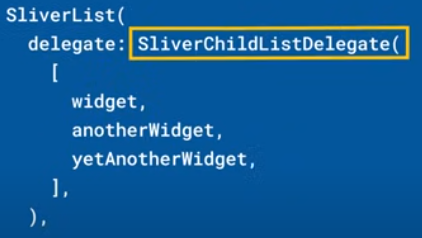
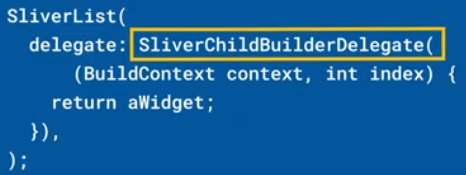
**SilverList y SilverGrid**

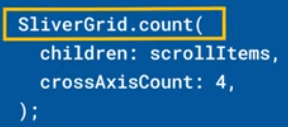
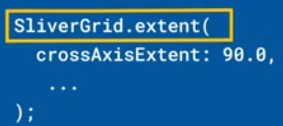
ListView y GridView permiten desplazarse por una lista o una cuadrícula por separado

SliverList y SliverGrid están para desplazar una lista o una cuadrícula juntas

SliverList toma un parámetro delegate que proporciona los elementos:

SliverGrid pueden especificar los elementos secundarios. Tiene dos constructores:

|  |  |
| --- | --- |
| **Código** | **Comentarios** |
| import 'dart:math';  import 'package:flutter/material.dart';  class SliverGridListScreen extends StatelessWidget {    const SliverGridListScreen({Key? key}) : super(key: key);    @override    Widget build(BuildContext context) {      return const \_MyHomeScreen();    }  }  //---------------------------------------------------------------------  class \_MyHomeScreen extends StatelessWidget {    const \_MyHomeScreen({      Key? key,    }) : super(key: key);    @override    Widget build(BuildContext context) {      final List<Widget> items = List.generate(100, (i) => const \_Rectangulo());      return Scaffold(        body: CustomScrollView(          slivers: [            const SliverAppBar(              title: Text("SliverGrid 1"),              centerTitle: true,            ),            SliverGrid.count(              crossAxisCount: 4,              children: items,            ),            const SliverAppBar(              title: Text("SliverGrid 2"),              centerTitle: true,            ),            SliverGrid.extent(              maxCrossAxisExtent: 50,              children: items,            ),            const SliverAppBar(              title: Text("SliverList"),              pinned: true,              centerTitle: true,            ),            SliverList(              delegate: SliverChildBuilderDelegate((context, i) {                return items[i];              }, childCount: items.length),            ),          ],        ),      );    }  }  //---------------------------------------------------------------------  class \_Rectangulo extends StatelessWidget {    const \_Rectangulo({Key? key}) : super(key: key);    @override    Widget build(BuildContext context) {      final rnd = Random();      return Container(        width: 150,        height: 150,        child: Align(          alignment: Alignment.center,          child: Text(generateRandomString(1),              style: TextStyle(                  color: Color.fromRGBO(                    rnd.nextInt(255),                    rnd.nextInt(255),                    rnd.nextInt(255),                    1,                  ),                  fontSize: 28,                  fontWeight: FontWeight.bold)),        ),        decoration: BoxDecoration(          color: Color.fromRGBO(            rnd.nextInt(255),            rnd.nextInt(255),            rnd.nextInt(255),            1,          ),        ),      );    }    String generateRandomString(int len) {      var r = Random();      return String.fromCharCodes(          List.generate(len, (index) => r.nextInt(33) + 65));    }  } |  |

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