

Flutter 2

More on Widgets with Children

Oum Saokosal

Master of Engineering in
Information Systems, South Korea
012 252 752 (Telegram)



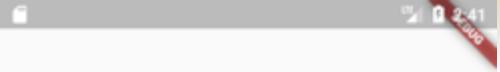


More on Widgets with Children

Row

- Row គឺមានក្នុងប្រើប្រាស់ (List<Widget>) ហើយតាំងរាយជាង្វែរដ៏ក្បាស់

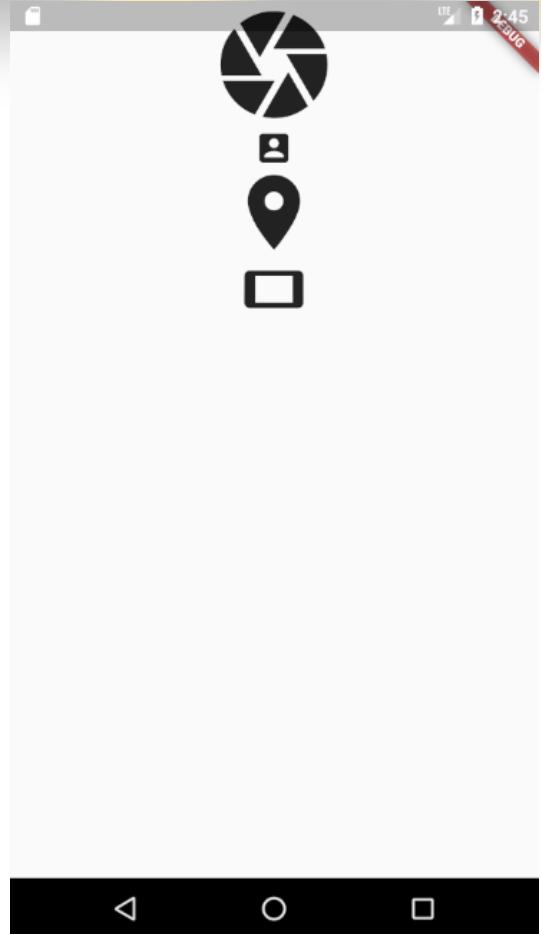
```
Row(  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
) ,
```



Column

- Column គឺមានក្នុងប្រើប្រាស់ (List<Widget>) ដូច row ដើម្បី តែតាំងរួម widget ទាំងនេះជាង្វែរ។

```
Column(  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
) ,
```



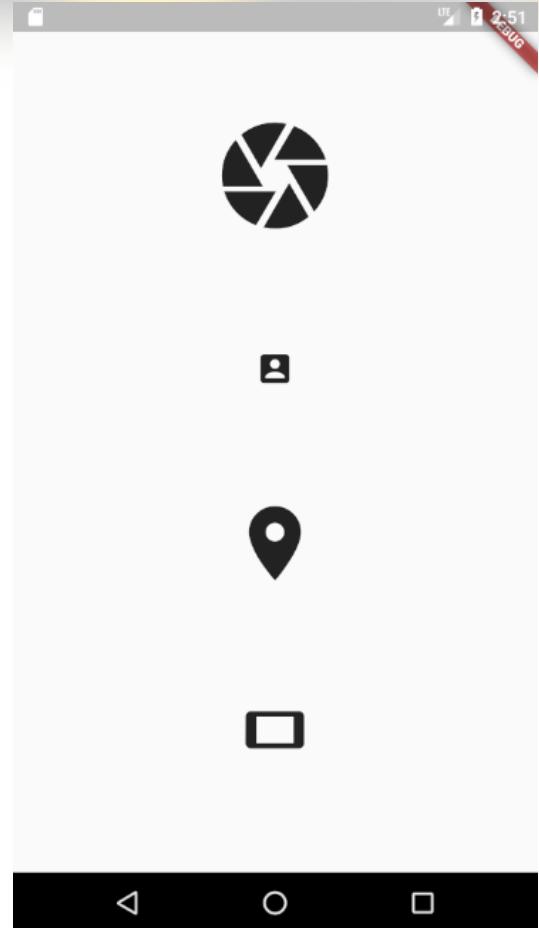
mainAxisAlignment

ទាំង row ទាំង column គឺមាន properties ដូចត្រូវបែនក្នុង

mainAxisAlignment គឺសំរាប់ព័ត៌មាន widget ទាំងអស់ ជាលក្ខណៈ៖

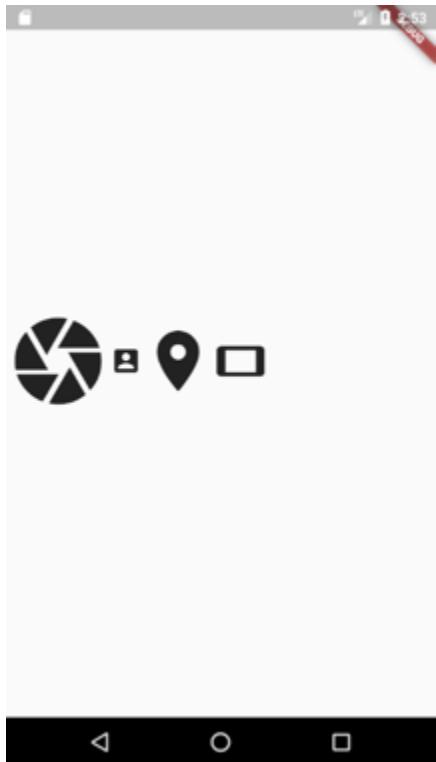
- MainAxisAlignment.start: តាំងរបស់ក្នុងបញ្ចប់ផ្តើម
- MainAxisAlignment.center: តាំងរបស់ក្នុងបញ្ចប់កណ្តាល
- MainAxisAlignment.end: តាំងរបស់ក្នុងបញ្ចប់ពីចិត្ត
- MainAxisAlignment.spaceBetween: តាំងរបស់ក្នុងបញ្ចប់កណ្តាល ដូចត្រូវបានបញ្ជាក់ថ្មី
- MainAxisAlignment.spaceAround: តាំងរបស់ក្នុងបញ្ចប់កណ្តាល ដូចត្រូវបានបញ្ជាក់ថ្មី និងស្រួលបន្ថែម
- MainAxisAlignment.spaceEvenly: តាំងរបស់ក្នុងបញ្ចប់កណ្តាលស្រួលស្រួល

```
Column(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,
```

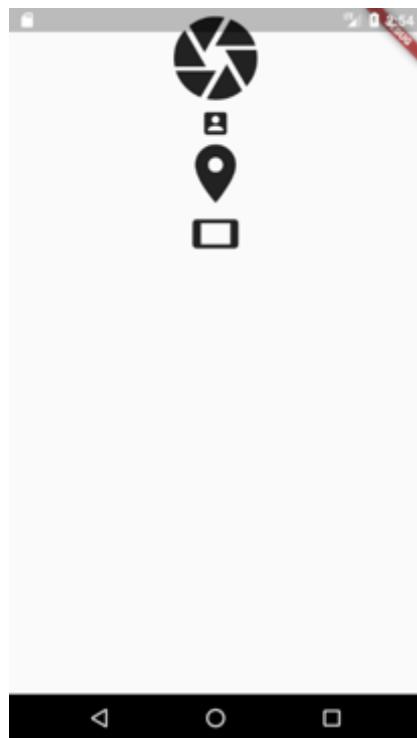


MainAxisAlignment.start: តិច្ឆិទនូចចាប់ផ្តើម

```
Row(  
    mainAxisAlignment: MainAxisAlignment.start,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```



```
Column(  
    mainAxisAlignment: MainAxisAlignment.start,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```

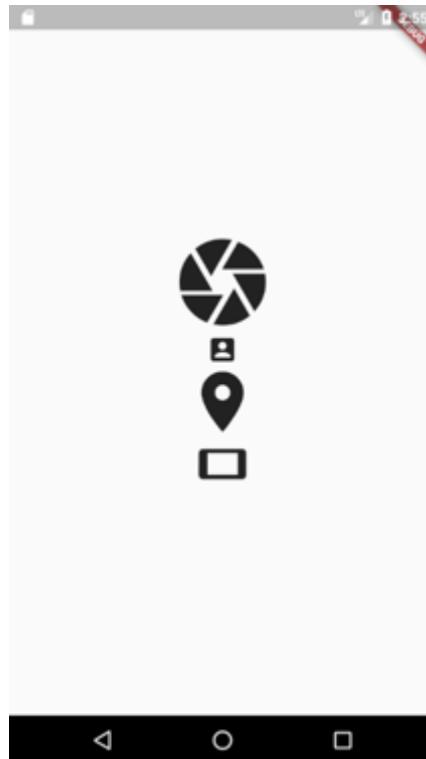


MainAxisAlignment.center: តាំងរបត្រចំណូល

```
Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```



```
Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```

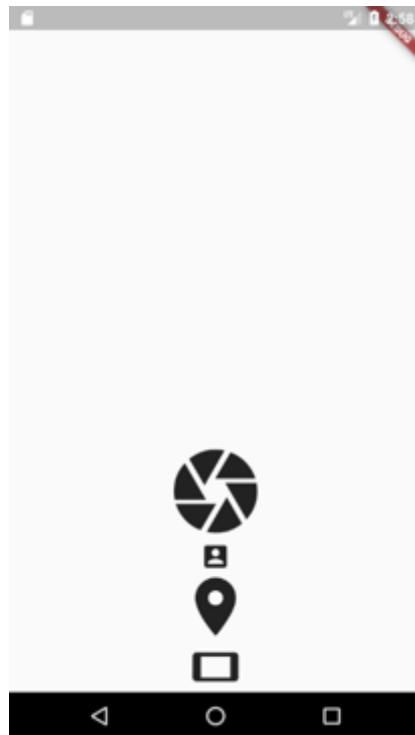


MainAxisAlignment.end: កំណត់បន្ទីចនុចបញ្ចប់

```
Row(  
    mainAxisAlignment: MainAxisAlignment.end,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```

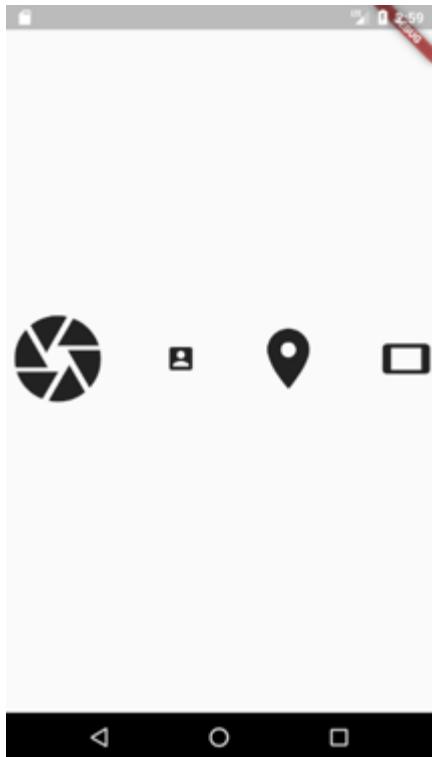


```
Column(  
    mainAxisAlignment: MainAxisAlignment.end,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```

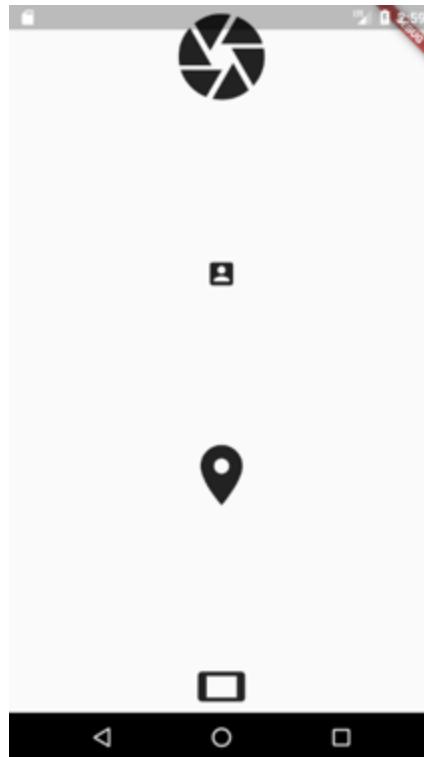


MainAxisAlignment.spaceBetween: តំបន់ដាក់ចន្លះកណ្តាល

```
Row(  
    mainAxisAlignment: MainAxisAlignment.spaceBetween,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```

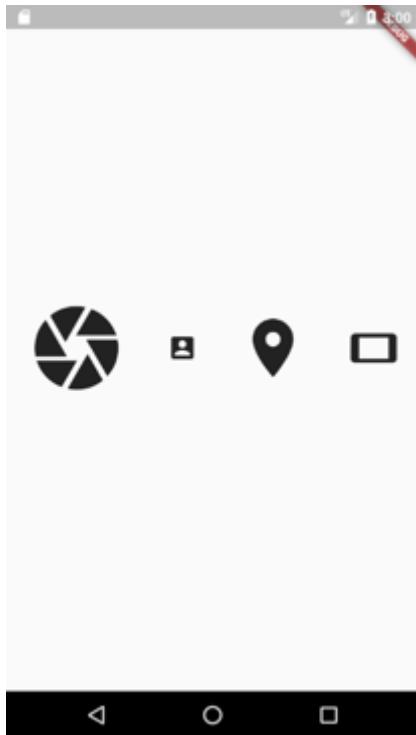


```
Column(  
    mainAxisAlignment: MainAxisAlignment.spaceBetween,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```

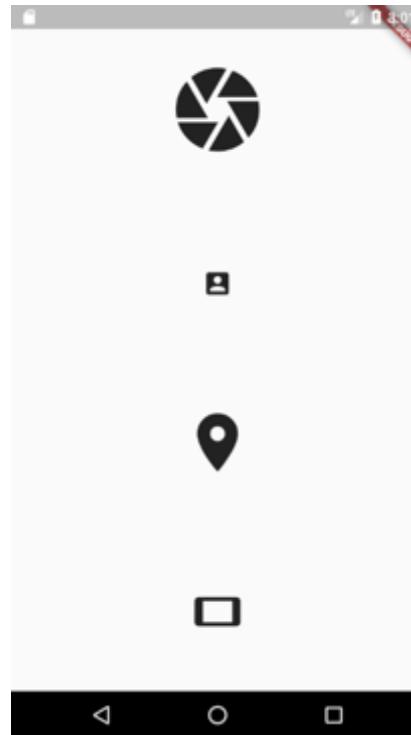


MainAxisAlignment.spaceAround: តិច្ឆេរបង្ហាញកំចនោះដុវីញ

```
Row(  
    mainAxisAlignment: MainAxisAlignment.spaceAround,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```



```
Column(  
    mainAxisAlignment: MainAxisAlignment.spaceAround,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```



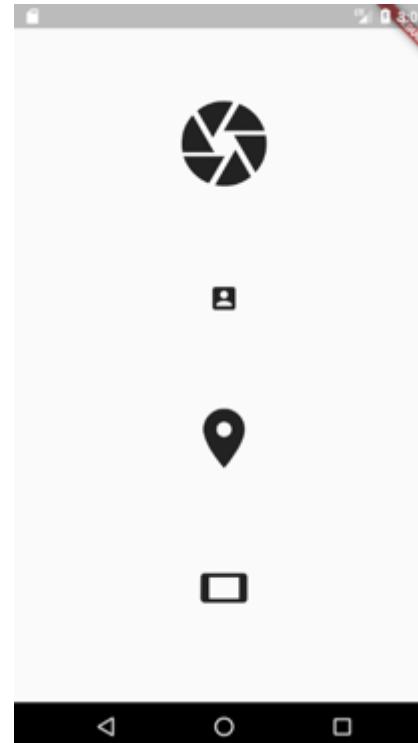
MainAxisAlignment.spaceEvenly:

កំរើបងាកំចន្ទោះកណ្តាលស្រីត្រា

```
Row(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```



```
Column(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,)
```

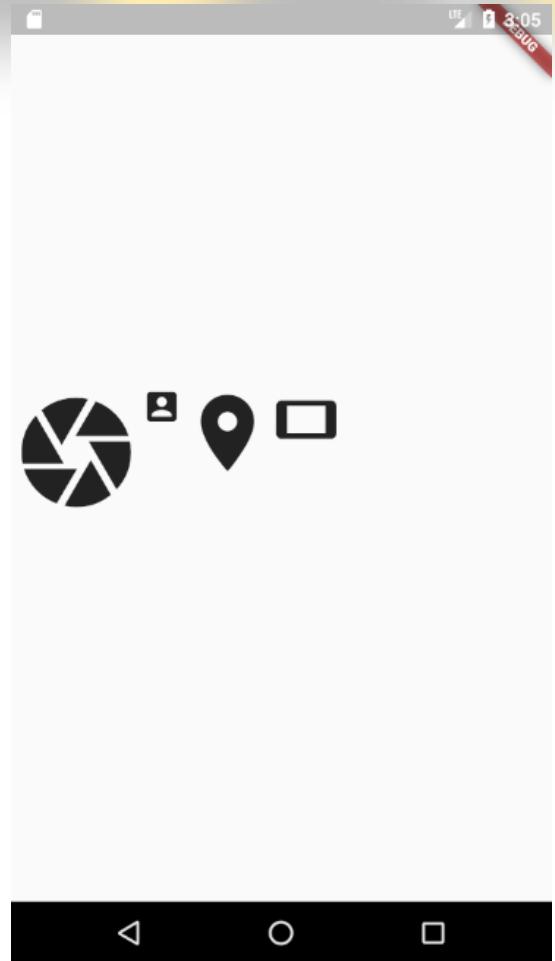


crossAxisAlignment

crossAxisAlignment គឺសំរាប់តាំងរវាង widget និមួយា ជាលក្ខណៈ៖

- CrossAxisAlignment.start: តាំងរវាងចំនួចបាប់ផ្តើមនៃ widget
- CrossAxisAlignment.end: តាំងរវាងចំនួចចុងក្រោយនៃ widget
- CrossAxisAlignment.center: តាំងរវាងចំនួចកណ្តាលនៃ widget
- CrossAxisAlignment.baseline: តាំងរវាងអក្សរនៃចំនួចដើម
 - ត្រូវការ៖ textBaseline: TextBaseline.alphabetic,
- CrossAxisAlignment.stretch: តាំងរវាងទំហំពេញនៃ widget

```
Row(  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
)
```



CrossAxisAlignment.start: តិច្ឆេរវាងចំណុចទាំងពីរនៃ widget

```
Row(  
    mainAxisAlignment: MainAxisAlignment.start,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
) ,
```



CrossAxisAlignment.end: តំរើបរភោងចំណុចបុងក្រាយនៃ widget

```
Row(  
    mainAxisAlignment: MainAxisAlignment.end,  
    children: <Widget>[  
        Icon(Icons.camera, size: 100.0,),  
        Icon(Icons.account_box, size: 30.0,),  
        Icon(Icons.place, size: 70.0,),  
        Icon(Icons.stay_primary_landscape, size: 50.0,),  
    ],  
,),
```



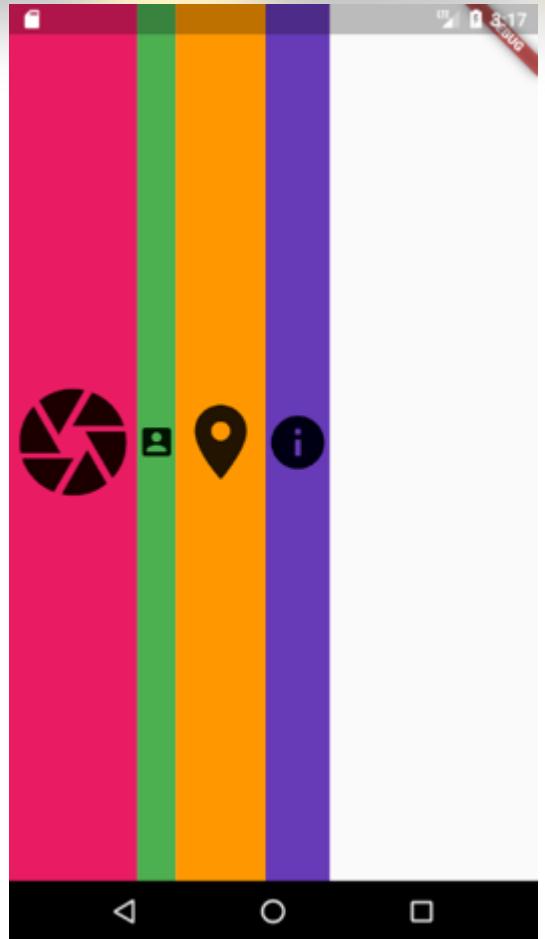
CrossAxisAlignment.baseline: តំរូវបរាជអក្សរនៃទំនុចដើម

```
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  crossAxisAlignment: CrossAxisAlignment.baseline,  
  textBaseline: TextBaseline.alphabetic,  
  children: <Widget>[  
    Icon(Icons.camera, size: 100.0,),  
    Icon(Icons.account_box, size: 30.0,),  
    Icon(Icons.place, size: 70.0,),  
    Icon(Icons.stay_primary_landscape, size: 50.0,),  
  ],  
,
```



CrossAxisAlignment.stretch: កំរើបរភ័ន្ធគំពេញនៃ widget

```
Row(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: <Widget>[  
        Container(  
            color: Colors.pink,  
            child: Icon(Icons.camera, size: 100.0,)),  
        Container(  
            color: Colors.green,  
            child: Icon(Icons.account_box, size: 30.0,)),  
        Container(  
            color: Colors.orange,  
            child: Icon(Icons.place, size: 70.0,)),  
        Container(  
            color: Colors.deepPurple,  
            child: Icon(Icons.info, size: 50.0,)),  
    ],  
) ,
```

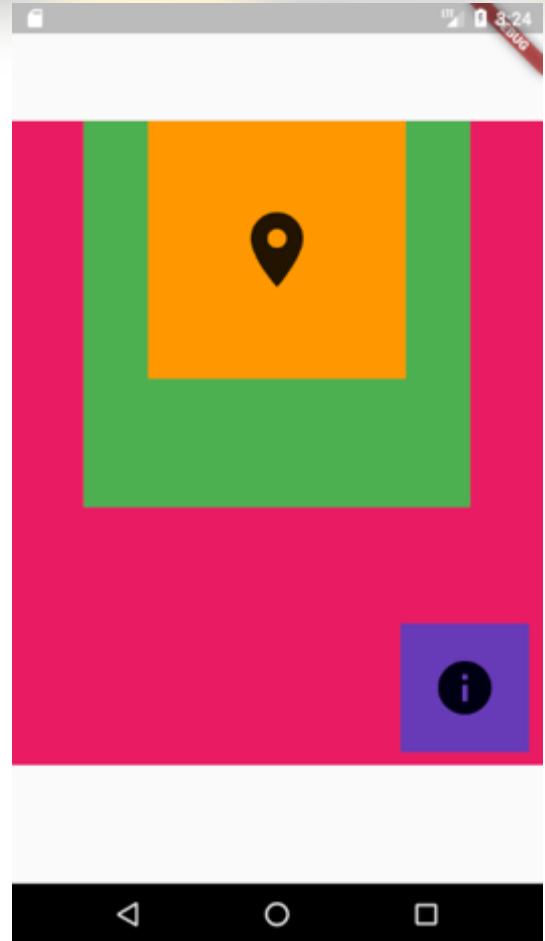


Stack

- Stack គឺសំរាប់តាំង widget គីឡូក្រាម
- យើងអាចប្រើ alignment សំរាប់តាំងគោរពចំណងផ្លូវ ស្ថា លើ ក្រោមនិងកណ្តាលជាដើម។
- បុគ្គលិកប្រើ Positioned សំរាប់កំណត់ទីតាំងគោរពគ្រែកនឹងខាងផ្លូវស្ថា លើ ក្រោម ជាដើម។

Stack

```
    alignment: Alignment.topCenter,
    children: <Widget>[
        Container(width: 500.0, height: 500.0,
            color: Colors.pink,
            child: Icon(Icons.camera, size: 100.0,)),
        Container(width: 300.0, height: 300.0,
            color: Colors.green,
            child: Icon(Icons.account_box, size: 30.0,)),
        Container(width: 200.0, height: 200.0,
            color: Colors.orange,
            child: Icon(Icons.place, size: 70.0,)),
        Positioned(
            right: 10.0,
            bottom: 10.0,
            child: Container(width: 100.0, height: 100.0,
                color: Colors.deepPurple,
                child: Icon(Icons.info, size: 50.0,))),
    ],
)
```



SingleChildScrollView

- SingleChildScrollView គឺជា widget មួយសំរាប់ដំឡើងដល់ការតាំងរៀប Row និង Column ត្រូវបានបញ្ជាមួយគ្រាន់ថា បច្ចេកវិភាសាដឹងថា screen តាមនីងចែង ដូចខាងក្រោម

```
Column(  
    children: <Widget>[  
        Icon(Icons.camera, size: 200,),  
        Icon(Icons.account_box, size: 230.0,),  
        Icon(Icons.place, size: 200.0,),  
        Icon(Icons.info, size: 250.0,),  
    ],  
,
```

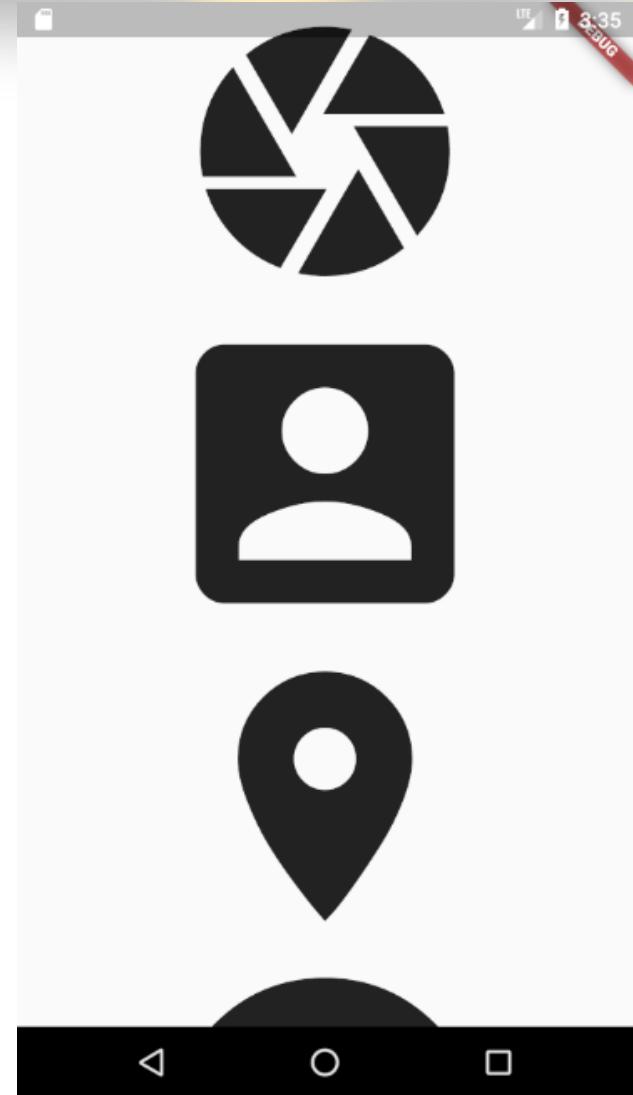


SingleChildScrollView សំរាប់ Column

```
SingleChildScrollView(  
    child: Column(  
        children: <Widget>[  
            Icon(Icons.camera, size: 200,),  
            Icon(Icons.account_box, size: 230.0,),  
            Icon(Icons.place, size: 200.0,),  
            Icon(Icons.info, size: 250.0,),  
        ],  
    ),  
,
```

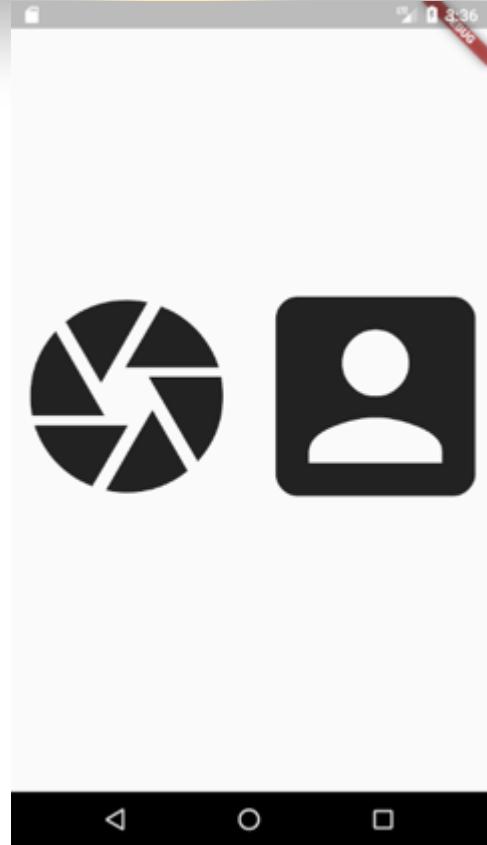
សំរាប់ Android គឺយើងគ្នារួចមិន physics អោយវាមាន
លក្ខណៈទាញយកទៅលម្អិតទាំង reload ទេ

```
SingleChildScrollView(  
    physics: BouncingScrollPhysics(),  
    child: Column(  
        children: <Widget>[  
            Icon(Icons.camera, size: 200,),  
            Icon(Icons.account_box, size: 230.0,),  
            Icon(Icons.place, size: 200.0,),  
            Icon(Icons.info, size: 250.0,),  
        ],  
    ),  
,
```



SingleChildScrollView នៃក្នុង Row

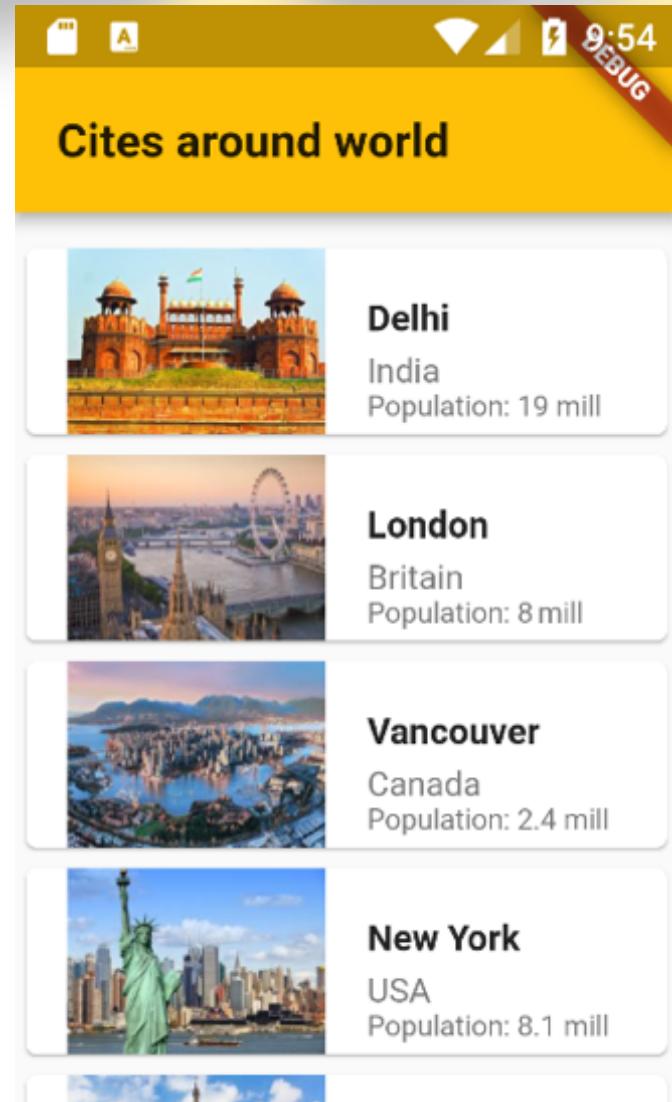
```
SingleChildScrollView(  
    physics: BouncingScrollPhysics(),  
    scrollDirection: Axis.horizontal,  
    child: Row(  
        children: <Widget>[  
            Icon(Icons.camera, size: 200,),  
            Icon(Icons.account_box, size: 230.0,),  
            Icon(Icons.place, size: 200.0,),  
            Icon(Icons.info, size: 250.0,),  
        ],  
    ),  
) ,
```



បំអិតលើ ListView, Detail Page, GridView និង PageView

ListView

- ListView គឺជា widget ដែលអាច scroll បាន ហើយគេប្រើវាស្ថើរតែគ្រប់ពេលសំភាប់ទិន្នន័យដែលប្រើនេះ។ យើងកំអាមេរិសដោក្នុងការ scroll ពីក្រោមឡើងលើ ពីស្តាំមកដែងកំបាន។
- ListView នៅក្នុង Flutter គឺល្អប្រើខ្លាំងណាស់ ដែលមិនធ្វើចូលជា ListView នៅក្នុង native Android ទេ ដែលវាគាមទារអាយុយយើងសរសើរក្នុងប្រើប្រាស់ដើម្បីទូប់បញ្ហាផួក ViewHolder, Lazy Load, និង cache image ជាដើម។ លើ ListView នៅក្នុង Flutter អនុញ្ញាតធមាយយើងដឹកទិន្នន័យប្រភេទអ្នកបាន អក្សរ រូបភាព បុទ្ទាជាបីជានីដៃអ្នកដោយ។

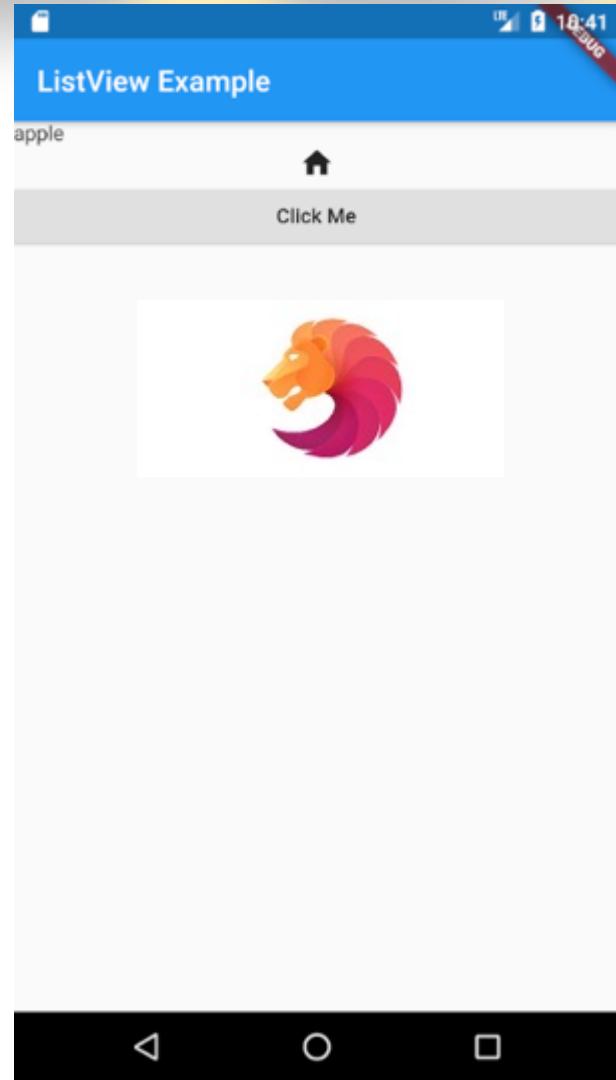


ListView Options

ListView មាន Option សំខាន់ៗដូចជា៖

- **children:** សំរាប់ជាក់ទិន្នន័យដែលមានប្រភេទណាមួយក៏បាន លាយត្តាក៏បាន អាយតែ សណ្ឋានរបស់ widget។
- **scrollDirection:** ប្រើសំរាប់ប្រឡទិសដោយ scroll
 - **Axis.horizontal:** scroll ផ្លូវតីឆ្លងឡាស្តា
 - **Axis.vertical:** scroll បញ្ចូរតីលីថុ៖ក្រាម
- **reverse:** true ៖ គឺអាយវារ៉ា scroll បញ្ចាស់
- **controller:** គឺសំរាប់យើងអាចសរសេរក្នុងអាយវារ៉ា scroll ដោយស្វ័យប្រវត្តិ
- **ListView.builder()** ៖ ប្រើសំរាប់ build ធាតុខាងក្រុងនៃ List

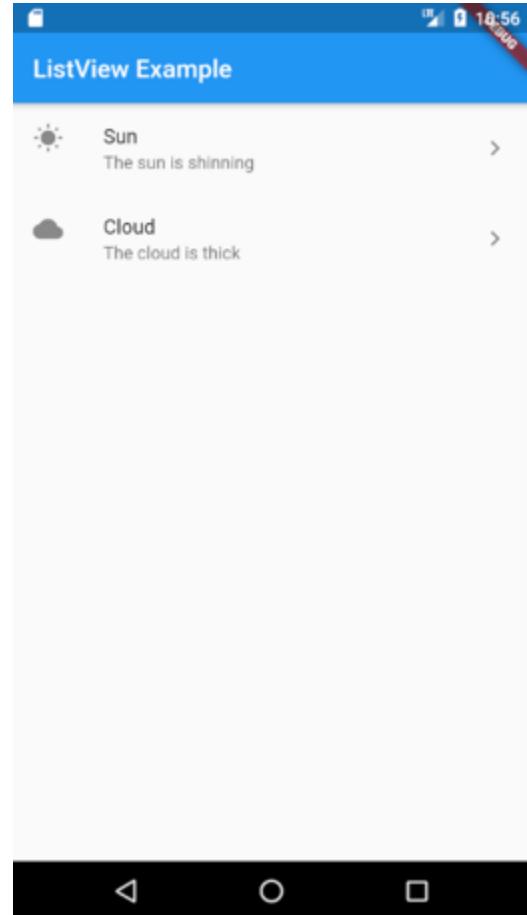
```
ListView(  
    scrollDirection: Axis.vertical,  
    reverse: false,  
    children: <Widget>[  
        Text("apple"),  
        Icon(Icons.home),  
        RaisedButton(  
            child: Text("Click Me"),  
            onPressed: () {},  
        ),  
        Container(  
            width: 300.0,  
            height: 200.0,  
            child: Image.network(  
                "http://bit.ly/2IGluzb",  
            ),  
        ),  
    ],  
)
```



ListView និង ListTile

- ListTile គឺជា widget ម្នាយដែលគេនិយមប្រើជាម្នាយ ListView ត្រូវ: ListTile មាន option ល្អុយចំនួនរបច្បាសបច្ចុប្បន្ន leading, title, subtitle, trailing, និង event click ជាដីម។

```
ListView(  
    children: <Widget>[  
        ListTile(  
            leading: Icon(Icons.wb_sunny),  
            title: Text('Sun'),  
            subtitle: Text("The sun is shinning"),  
            trailing: Icon(Icons.keyboard_arrow_right),  
        ),  
        ListTile(  
            leading: Icon(Icons.wb_cloudy),  
            title: Text('Cloud'),  
            subtitle: Text("The cloud is thick"),  
            trailing: Icon(Icons.keyboard_arrow_right),  
        ),  
    ],  
)
```

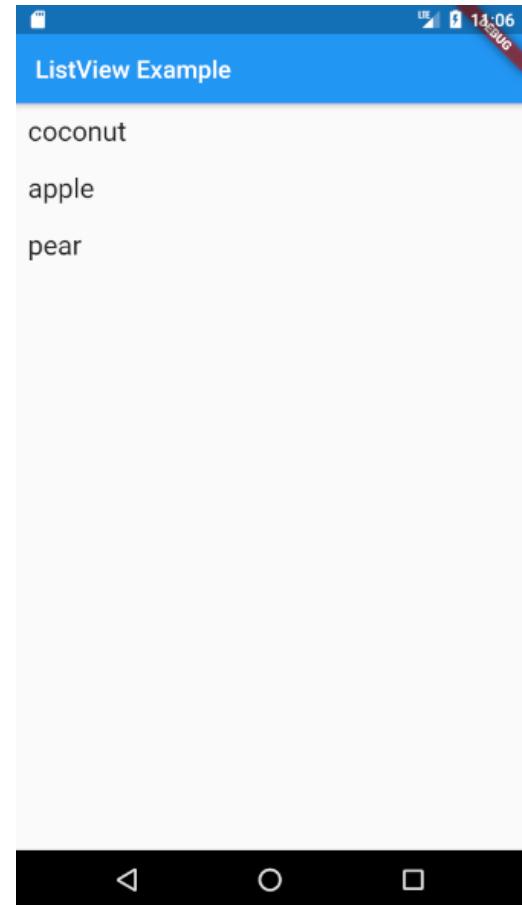


ListView.builder

ListView.builder គឺប្រើសំរាប់ជាក់ធាតុរបស់វាចាត់នៅ
dynamic។ ListView.builder មាន parameter សំខាន់ៗ
គឺ itemCount សំរាប់កំណត់ចំនួនធាតុ និង itemBuilder សំ
រាប់អាយូរើងបង្កើតធាតុជាលក្ខណៈ dynamic។ វាគ្មេការ
លើជាមួយនឹងទិន្នន័យប្រភេទជាឌី List។

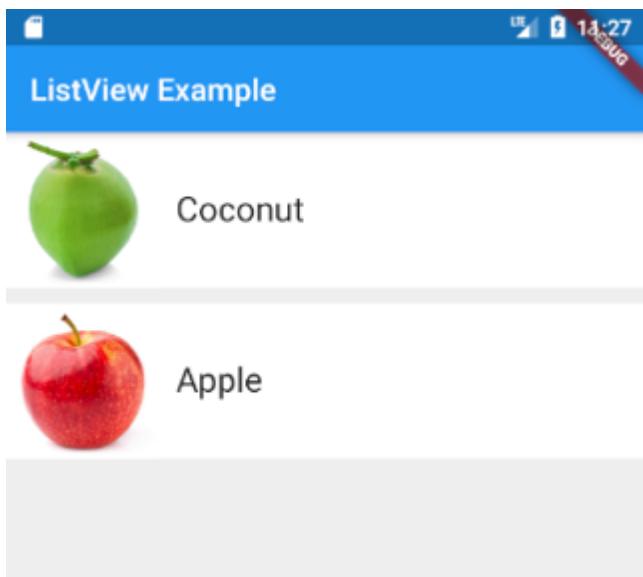
```
List<String> fruits = ["coconut", "apple", "pear"];
```

```
ListView.builder(  
    itemCount: fruits.length,  
    itemBuilder: (context, index) {  
        return Container(  
            padding: EdgeInsets.all(10.0),  
            child: Text(  
                fruits[index],  
                style: TextStyle(fontSize: 22.0),  
            ),  
        );  
    },
```



ListView.builder

```
class Fruit {  
    String title;  
    String imageUrl;  
  
    Fruit({this.title = "", this.imageUrl});  
}  
  
List<Fruit> fruits = [  
    Fruit(title: "Coconut",  
        imageUrl: "assets/coconut.jpg"),  
    Fruit(title: "Apple",  
        imageUrl: "assets/apple.jpg")  
];
```

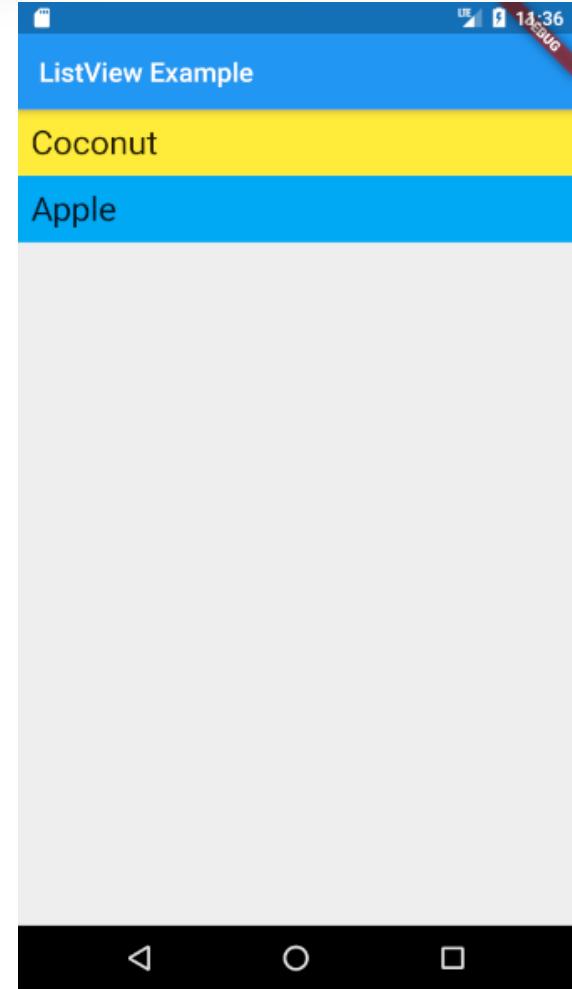


```
return Container(  
    color: Colors.grey[200],  
    child: ListView.builder(  
        itemCount: fruits.length,  
        itemBuilder: (context, index) {  
            return Container(  
                margin: EdgeInsets.only(bottom: 10.0),  
                color: Colors.white,  
                child: Row(  
                    children: <Widget>[  
                        Container(  
                            width: 100.0,  
                            height: 100.0,  
                            child: Image.asset(  
                                fruits[index].imageUrl,  
                                fit: BoxFit.cover,  
                            ),  
                        ),  
                        Container(  
                            padding: EdgeInsets.all(10.0),  
                            child: Text(  
                                fruits[index].title,  
                                style: TextStyle(fontSize: 22.0),  
                            ),  
                        ),  
                    ],  
                );  
        },  
    );
```

ListView.builder

យើងអាចសង្គតយើងក្នុងក្នុងមុននេះថា នៅក្នុង itemBuilder តី
មាន return widget ។ លក្ខណៈនេះគឺអាចអរយយើងអាចធ្វាសប្រ
ប្បកំណត់របស់ជាតុបានយ៉ាងងាយស្រួល។

```
ListView.builder(  
    itemCount: fruits.length,  
    itemBuilder: (context, index) {  
        if(index % 2 == 0){  
            return Container(  
                color: Colors.yellow,  
                padding: EdgeInsets.all(10.0),  
                child: Text(fruits[index].title,  
                    style: TextStyle(fontSize: 25.0),),  
            );  
        }  
        else{  
            return Container(  
                color: Colors.lightBlue,  
                padding: EdgeInsets.all(10.0),  
                child: Text(fruits[index].title,  
                    style: TextStyle(fontSize: 25.0),),  
            );  
        }  
    })
```



Detail Page

ជាចម្លោត ListView គឺអាចបង្ហាញទិន្នន័យបានប្រចើនមែន តែដោយសារទំហំអក្សរក្នុង យើងមិនអាចបង្ហាញត្រូវបែងទិន្នន័យទាំងអស់ទេ។ ដូច្នេះយើងត្រូវ Page មួយឡើត ដើម្បីធ្វើនូយក្នុងបង្ហាញទិន្នន័យអាយុយលំអិតជាមុន។ យើងអាចជាក់ event tab លើ item និមួយបានតាមរយៈ widget មួយចំនួនដូចជា ListTile, InkWell, RaisedButton, FlatButton, IconButton ជាដើម។ ហើយពេលយើងចូចលើ item និមួយនៅ៖ហើយ យើងនឹងបើក Page មួយឡើតប៉ឺ Navigator។

```
ListView.builder(  
    itemCount: fruits.length,  
    itemBuilder: (context, index) {  
        return ListTile(  
            title: Text(fruits[index].title),  
            trailing: Icon(Icons.keyboard_arrow_right),  
            onTap: (){  
                Navigator.push(context,  
                    MaterialPageRoute(builder: (context)=> DetailPage()));  
            },  
        );  
    }  
)
```

Navigator សំរាប់បើក Page មួយទៀត

Navigator មានត្រូវនាទីសំរាប់បើកផ្ទាំង Page មួយដែលទេរកឡើង។ យើងអាចប្រើ push សំរាប់បើក Page ចិត្តតាម constructor បុប្ផិយៗ pushNamed សំរាប់បើក Page ដែលបានកំណត់ហើយក្នុង routes។

- Push:

```
Navigator.push(context, MaterialPageRoute(builder: (context) => NextPage()));
```

- PushNamed ជាមួយ Route:

```
Navigator.of(context).pushNamed("/nextpage");
```

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
//      home: MyHome(), //home cannot be with routes
      routes: {
        "/" : (context) => MyHome(),
        "/nextpage" : (context) => NextPage(),
      },
    );
}
```

Navigating ដោយមិនគ្រប់គ្រាយ

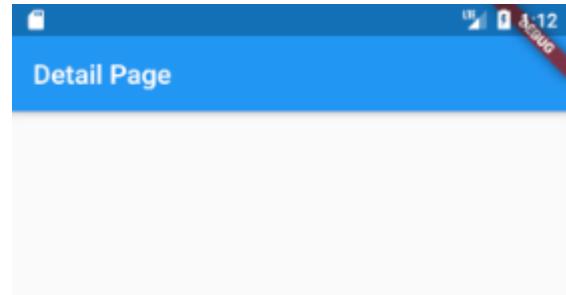
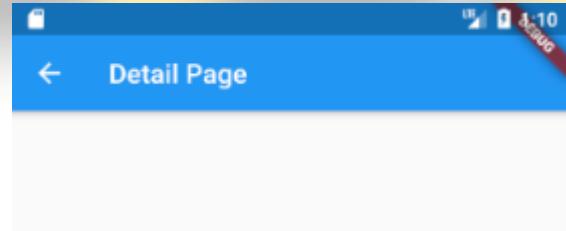
ពេលខ្លះយើងបើក Page ថ្មីហើយតើយើងមិនចង់អាយកេបក
ក្រាយបានទៅ។ ដើម្បីធ្វើចំងារបានតើយើងអាចប្រើ
pushReplacement ឬ pushReplacementNamed។

- PushReplacement:

```
Navigator.pushReplacement(context,  
    MaterialPageRoute(builder: (context) =>  
        NextPage()));
```

- PushReplaceNamed:

```
Navigator.of(context).pushReplacementNamed("/nextpage");
```



បញ្ជីនទិន្នន័យជាមួយ Navigator

- យើងអាចប្រើ constructor ដើម្បី បញ្ជីនទិន្នន័យជាមួយនឹង Navigator។

```
Navigator.push(context, MaterialPageRoute(builder: (context) =>
    NextPage(text: "some text")
));
```

```
class NextPage extends StatefulWidget {

    final String text;
    NextPage({this.text});

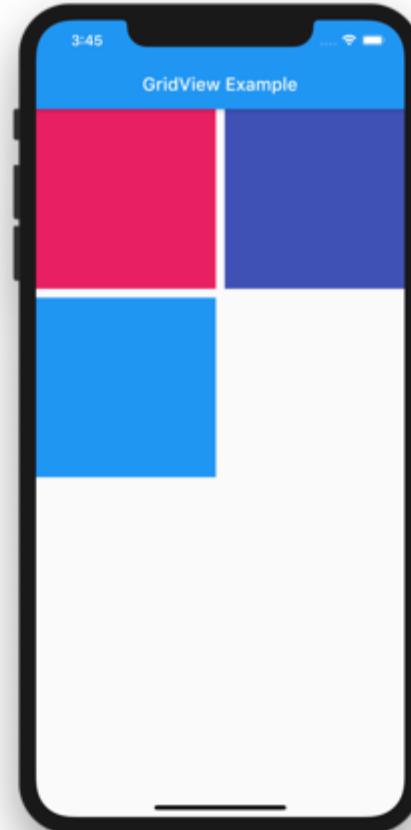
    @override
    _NextPageState createState() => _NextPageState();
}

class _NextPageState extends State<NextPage> {
    @override
    Widget build(BuildContext context) {
        return Container(child: Text(this.widget.text));
    }
}
```

GridView: GridView.count

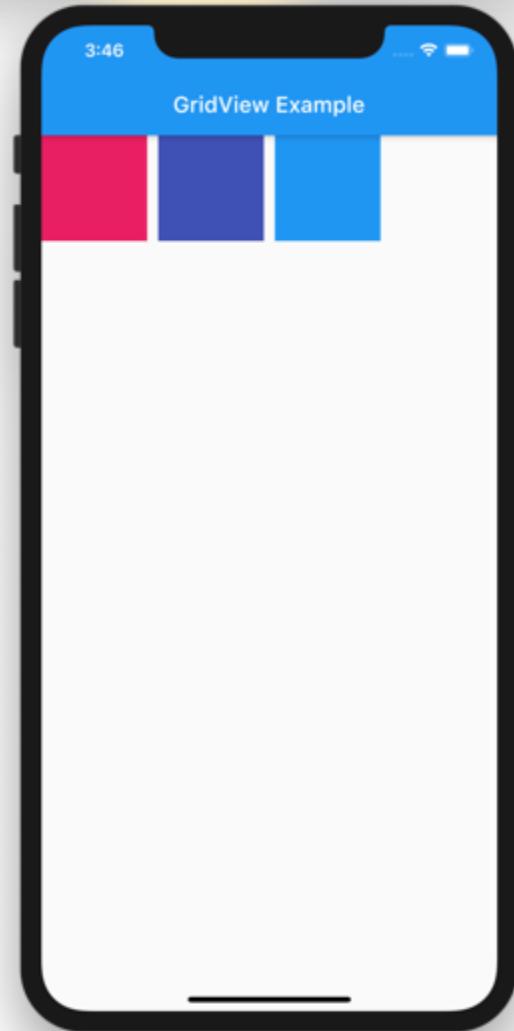
- GridView គឺសំរាប់បង្ហាញ Item ជាលក្ខណៈ Grid ដើម្បីទាំងអស់ជាមុខងារ Matrix។
- GridView.count()៖ យើងអាចប្រើវា ប្រសិនបើចង់បានចំនួន Column ជាក់លាក់
- GridView.extent()៖ យើងអាចប្រើវា ប្រសិនបើចង់កំណត់ប្រវែងផ្ទៃករបស់ Item

```
GridView.count(  
    crossAxisCount: 2,  
    mainAxisSpacing: 10,  
    crossAxisSpacing: 10,  
    children: <Widget>[  
        Container(color: Colors.pink),  
        Container(color: Colors.indigo),  
        Container(color: Colors.blue),  
    ],  
) ,
```



GridView.extent

```
GridView.extent(  
  maxCrossAxisExtent: 100,  
  mainAxisSpacing: 10,  
  crossAxisSpacing: 10,  
  children: <Widget>[  
    Container(color: Colors.pink),  
    Container(color: Colors.indigo),  
    Container(color: Colors.blue),  
  ],  
)
```



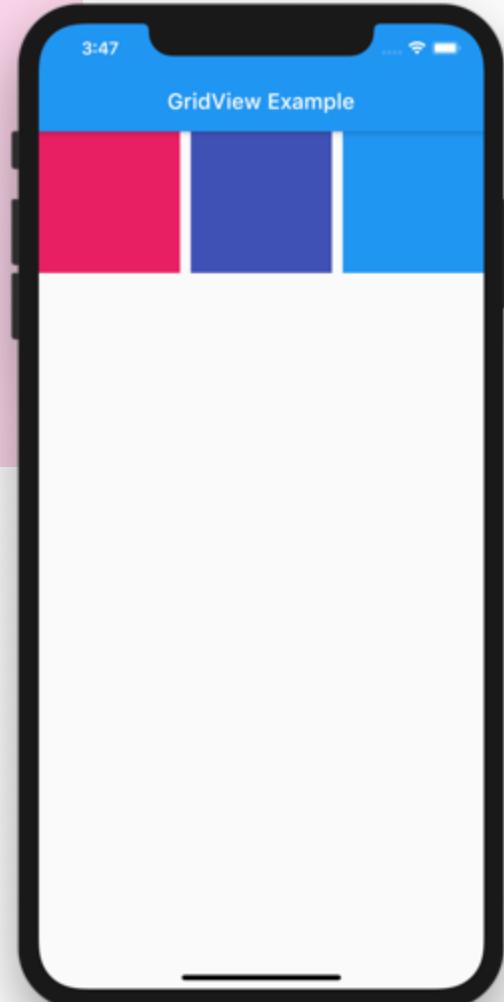
GridView ជាមួយ gridDelegate

GridView(

```
gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
    crossAxisCount: 3,  
    mainAxisSpacing: 10,  
    crossAxisSpacing: 10,  
,  
    children: <Widget>[  
        Container(color: Colors.pink),  
        Container(color: Colors.indigo),  
        Container(color: Colors.blue),  
    ],  
,
```

gridDelegate អាចជា៖

- SliverGridDelegateWithFixedCrossAxisCount
- SliverGridDelegateWithMaxCrossAxisExtent

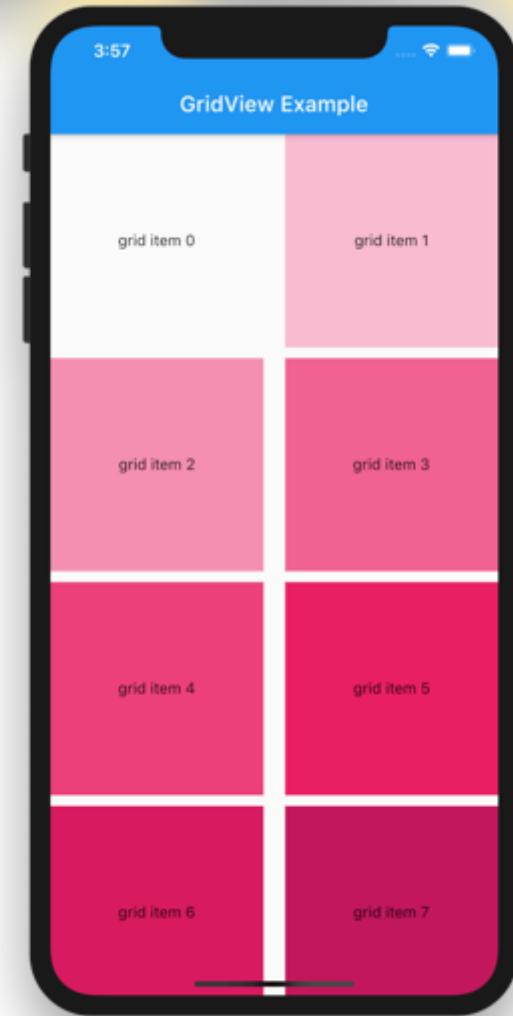


GridView.builder

GridView.builder គឺប្រើសំរាប់ build item ដែលមានទំនុកត្រូវដោយនឹង ListView.builder ដើម្បី រាយការណ៍ Option សំខាន់ៗ

1. itemCount: ចំនួនរបស់ item
2. gridDelegate ដែលអាចជាំ
 1. SliverGridDelegateWithFixedCrossAxisCount
 2. SliverGridDelegateWithMaxCrossAxisExtent
3. itemBuilder ប្រើសំរាប់ build item

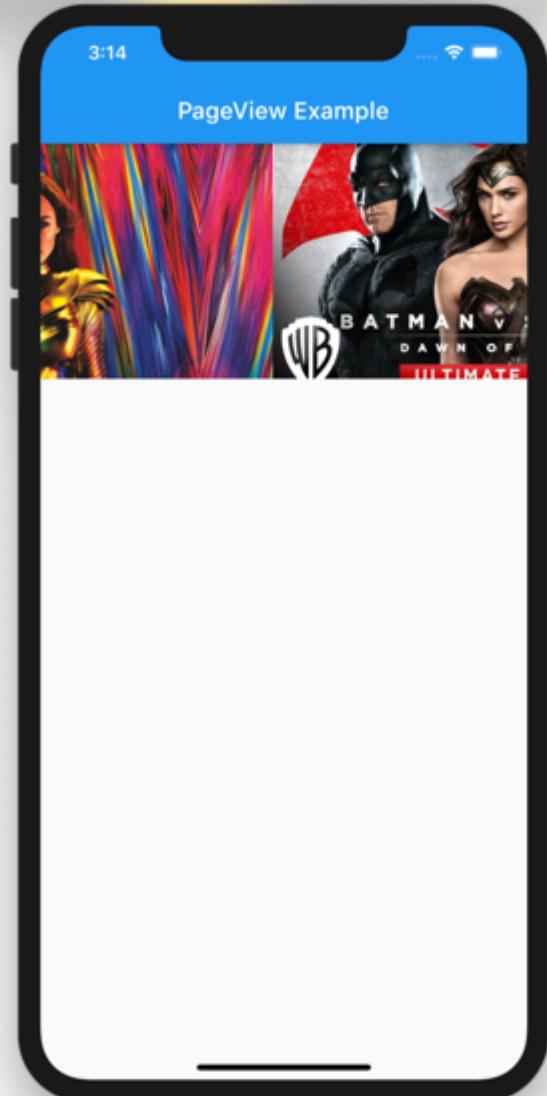
```
GridView.builder(  
    itemCount: 10,  
    gridDelegate:  
        SliverGridDelegateWithFixedCrossAxisCount(  
            crossAxisCount: 2,  
            mainAxisSpacing: 10,  
            crossAxisSpacing: 20,  
        ),  
    itemBuilder: (context, index) {  
        return Container(  
            alignment: Alignment.center,  
            color: Colors.pink[100 * (index % 9)],  
            child: Text('grid item $index'),  
        );  
    },
```



PageView

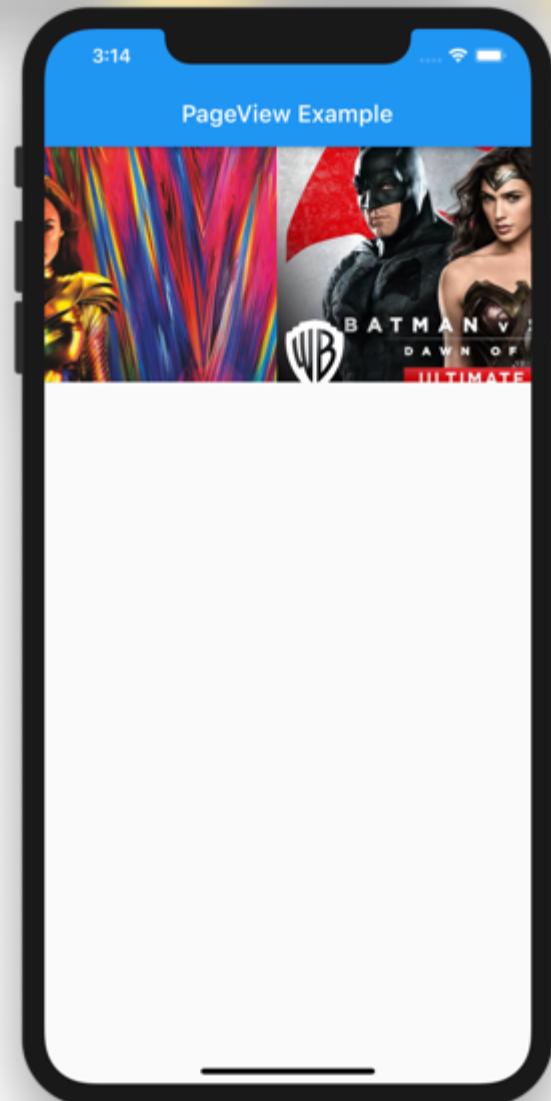
PageView មានទំនួងដូចជា ListView ដើរ តែវា scroll ម្ចងម្ខយ Page ។ Option មាន៖

- scrollDirection
- physics
- controller



```
Container(  
  alignment: Alignment.topCenter,  
  height: 200,  
  child: PageView(  
    physics: BouncingScrollPhysics(),  
    scrollDirection: Axis.horizontal,  
    reverse: false,  
    pageSnapping: true,  
    children: <Widget>[  
      Image.network(_img1, fit: BoxFit.cover,),  
      Image.network(_img2, fit: BoxFit.cover,),  
      Image.network(_img3, fit: BoxFit.cover,),  
    ],  
,  

```

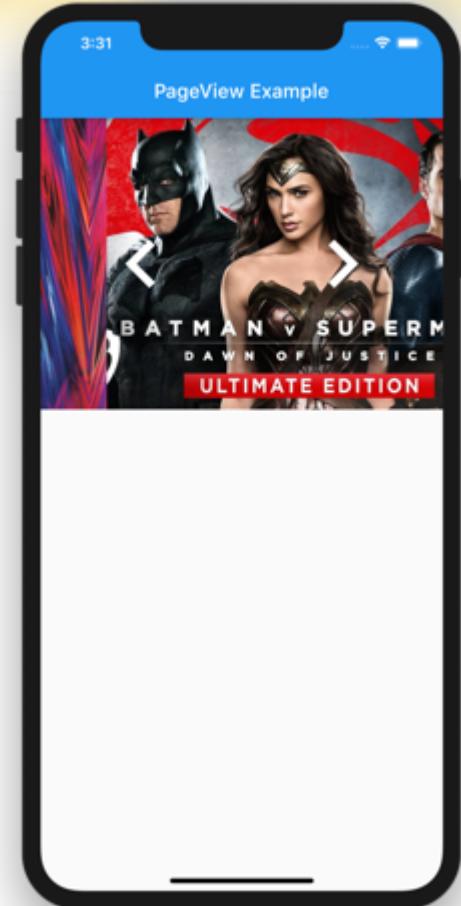


PageView ជាមួយ PageController

```
PageController _scroller = PageController();
int _currentIndex = 0;

get _buildBody {
    return Stack(
        alignment: Alignment.center,
        children: <Widget>[
            _buildPageView,
            _buildNavigations,
        ],
    );
}

get _buildPageView {
    return Container(
        alignment: Alignment.topCenter,
        height: 300,
        child: PageView(
            controller: _scroller,
            physics: BouncingScrollPhysics(),
            scrollDirection: Axis.horizontal,
            reverse: false,
            pageSnapping: true,
            onPageChanged: (index) {
                setState(() {
                    _currentIndex = index;
                });
            },
            children: <Widget>[
                Image.network(_img1, fit: BoxFit.cover),
                Image.network(_img2, fit: BoxFit.cover),
                Image.network(_img3, fit: BoxFit.cover),
            ],
        ),
    );
}
```



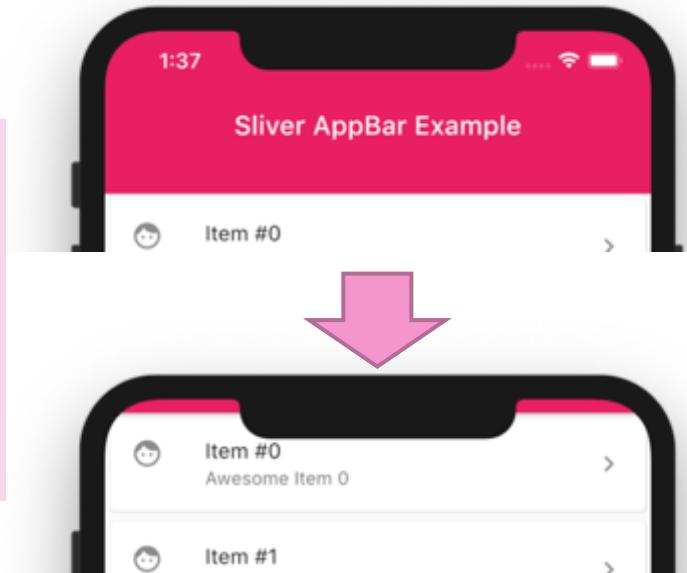
```
get _buildNavigations {
  return Row(
    mainAxisAlignment: MainAxisAlignment.spaceAround,
    children: <Widget>[
      Container(
        width: 120, height: 120,
        child: IconButton(
          icon: Icon(Icons.navigate_before, color: Colors.white, size: 100),
          onPressed: () {
            _scroller.animateToPage(_currentIndex - 1,
              duration: Duration(milliseconds: 300),
              curve: Curves.easeInOut,
            );
          },
        ),
      ),
      Container(
        width: 120, height: 120,
        child: IconButton(icon: Icon(Icons.navigate_next, color: Colors.white, size: 100),
          onPressed: () {
            _scroller.animateToPage(_currentIndex + 1,
              duration: Duration(milliseconds: 300),
              curve: Curves.easeInOut,
            );
          },
        ),
      ),
    ],
  );
}
```

CustomScrollView with SliverAppBar, SliverList and SliverGrid

CustomScrollView

- CustomScrollView គឺជា widget មួយដែលអនុញ្ញាតឱ្យអាចធានាកំបញ្ឈូល Sliver ផ្សេងៗគ្នា ចូលទៅក្នុងវាងបាន ដែលវានឹងបង្កើននូវសេវាកំណត់ដោយការ Scroll ចុះទេរីនបានស្ថាតជាងមុន។ ជាមួយតាលេយើងប្រើ AppBar គឺវានៅជាប់ស្វែម កំបុងតាលេយើងប្រើ SliverAppBar ដែលជាក្នុងរបស់ CustomScrollView នេះ: វានឹងធ្វើឡាយ SliverAppBar លិចចាត់ នៅពេលដែលយើង Scroll ចុះក្រាយ។
- Sliver ត្រូវបានកំណត់ ប្រចាំជាប់ភាព។

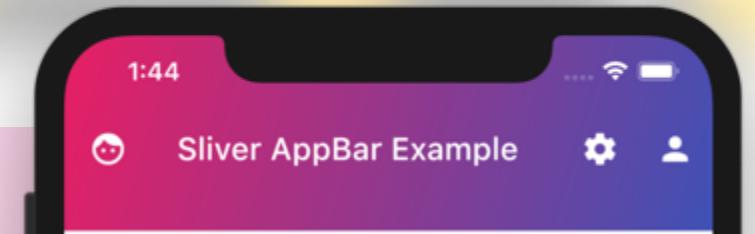
```
CustomScrollView(slivers: [  
    SliverAppBar(...),  
    SliverPadding(...),  
    SliverList(...),  
    SliverGrid(...),  
    SliverToBoxAdapter(child: ),  
]
```



SliverAppBar

- នៅក្នុង CustomScrollView, ក្នុងដែលសំខាន់ជាងគៀ SliverAppBar និនេះអាចរារីនឹងលិចបាត់ បើយើង Scroll ចុះក្រាយ ហើយវានឹងលេចចេញកិច្ច ពេលយើង Scroll ទើនឹងលិចបាត់។
- Option សំខាន់របស់វាតី:
 - pinned:** បើ false គឺវានឹងបាត់ពេល Scroll ចុះ, តើបើ true គឺវាអត់បាត់ចេញទេ
 - floating:** បើ true វានឹងរហាយបាត់មួងបន្ទិច។ ពេលយើង Scroll ចុះ
 - flexibleSpace:** វាតីជា Widget ដែលជាដៃគូបាត់ background។ វាតាម Option ដែលយើងអាចរចនាថោអ្នក៍បាន អាចប្រើ Container ហើយជាក់ decoration ក៍បាន, បុអាចជាក់រូបភាពក៍បាន។
 - expandedHeight:** កំពស់របស់ AppBar ពេលដែលរារាងអស់។

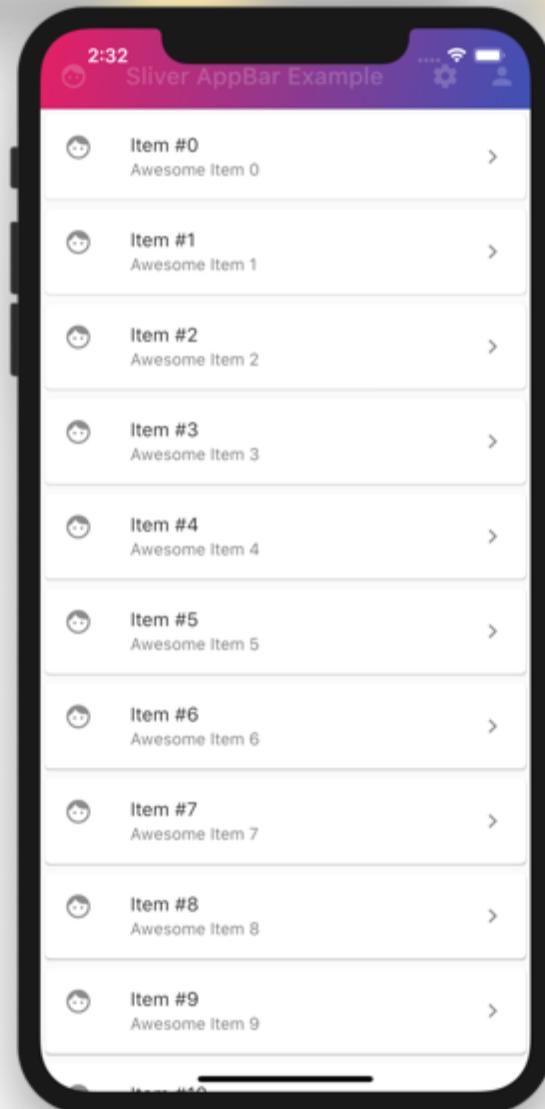
```
SliverAppBar(  
    leading: Icon(Icons.face),  
    actions: <Widget>[  
        IconButton(icon: Icon(Icons.settings), onPressed: (){},),  
        IconButton(icon: Icon(Icons.person), onPressed: (){},),  
    ],  
    pinned: false,  
    title: Text("Sliver AppBar Example"),  
    floating: true,  
    flexibleSpace: Container(  
        decoration: BoxDecoration(  
            gradient: LinearGradient(  
                begin: Alignment.topLeft,  
                end: Alignment.bottomRight,  
                colors: [Colors.pink, Colors.indigo],  
            ),  
        ),  
    ),  
    expandedHeight: 80,  
)
```



SliverList

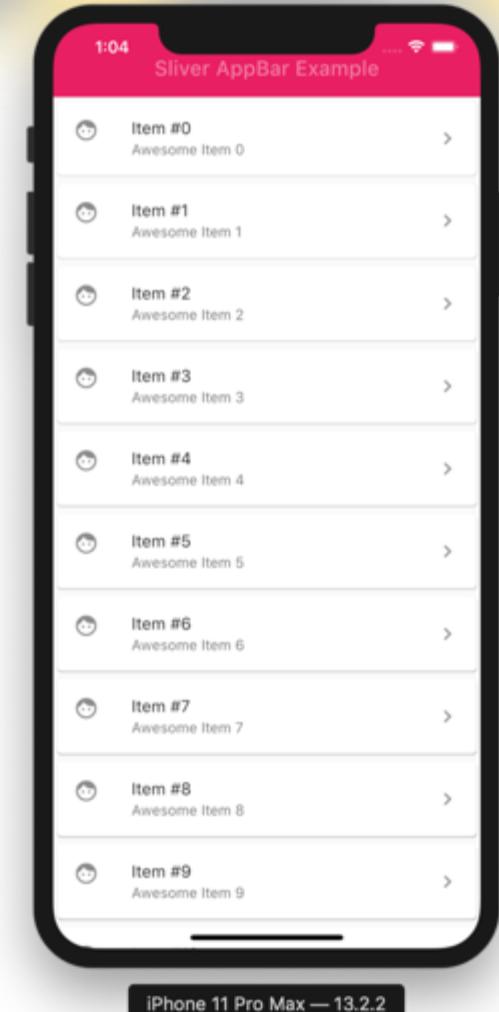
- SliverList គឺជំបះជាការបង្កើតនៃក្នុង ListView ដូចជាអ្នកអាជីវកម្ម SliverList គឺមានកែវ Option តើមួយគត់គឺ delegate ដែលមានប្រភេទជាសង្គមជាក់ ដើម្បីផ្តល់ជាបន្ទូលបានជំបះជាក់
 - SliverChildListDelegate: នេះគឺជំបះជាការបង្កើតនៃក្នុង ListView ដូចជា
 - SliverChildBuilderDelegate: នេះគឺជំបះជាការបង្កើតនៃក្នុង ListView.builder ដូចជា
- យើងអាចប្រើ SliverFixedExtentList បាន ហើយយើងចង់បានអាយុយទំហំកំពស់របស់ Item ជំបះ។

```
SliverList(  
  delegate: SliverChildBuilderDelegate(  
    (context, index) {  
      return Card(  
        child: ListTile(  
          leading: Icon(Icons.face),  
          title: Text('Item #$index'),  
          subtitle: Text("Awesome Item $index"),  
          trailing: Icon(Icons.navigate_next),  
        ),  
      );  
    }, childCount: 100),  
,
```



Full Code:

```
return Scaffold(  
  body: CustomScrollView(  
    slivers: <Widget>[  
      SliverAppBar(  
        title: Text("Sliver AppBar Example"),  
        floating: true,  
        flexibleSpace: Container(  
          color: Colors.pink,  
        ),  
        expandedHeight: 80,  
      ),  
      SliverList(  
        delegate: SliverChildBuilderDelegate((context, index) {  
          return Card(  
            child: ListTile(  
              leading: Icon(Icons.face),  
              title: Text('Item #${index}'),  
              subtitle: Text("Awesome Item ${index}"),  
              trailing: Icon(Icons.navigate_next),  
            ),  
          );  
        }, childCount: 100),  
      ),  
    ],  
  );
```



SliverGrid

- SliverGrid គឺជំពូជាក្នុង GridView វាគាមទារអាយងក់ Option ចំនួន២ នៅយក្សាតី gridDelegate និង delegate។

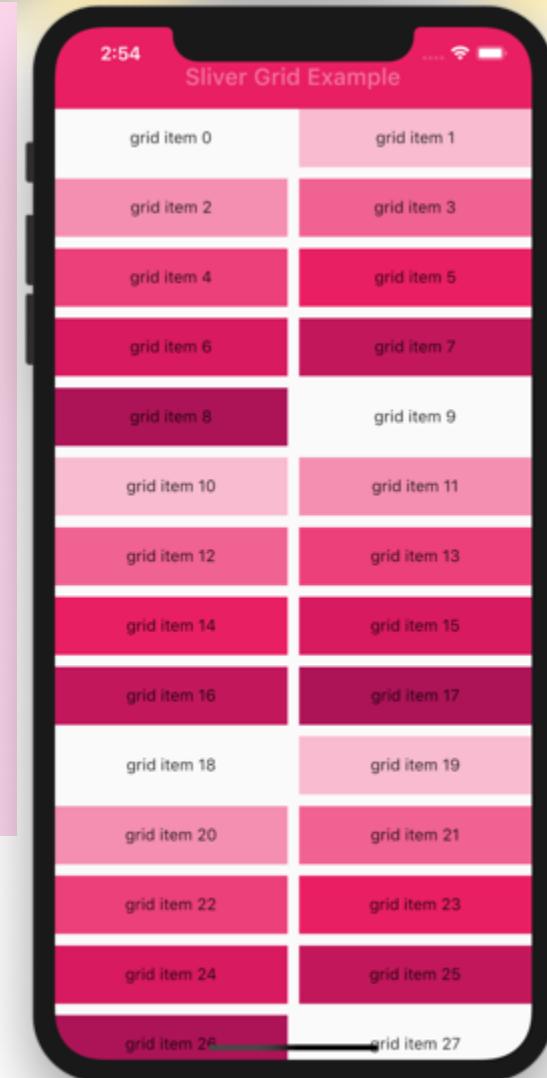
1. gridDelegate អាចជាកំច្បែលនូវ៖

- SliverGridDelegateWithFixedCrossAxisCount: យើងអាចកំណត់ចំនួន Column អាយការមួយ: crossAxisCount
- SliverGridDelegateWithMaxCrossAxisExtent: យើងកំណត់ទំហំផ្ទៃកអាយការបាន តាមរយៈ: maxCrossAxisExtent

2. delegate អាចជាកំច្បែលនូវ៖

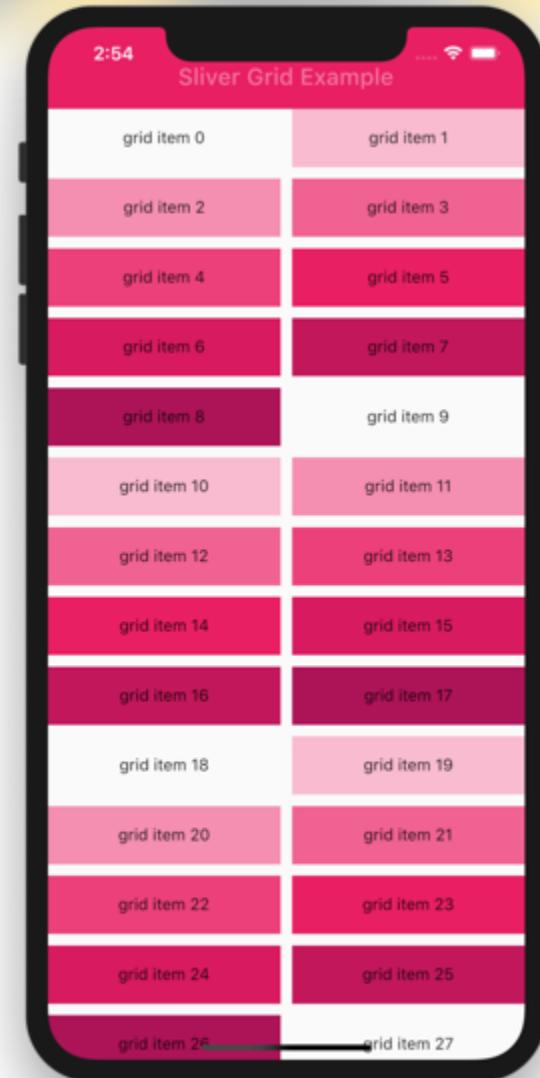
- SliverChildListDelegate: មានចំនួន Item កំណត់, ដូចគ្នានឹង ListView
- SliverChildBuilderDelegate: យើងអាច build item បាន, ដូច ListView builder

```
SliverGrid(  
    gridDelegate: SliverGridDelegateWithMaxCrossAxisExtent(  
        maxCrossAxisExtent: 200,  
        mainAxisSpacing: 10.0,  
        crossAxisSpacing: 10.0,  
        childAspectRatio: 4.0,  
    ),  
    delegate: SliverChildBuilderDelegate(  
        (BuildContext context, int index) {  
            return new Container(  
                alignment: Alignment.center,  
                color: Colors.pink[100 * (index % 9)],  
                child: new Text('grid item $index'),  
            );  
        },  
        childCount: 50,  
    ),  
,
```



Full code:

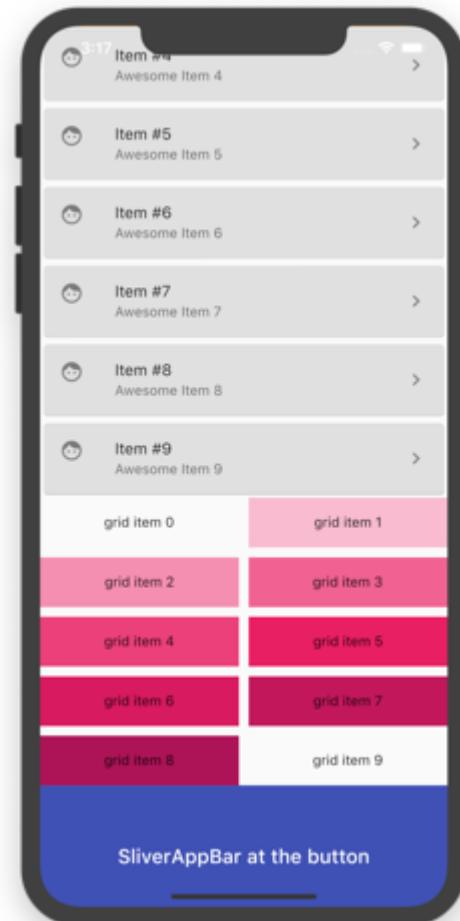
```
return Scaffold(  
    body: CustomScrollView(  
        slivers: <Widget>[  
            SliverAppBar(  
                pinned: false, floating: true,  
                title: Text("Sliver Grid Example"), expandedHeight: 80,  
                flexibleSpace: Container(color: Colors.pink,),  
>,  
            SliverGrid(  
                gridDelegate: SliverGridDelegateWithMaxCrossAxisExtent(  
                    maxCrossAxisExtent: 200, mainAxisSpacing: 10.0,  
                    crossAxisSpacing: 10.0, childAspectRatio: 4.0,  
>,  
                delegate: SliverChildBuilderDelegate(  
                    (BuildContext context, int index) {  
                        return new Container(  
                            alignment: Alignment.center,  
                            color: Colors.pink[100 * (index % 9)],  
                            child: new Text('grid item $index'),  
                        );  
                    },  
                    childCount: 50,  
>,  
                ),  
            ),  
        ],  
>);
```



SliverAppBar + SliverList + SliverGrid

- ចំនួចពិសេសនៅក្នុង CustomScrollView តើយើងអាចដាក់ SliverAppBar លាយទៅងីរ និង SliverList ជាមួយនឹង SliverGrid បាន។ មិនតែបុណ្យានេះ យើងកំអាចដាក់ SliverAppBar ខាងក្រោមបំផុតកំបាន។

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    body: CustomScrollView(  
      slivers: <Widget>[  
        SliverAppBar(...),  
        SliverList(...),  
        SliverGrid(...),  
        SliverAppBar(...),  
      ],  
    ),  
  );  
}
```

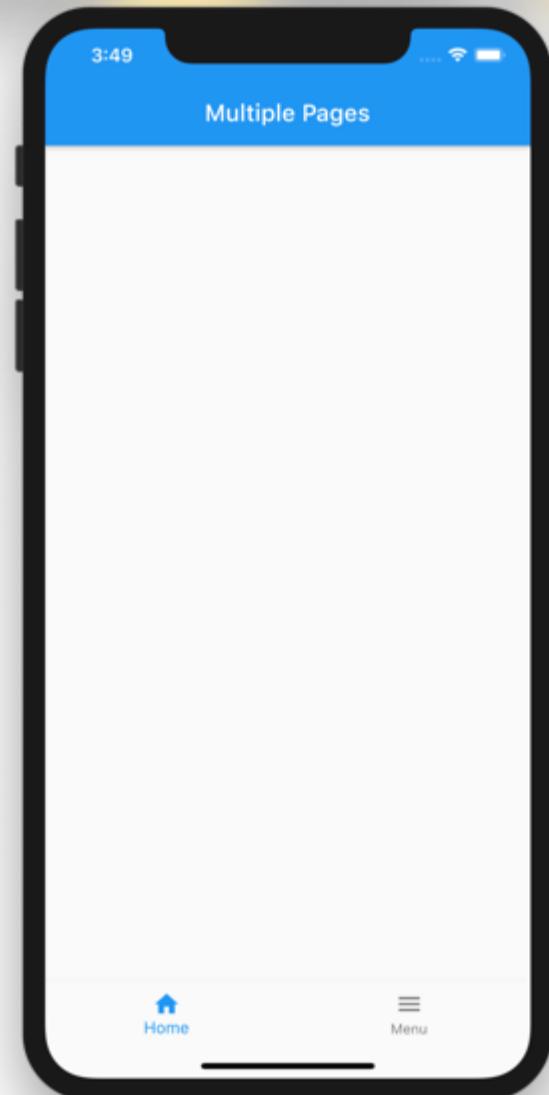


Multiple Pages ជាមួយ BottomNavigationBar

BottomNavigationBar

- BottomNavigationBar គឺជា Widget ក្នុងរបស់ Scaffold ដែលវាប្រើសំរាប់ជាក់ menu ជាកំណែង icon នៅផ្ទៃកខាងក្រោមនៃ App។
- ពេលប្រើវា គីឡូទាមទារអាយមាន Option សំខាន់ៗខាងក្រោម៖
 1. items: ដែលមានប្រភេទជា List នៃ BottomNavigationBarItem
 2. ត្រូវមានយ៉ាងតិច២ items
 3. ហើយនៅក្នុង BottomNavigationBarItem គីត្រូវមាន icon និង title

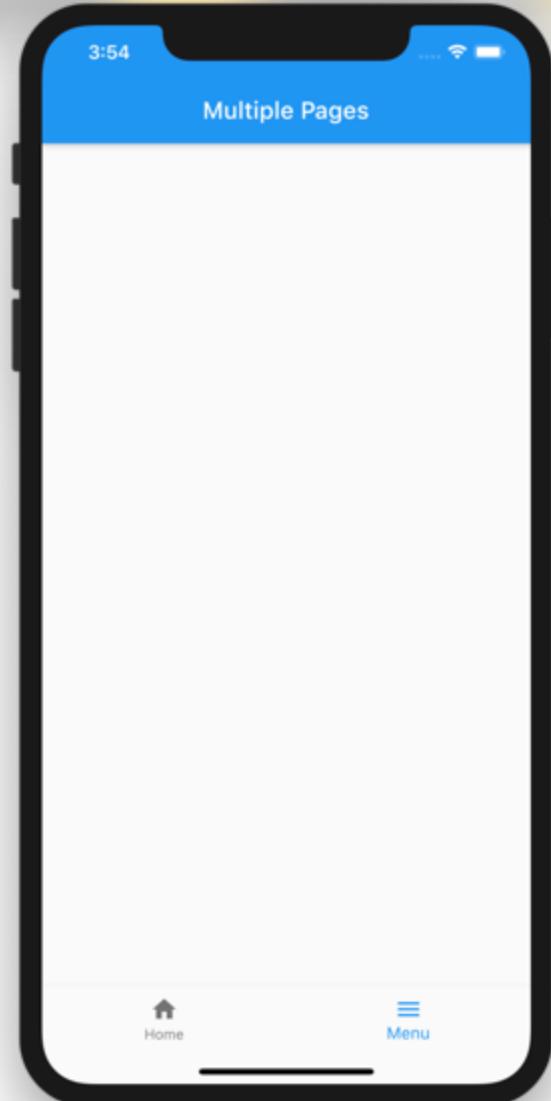
```
return BottomNavigationBar(  
    items: <BottomNavigationBarItem>[  
        BottomNavigationBarItem(  
            icon: Icon(Icons.home),  
            label: "Home",  
        ),  
        BottomNavigationBarItem(  
            icon: Icon(Icons.menu),  
            label: "Menu",  
        ),  
    ],  
);
```



តែងដើម្បីអាចអាយចុចលើ menu និមួយបាន យើងត្រូវ
បន្ថែម currentIndex និង onTap:

```
int _cIndex = 0;

get _buildBottom {
    return BottomNavigationBar(
        onTap: (index){
            setState(() {
                _cIndex = index;
            });
        },
        currentIndex: _cIndex,
        items: <BottomNavigationBarItem>[
            BottomNavigationBarItem(
                icon: Icon(Icons.home),
                label: "Home",
            ),
            BottomNavigationBarItem(
                icon: Icon(Icons.menu),
                label: "Menu",
            ),
        ],
    );
}
```



BottomNavigationBar ជាមួយនឹង PageView

1. ដើម្បីអាយយោងអាចចូចបញ្ជា menu អាយធានាសំបុរ Screen តីមួយត្រូវបន្ថែម PageView មួយឡើតនៅក្នុង body
2. តែយោងត្រូវបង្កើត Screen និមួយនៅក្នុង class ធ្វើងទូទាត់ ព្រមទាំងការងារធ្វើងទូទាត់
3. បន្ទាប់មកយោងទាញយក Screen ទាំងនេះមកជាក់ក្នុង List មួយ ដើម្បីយកឡាដាក់ចូលដើរការទិន្នន័យបស់ PageView។
4. ហើយដើម្បីអាយ Screen ចាំទីតាំង Scroll ពេលចេញហើយចូលវិញ តីនៅក្នុង class Screen និមួយនេះ យើងជាក់បន្ថែម Mixin មួយណាម៉ោះថា AutomaticKeepAliveClientMixin ព្រមទាំង override getter wantKeepAlive ជាក់អាយស្តី true។

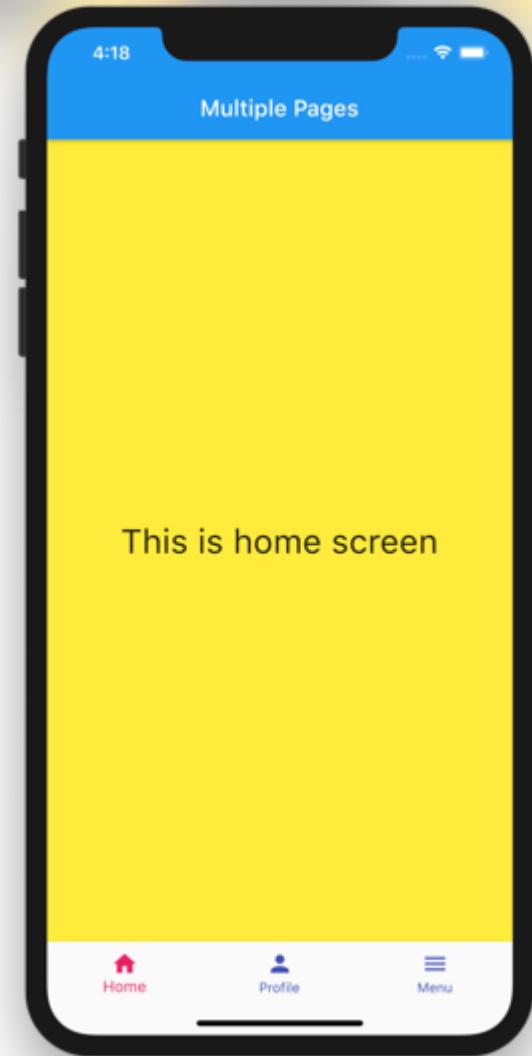
■ Full Code:

```
1 import 'package:flutter/material.dart';
2
3 class MyBottomNavigationBarExample extends StatefulWidget {
4     @override
5     _MyBottomNavigationBarExampleState createState() =>
6         _MyBottomNavigationBarExampleState();
7 }
8
9 class _MyBottomNavigationBarExampleState
10    extends State<MyBottomNavigationBarExample> {
11     List<Widget> _pageList;
12
13     HomeScreen _homeScreen = HomeScreen();
14     ProfileScreen _profileScreen = ProfileScreen();
15     MenuScreen _menuScreen = MenuScreen();
16
17     @override
18     void initState() {
19         super.initState();
20
21         _pageList = [_homeScreen, _profileScreen, _menuScreen];
22     }
23 }
```

```
24 @override
25 Widget build(BuildContext context) {
26   return Scaffold(
27     appBar: AppBar(
28       title: Text("Multiple Pages"),
29     ),
30     body: _buildBody,
31     bottomNavigationBar: _buildBottomNavigationBar,
32   );
33 }
34
35 PageController _scroller = PageController();
36
37 get _buildBody {
38   return Container(
39     alignment: Alignment.center,
40     child: PageView(
41       controller: _scroller,
42       physics: NeverScrollableScrollPhysics(),
43       children: _pageList,
44       onPageChanged: (index) {
45         setState(() {
46           _currentIndex = index;
47         });
48       },
49     ),
50   );
51 }
52 }
```

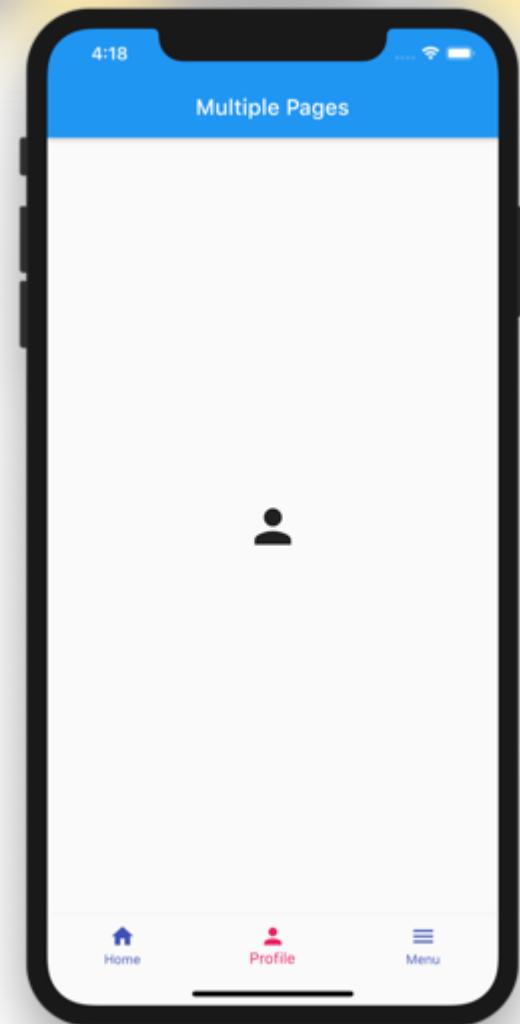
```
53     int _currentIndex = 0;
54
55     Widget get _buildBottomNavigationBar {
56         return BottomNavigationBar(
57             selectedItemColor: Colors.pink,
58             unselectedItemColor: Colors.indigo,
59             currentIndex: _currentIndex,
60             onTap: (index) {
61                 setState(() {
62                     _scroller.animateToPage(index,
63                         duration: Duration(milliseconds: 300), curve: Curves.easeOut);
64                 });
65             },
66             items: <BottomNavigationBarItem>[I
67                 BottomNavigationBarItem(
68                     icon: Icon(Icons.home),
69                     label: "Home",
70                 ),
71                 BottomNavigationBarItem(
72                     icon: Icon(Icons.person),
73                     label: "Profile",
74                 ),
75                 BottomNavigationBarItem(
76                     icon: Icon(Icons.menu),
77                     label: "Menu",
78                 ),
79                 I,
80             );
81         }
82     }
```

```
84 class HomeScreen extends StatefulWidget {
85     @override
86     _HomeScreenState createState() => _HomeScreenState();
87 }
88
89     class _HomeScreenState extends State<HomeScreen>
90     with AutomaticKeepAliveClientMixin {
91         @override
92         bool get wantKeepAlive => true;
93
94         @override
95         Widget build(BuildContext context) {
Formatted 8 lines
Show reformat dialog: ⌘K
96             scaffold(
97                 body: Container(
98                     alignment: Alignment.center,
99                     color: Colors.yellow,
100                    child: Text(
101                        "This is home screen",
102                        style: TextStyle(fontSize: 30),
103                    ),
104                ),
105            );
106        }
107    }
108 }
```



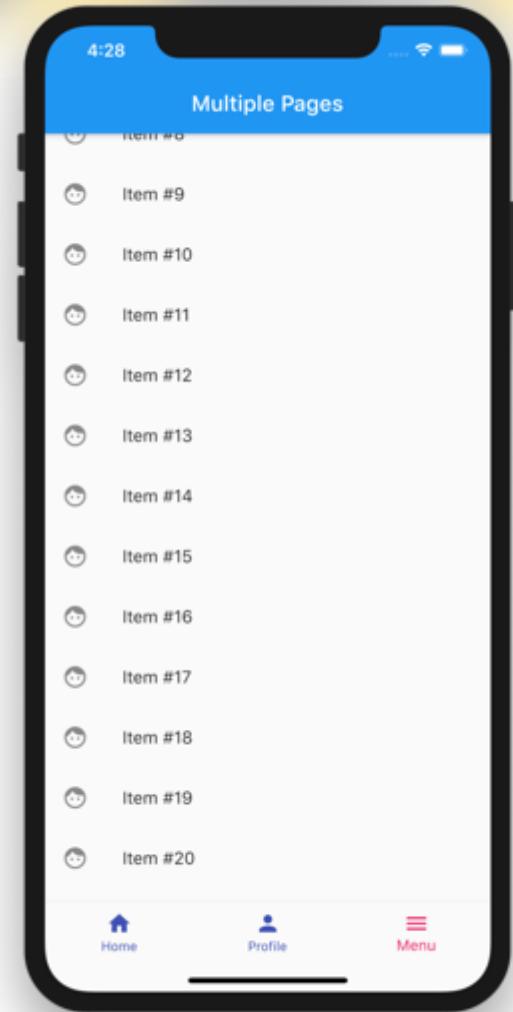
iPhone 11 Pro Max — 13.2.2

```
109 class ProfileScreen extends StatefulWidget {  
110     @override  
111     _ProfileScreenState createState() => _ProfileScreenState();  
112 }  
113  
114 class _ProfileScreenState extends State<ProfileScreen>  
115     with AutomaticKeepAliveClientMixin {  
116     @override  
117     bool get wantKeepAlive => true;  
118  
119     @override  
120     Widget build(BuildContext context) {  
121         return Scaffold(  
122             body: Center(  
123                 child: Icon(  
124                     Icons.person,  
125                     size: 50,  
126                     ),  
127                 ),  
128             );  
129     }  
130 }
```



iPhone 11 Pro Max — 13.2.2

```
132 class MenuScreen extends StatefulWidget {  
133     @override  
134     _MenuScreenState createState() => _MenuScreenState();  
135 }  
136  
137 class _MenuScreenState extends State<MenuScreen>  
138     with AutomaticKeepAliveClientMixin {  
139         @override  
140         bool get wantKeepAlive => true;  
141  
142         @override  
143         Widget build(BuildContext context) {  
144             return Scaffold(  
145                 body: ListView.builder(  
146                     itemCount: 100,  
147                     itemBuilder: (context, index) {  
148                         return ListTile(  
149                             leading: Icon(Icons.face),  
150                             title: Text("Item #${index}"),  
151                         );  
152                     },  
153                 ));  
154 }  
155 }  
156 }
```



iPhone 11 Pro Max — 13.2.2

របៀបប្រាប់ចេញពី main page ទៅកាន់ method ពីក្នុង Screen

- នៅក្នុង Screen និមួយា យើងត្រូវប្រកាស global variable មួយ ប្រភេទជារថា GlobalKey។ ហើយនៅក្នុង Screen យើងត្រូវ assign តំណែល global variable ទៅអាយុយ variable ពិសេសរបស់វា ឬឱ្យបាន key
- ចំណោកដក្នុង main page វិញ គឺយើងអាចប្រាប់ method ណាមួយនៅក្នុង screen មកប្រើបាន តាមរយៈ global variable នេះ
- បញ្ជាក់ method ក្នុង Screen ត្រូវតែប្រកាសជារថា public។

▪ ជាក់ key ត្រូវ Screen:

```
140     var menuScreenKey = GlobalKey<_MenuScreenState>();
141
142     class MenuScreen extends StatefulWidget {
143
144         final Key key = menuScreenKey;
145
146         @override
147         _MenuScreenState createState() => _MenuScreenState();
148     }
149
150     class _MenuScreenState extends State<MenuScreen>
151         with AutomaticKeepAliveClientMixin {
152             @override
153             bool get wantKeepAlive => true;
154
155             ScrollController _scroller = ScrollController();
156
157             gotoTop() {
158                 _scroller.animateTo(
159                     0,
160                     duration: Duration(milliseconds: 300),
161                     curve: Curves.easeOut,
162                 );
163             }
164         }
```

```
165     @override
166     Widget build(BuildContext context) {
167         return Scaffold(
168             body: ListView.builder(
169                 controller: _scroller,
170                 itemCount: 100,
171                 itemBuilder: (context, index) {
172                     return ListTile(
173                         leading: Icon(Icons.face),
174                         title: Text("Item #${index}"),
175                     );
176                 },
177             );
178         }
179     }
180 }
```

- នៅក្នុង main page យើងអាចចូលរួមទៅ method បន្ថែម Screen បាន៖

```

9   class _MyBottomNavigationBarExampleState
10  extends State<MyBottomNavigationBarExample> {
11    List<Widget> _pageList;
12
13    HomeScreen _homeScreen = HomeScreen();
14    ProfileScreen _profileScreen = ProfileScreen();
15    MenuScreen _menuScreen = MenuScreen();
16
17    @override
18    void initState() {
19      super.initState();
20
21      _pageList = [_homeScreen, _profileScreen, _menuScreen];
22    }
23
24    @override
25    Widget build(BuildContext context) {
26      return Scaffold(
27        appBar: AppBar(
28          title: Text("Multiple Pages"),
29          actions: <Widget>[
30            IconButton(
31              icon: Icon(Icons.keyboard_arrow_up),
32              onPressed: () {
33                menuScreenKey.currentState.gotoTop();
34              },
35            ),
36            ],
37          ),
38        ),
39      );
40    }
41  }

```

