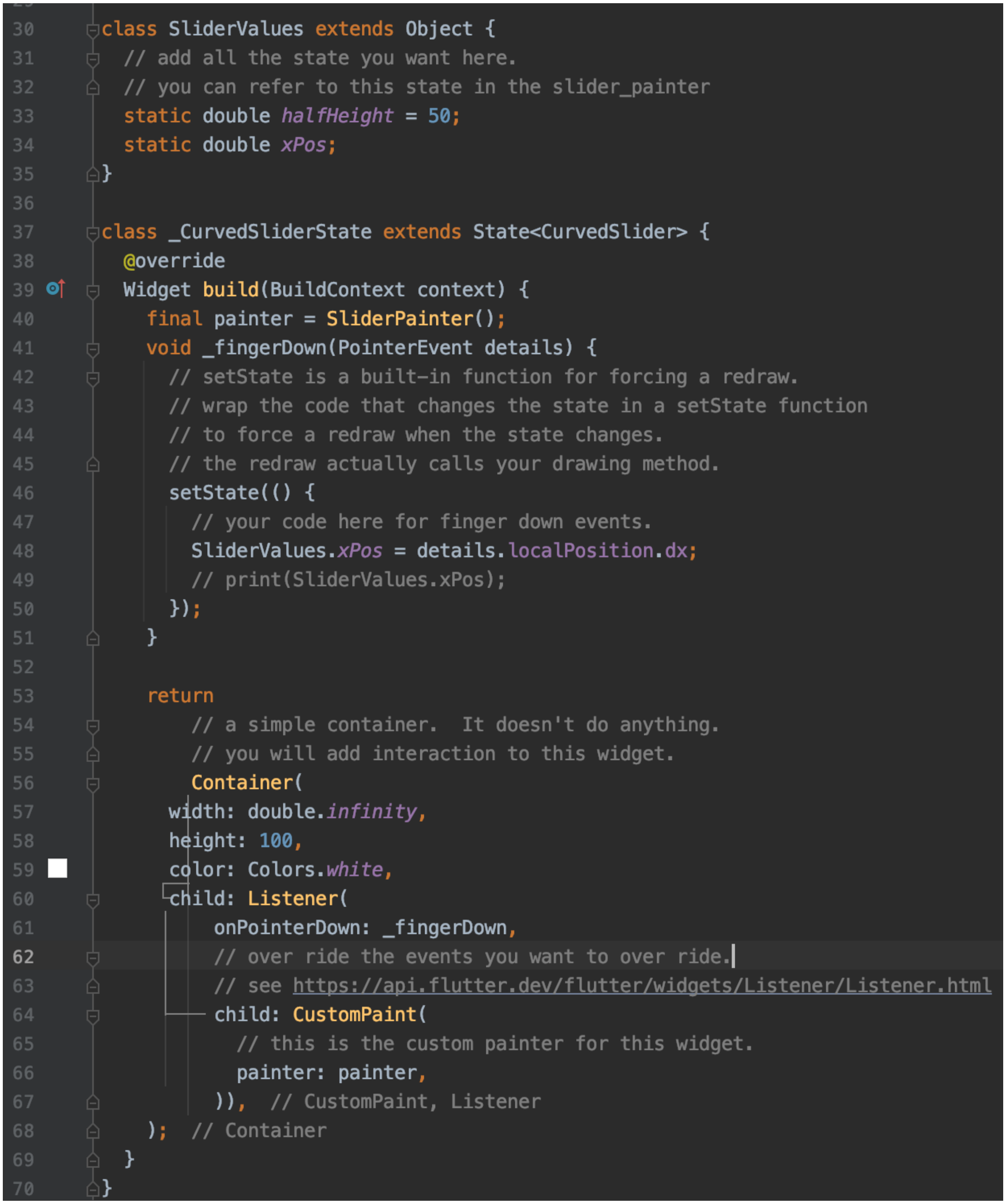




```
class SliderPainter extends CustomPainter {
  @override
  void paint(Canvas canvas, Size size) {
    double height = SliderValues.halfHeight;
    double radius = size.height / 4;
    double xPos = SliderValues.xPos;
    double colorVal = xPos * 0.005;
    var line = Paint();
    line.color = Colors.grey;
    line.style = PaintingStyle.fill;
    var circle = Paint();
    circle.color = Color.lerp(Colors.pink[300], Colors.yellow[300], colorVal);
    circle.style = PaintingStyle.fill;

    canvas.drawLine(Offset(0, height), Offset(size.width, height), line);
    canvas.drawCircle(Offset(xPos, height), radius, circle);
  }

  @override
  bool shouldRepaint(CustomPainter oldDelegate) {
    return true;
  }
}
```



```
class SliderValues extends Object {
  // add all the state you want here.
  // you can refer to this state in the slider_painter
  static double halfHeight = 50;
  static double xPos;
}

class _CurvedSliderState extends State<CurvedSlider> {
  @override
  Widget build(BuildContext context) {
    final painter = SliderPainter();
    void _fingerDown(PointerEvent details) {
      // setState is a built-in function for forcing a redraw.
      // wrap the code that changes the state in a setState function
      // to force a redraw when the state changes.
      // the redraw actually calls your drawing method.
      setState(() {
        // your code here for finger down events.
        SliderValues.xPos = details.localPosition.dx;
        // print(SliderValues.xPos);
      });
    }

    return
        // a simple container.  It doesn't do anything.
        // you will add interaction to this widget.
        Container(
      width: double.infinity,
      height: 100,
      color: Colors.white,
      child: Listener(
          onPointerDown: _fingerDown,
          // over ride the events you want to over ride.
          // see https://api.flutter.dev/flutter/widgets/Listener/Listener.html
          child: CustomPaint(
            // this is the custom painter for this widget.
            painter: painter,
          )),  // CustomPaint, Listener
    );  // Container
  }
}
```