

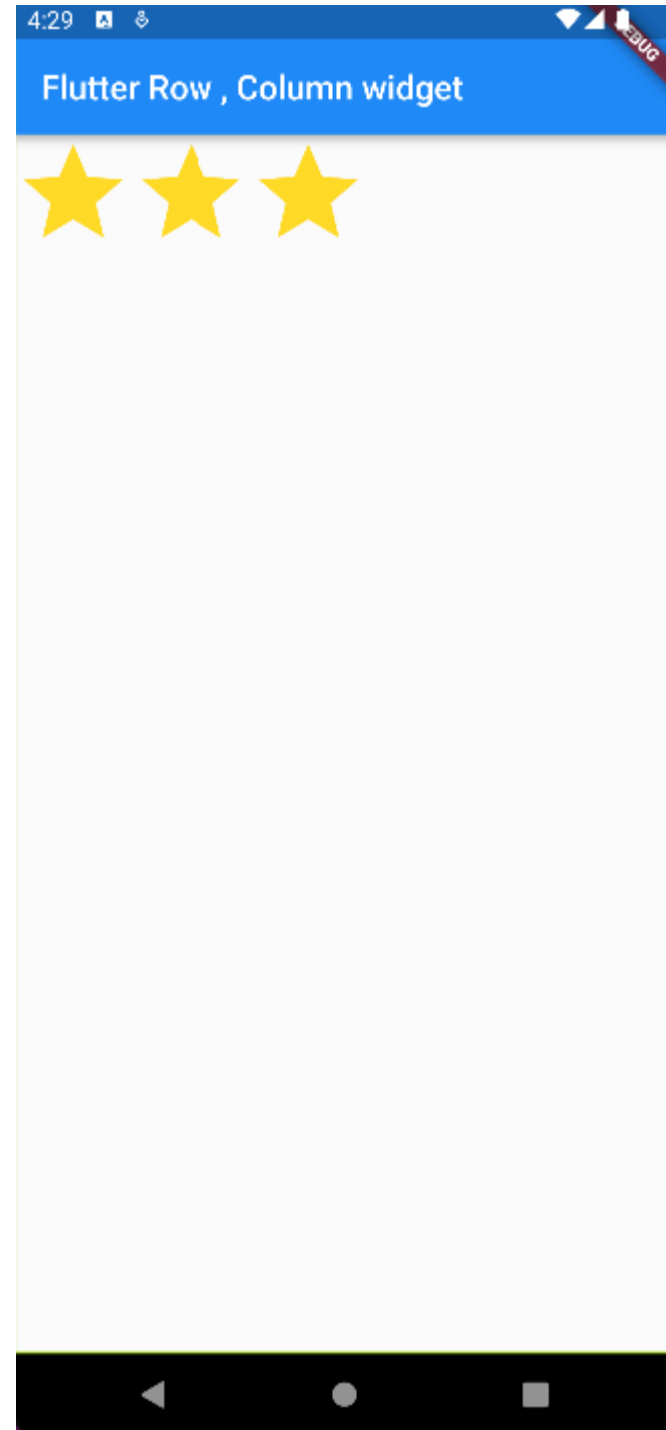
App Front 3주차 과제 풀이

발표자: (App General) 박시형

```
lookup.KeyValue  
f.constant(['en  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

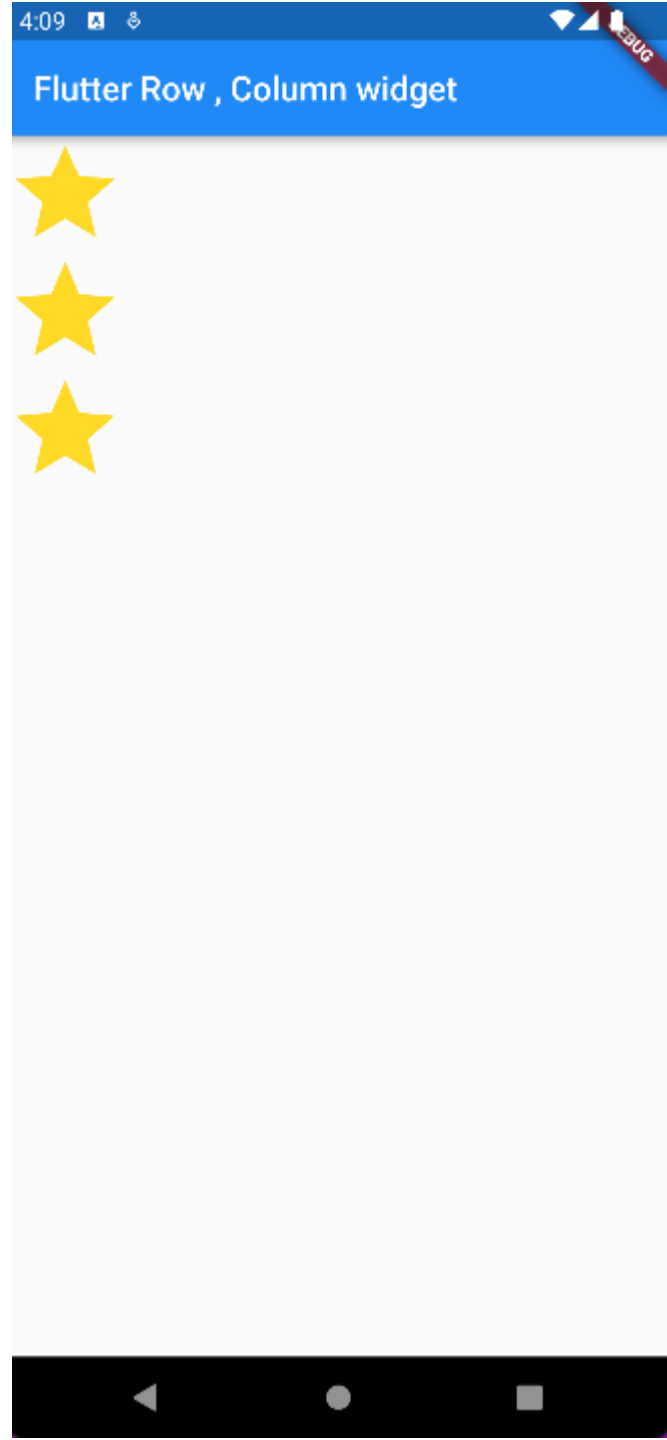
👉 Row Widget (행, 가로)

```
body: Container(  
  child: Row(  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
),
```



Column Widget (열,세로)

```
body: Column(  
  children: const [  
    Icon(Icons.star,size:70,color: Colors.amberAccent),  
    Icon(Icons.star,size:70,color: Colors.amberAccent),  
    Icon(Icons.star,size:70,color: Colors.amberAccent),  
  ],  
)
```



[3주차 과제]

`mainAxisAlignment` 과 `crossAxisAlignment` 는 Row 나 Column 에서 children 을 정렬하기 위해 사용합니다.

`mainAxisAlignment` 는 자식 widget을 주축에 따라 정렬하고,

`crossAxisAlignment` 는 자식 위젯을 주축에 수직으로 교차하는 교차 축을 따라 정렬합니다.

2. 문제

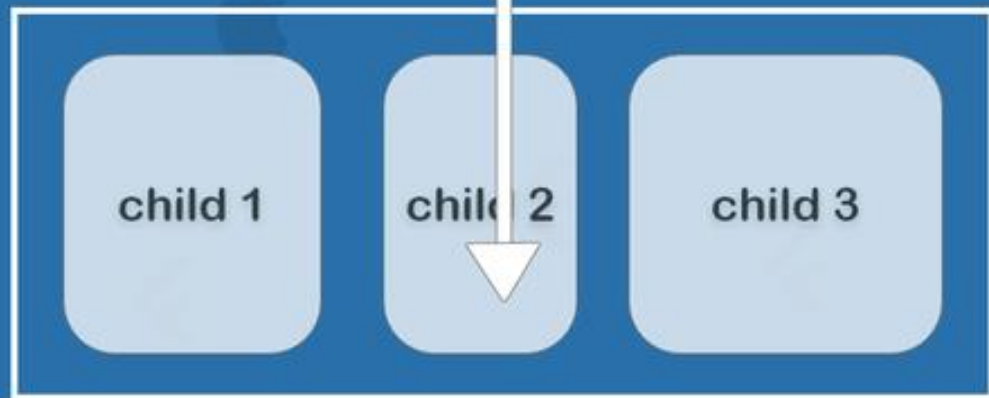
① 1번 문제

`crossAxisAlignment` 는 `start`, `end` 과 같이 총 5개의 값을 가질 수 있는데, 각 값이 어떻게 작동하는지 찾아 작성해주세요.

① 1번 답변

- `mainAxisAlignment`는 주축(mainAxis)가 수평(가로)라는 뜻이라는 점에서 위젯을 가로 방향으로 정렬하는 데 사용되는 속성이다. 행(Row) 위젯 사용시, 주축(mainAxis)은 수평(가로)이며, 횡축(crossAxis)는 세로이다. 열(Column)위젯 사용시, 주축(mainAxis)은 수직(세로)이며, 횡축(crossAxis)는 가로이다.
- `MainAxisAlignment.start`: 자식 위젯을 가로 방향의 시작 부분에 정렬함. 시작 부분은 주로 왼쪽 (LTR - 왼쪽에서 오른쪽) 또는 오른쪽 (RTL - 오른쪽에서 왼쪽)에 해당함.
- `MainAxisAlignment.end`: 자식 위젯을 가로 방향의 끝 부분에 정렬. 끝 부분은 주로 오른쪽 (LTR) 또는 왼쪽 (RTL)에 해당
- `MainAxisAlignment.center`: 자식 위젯을 가로 방향의 중앙에 정렬
- `MainAxisAlignment.spaceBetween`: 자식 위젯 사이에 동일한 간격을 두며, 첫 번째와 마지막 위젯은 시작 및 끝 부분에 정렬됨.
- `MainAxisAlignment.spaceAround`: 자식 위젯 주위에 동일한 간격을 두며, 시작과 끝 부분에도 반 간격을 둠.
- `MainAxisAlignment.spaceBetween`: 자식 위젯을 가능한 한 큰 간격을 두며 시작과 끝 부분은 그에 맞게 정렬됨.
- `MainAxisAlignment.spaceEvenly`: 자식 위젯을 가능한 한 큰 간격을 두며, 모든 위젯 주위에 동일한 간격을 둠.

crossAxisAlignment



mainAxisAlignment

〈 행(Row) 위젯 〉

주축(mainAxis): 수평(가로)

횡축(crossAxis): 세로

mainAxisAlignment

**crossAxis
Alignment**



< 열(Column)위젯 >

주축(mainAxis): 수직(세로)

횡축(crossAxis): 가로

MainAxisAlignment.center

Row(행) -> 가로축을 기준으로 가운데 정렬

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```



MainAxisAlignment.center

column(열) -> 세로축을 기준으로 가운데 정렬

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```

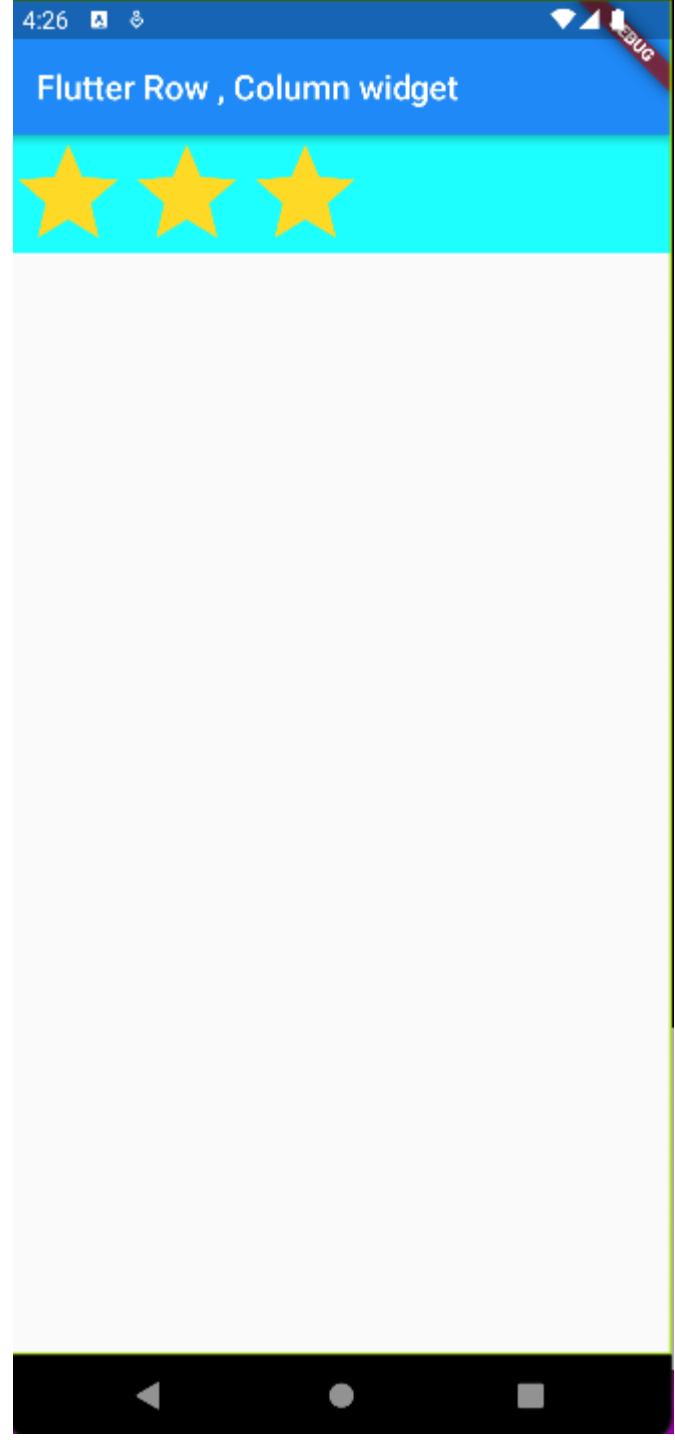
Flutter Row , Column widget



MainAxisAlignment.start

Row(행) -> 가로축을 기준으로 왼쪽으로 정렬

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.start,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```



MainAxisAlignment.start

column(열) -> 세로축을 기준으로 위쪽으로 정렬

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.start,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```

4:24

Flutter Row , Column widget



MainAxisAlignment.end

Row(행) -> 가로축을 기준으로 오른쪽으로 정렬

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.end,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```

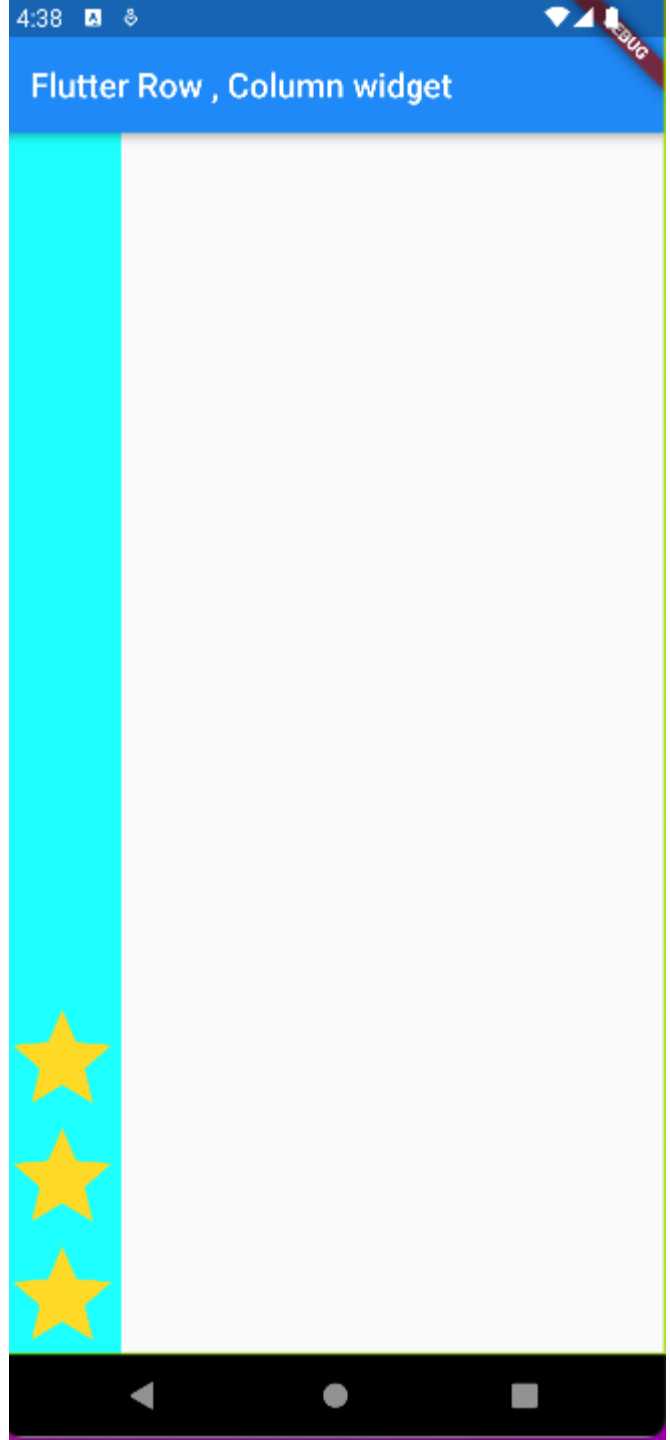
Flutter Row , Column widget



MainAxisAlignment.end

column(열) -> 세로축을 기준으로 아래쪽으로 정렬

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.end,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```

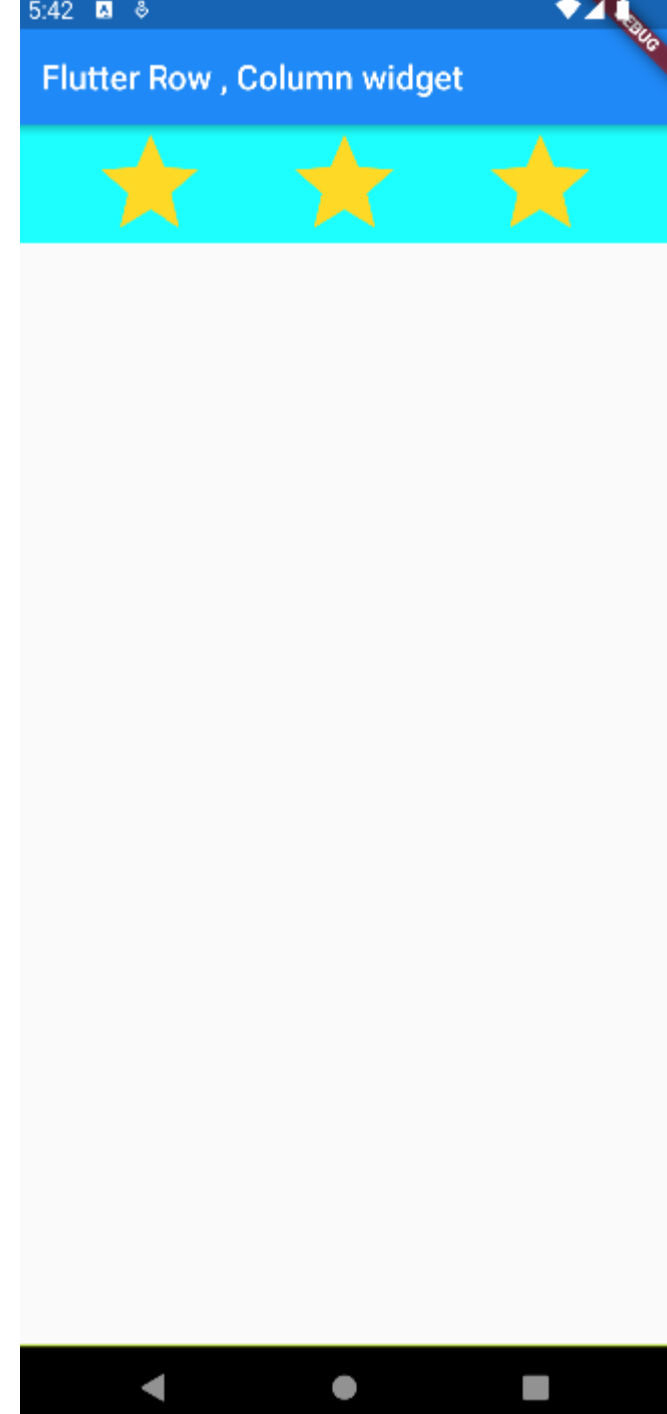


MainAxisAlignment.spaceEvenly

child 위젯 간의 공간을 두고, 같은 크기만큼 양 옆에 공간을 둔다.

Row(행)

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
),
```



MainAxisAlignment.spaceEvenly

column(열)

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```

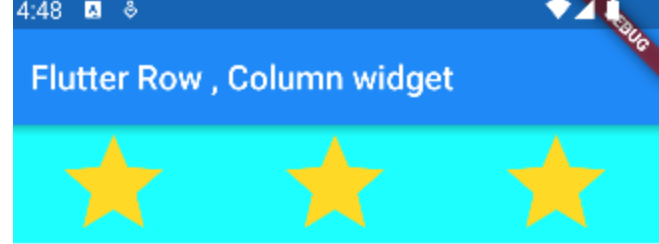


MainAxisAlignment.spaceAround

spaceEvenly와 비슷하지만, child 위젯간의 공간을 두고, 절반 크기만큼 양 옆에 공간을 둔다.

Row(행)

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.spaceAround,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
),
```



MainAxisAlignment.spaceAround

column(열)

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.spaceAround,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```

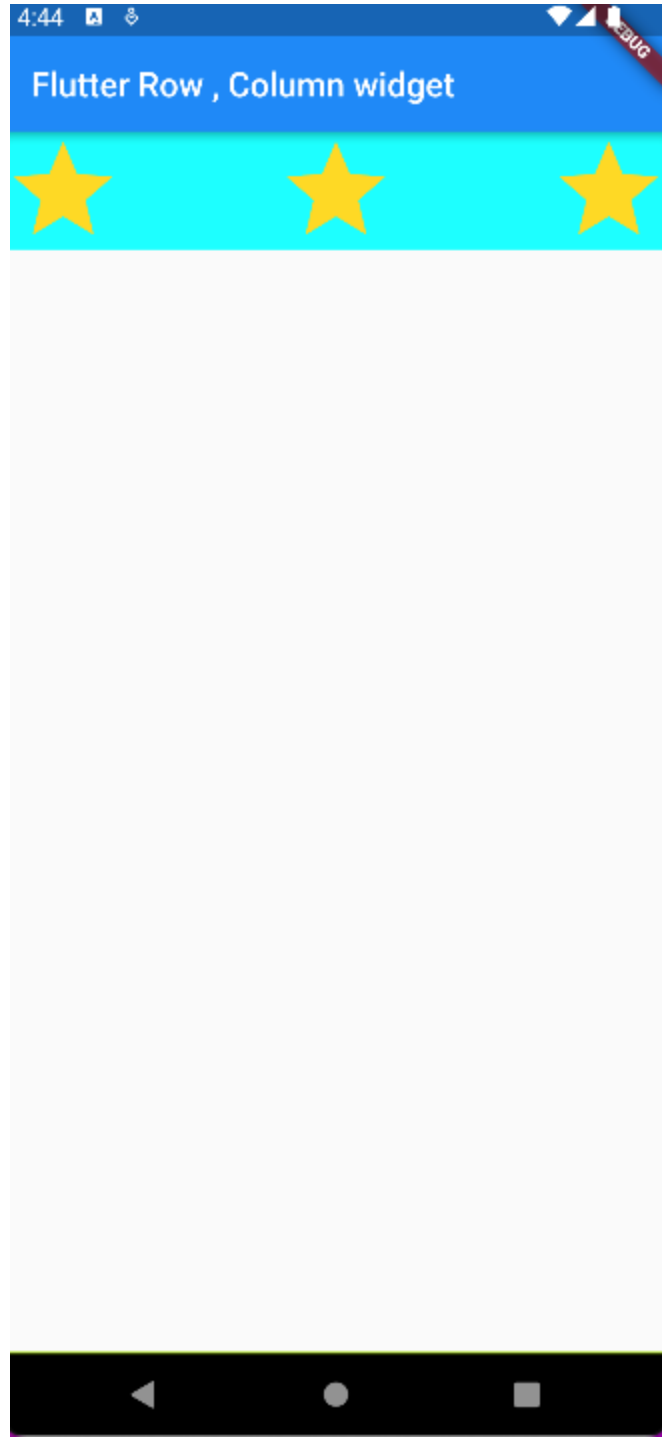


MainAxisAlignment.spaceBetween

child 위젯간의 공간을 두지만, spaceEvenly와 spaceAround와는 다르게 양 옆에 공간을 만들지 않는다.

Row(행)

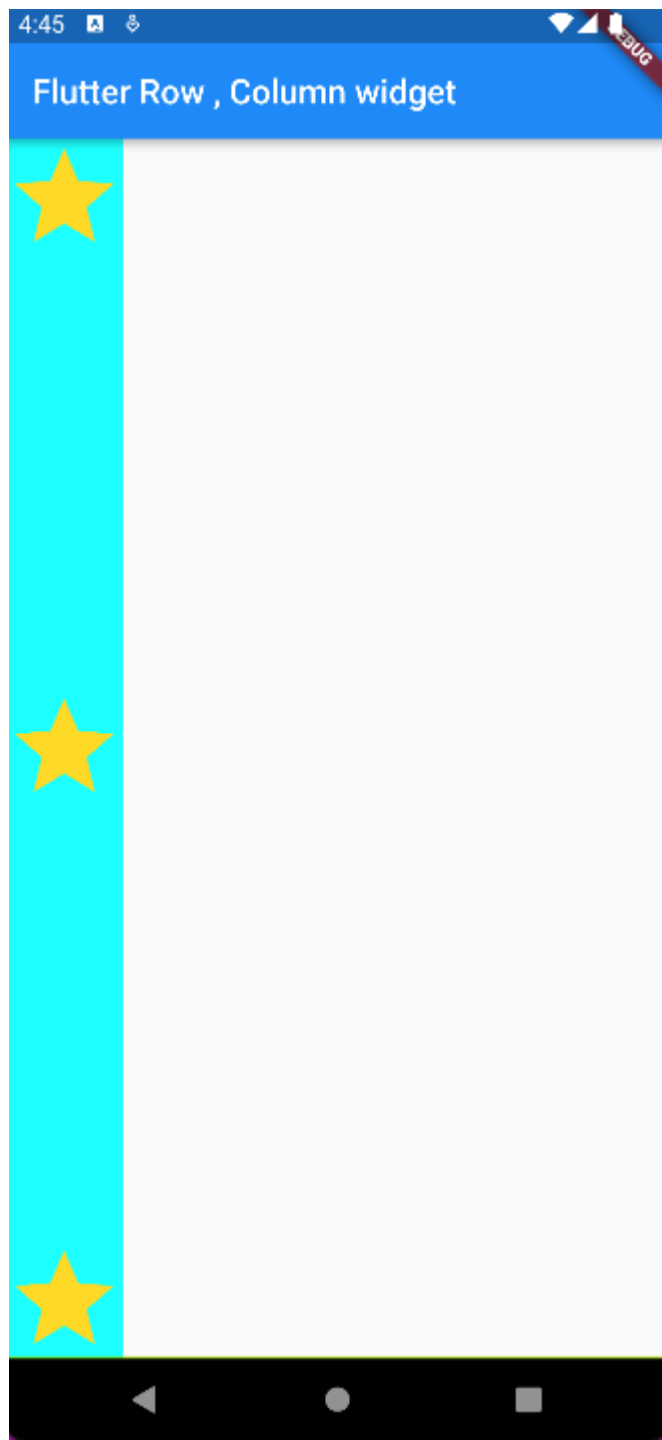
```
body: Container(  
  color: Colors.cyanAccent,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.spaceBetween,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
)
```



MainAxisAlignment.spaceBetween

column()

```
body: Container(  
  color: Colors.cyanAccent,  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.spaceBetween,  
    children: const [  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
      Icon(Icons.star, size: 70, color: Colors.amberAccent),  
    ],  
  ),  
),
```



2. 문제

② 2번 문제

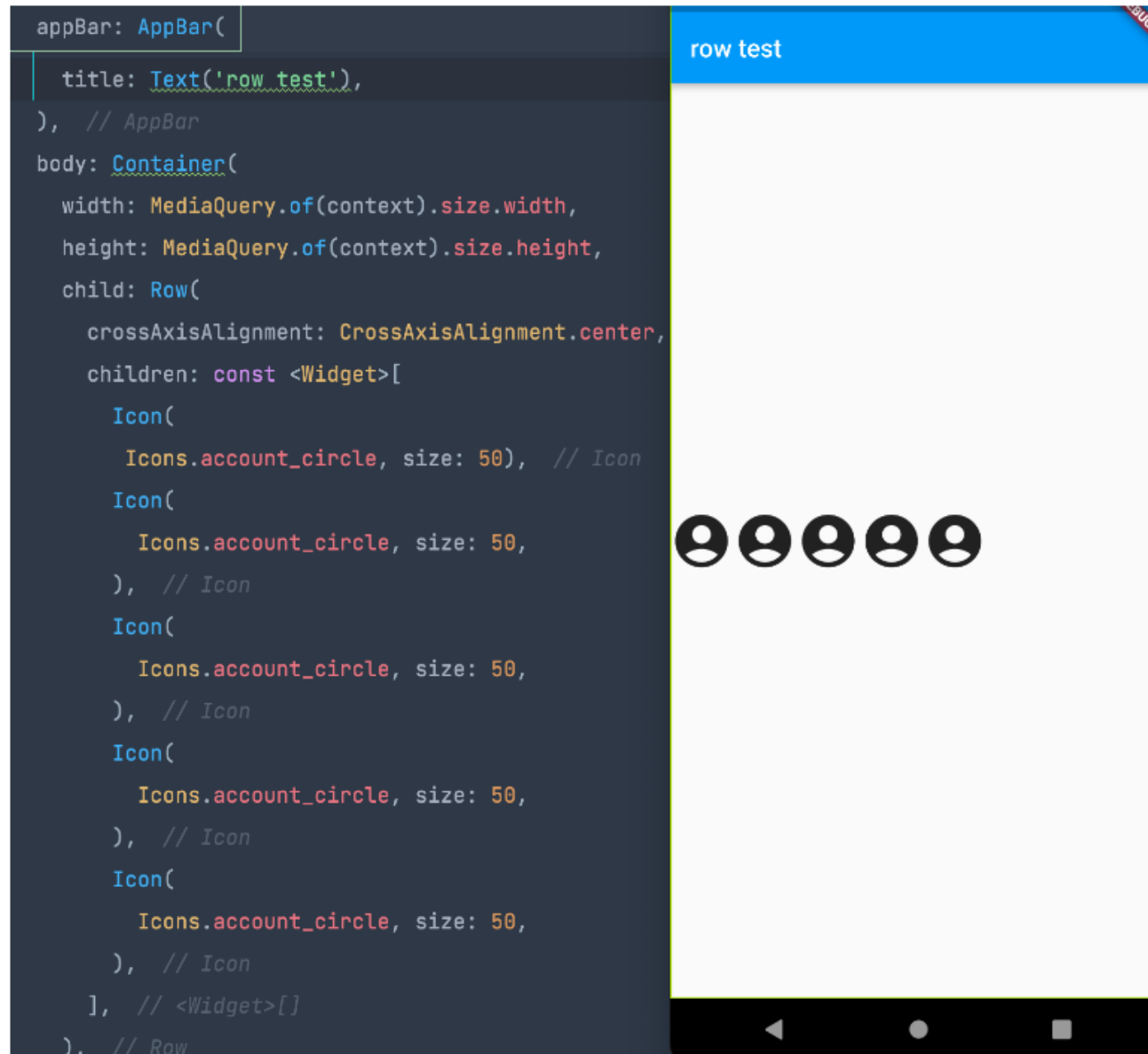
`crossAxisAlignment` 는 `start`, `end` 과 같이 총 5개의 값을 가질 수 있는데, 각 값이 어떻게 작동하는지 찾아 작성해주세요.

② 2번 답변

- `CrossAxisAlignment`는 횡축(`crossAxis`)가 세로라는 뜻이라는 점에서 위젯을 세로 방향으로 정렬하는 데 사용되는 속성이다. 행(`Row`) 위젯 사용시, 주축(`mainAxis`)은 수평(가로)이며, 횡축(`crossAxis`)는 세로이다. 열(`Column`)위젯 사용시, 주축(`mainAxis`)은 수직(세로)이며, 횡축(`crossAxis`)는 가로이다.
- `CrossAxisAlignment.start`: 자식 위젯을 교차 축의 시작 부분에 정렬한다. `Column` 내에서 자식 위젯을 가장 위에 정렬하거나 `Row` 내에서 자식 위젯을 가장 왼쪽에 정렬한다.
- `CrossAxisAlignment.end`: 자식 위젯을 교차 축의 끝 부분에 정렬한다. `Column` 내에서 자식 위젯을 가장 아래에 정렬하거나 `Row` 내에서 자식 위젯을 가장 오른쪽에 정렬한다.
- `CrossAxisAlignment.center`: 자식 위젯을 교차 축의 중앙에 정렬한다.
- `CrossAxisAlignment.stretch`: 자식 위젯을 교차 축에서 확장하여 부모 위젯과 같은 크기로 만든다. 이를 통해 교차 축에서 자식 위젯의 크기를 확장 가능하며 주로 `Column` 내에서 수직으로 전체 공간을 차지하거나 `Row` 내에서 수평으로 전체 공간을 차지하는 데 사용된다.
- `CrossAxisAlignment.baseline`: 자식 위젯을 텍스트 기준선과 맞춘다. (텍스트 요소가 있는 위젯을 다룰 때 유용함)

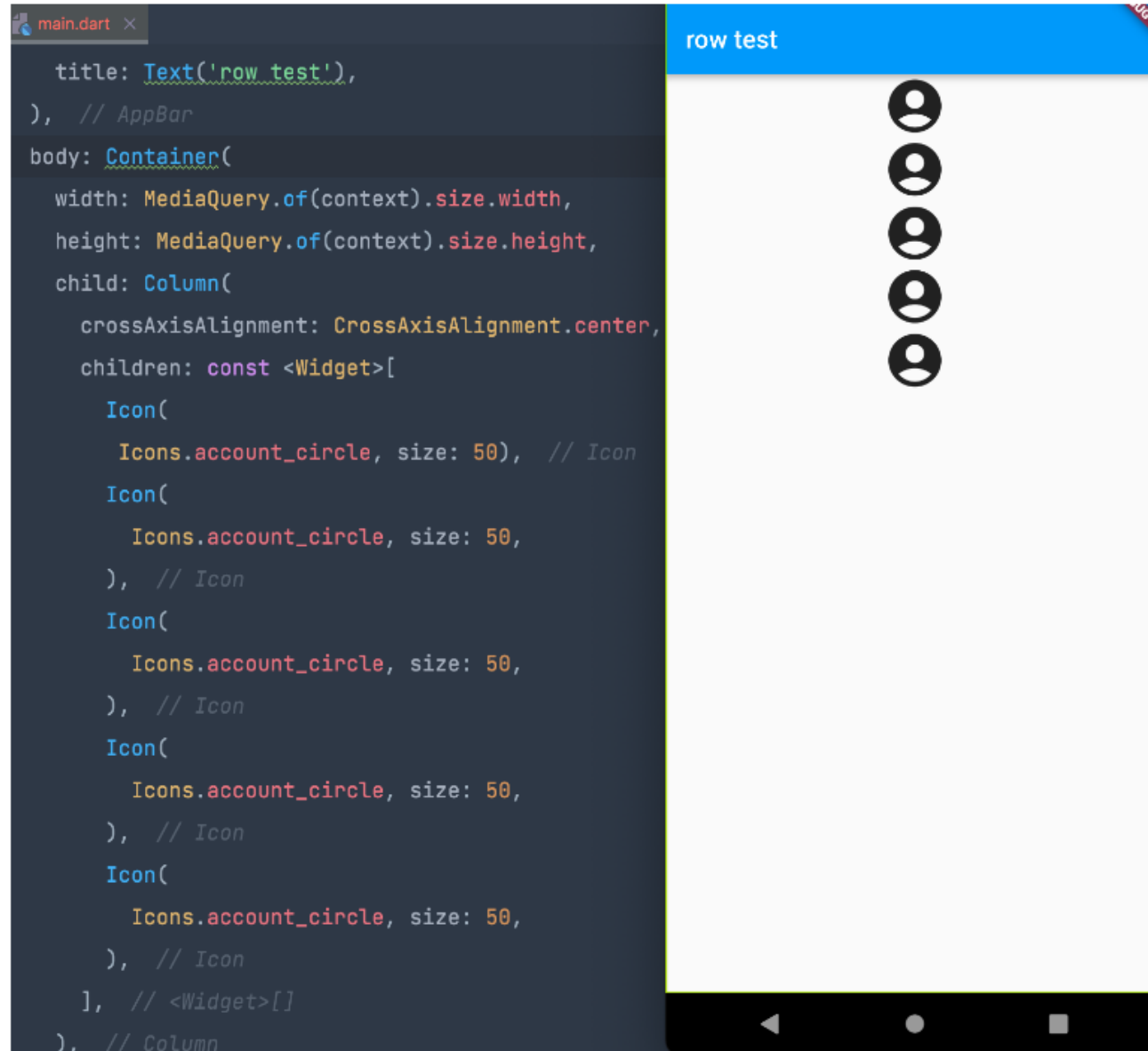
CrossAxisAlignment.center

Row(행)



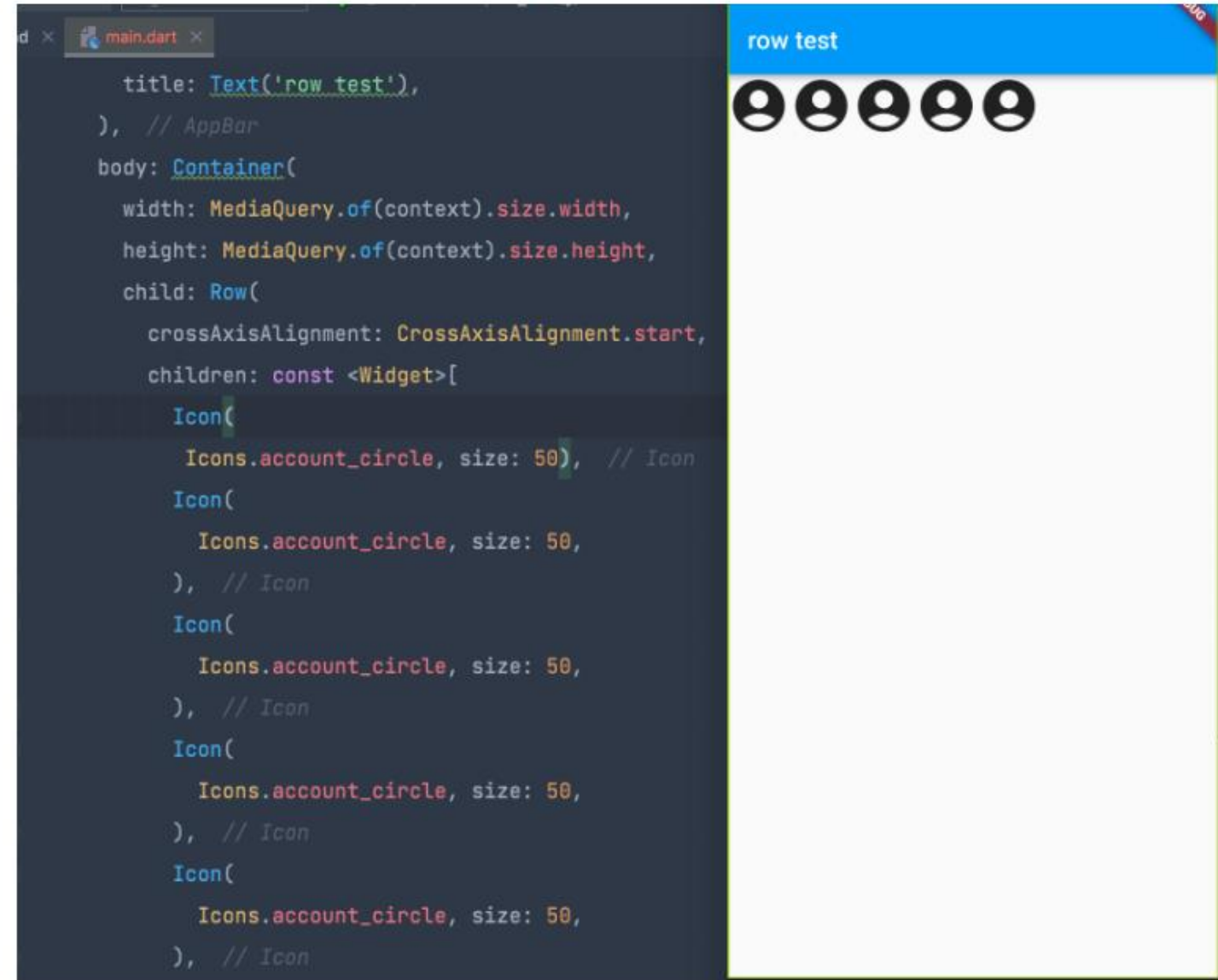
CrossAxisAlignment.center

column(열)



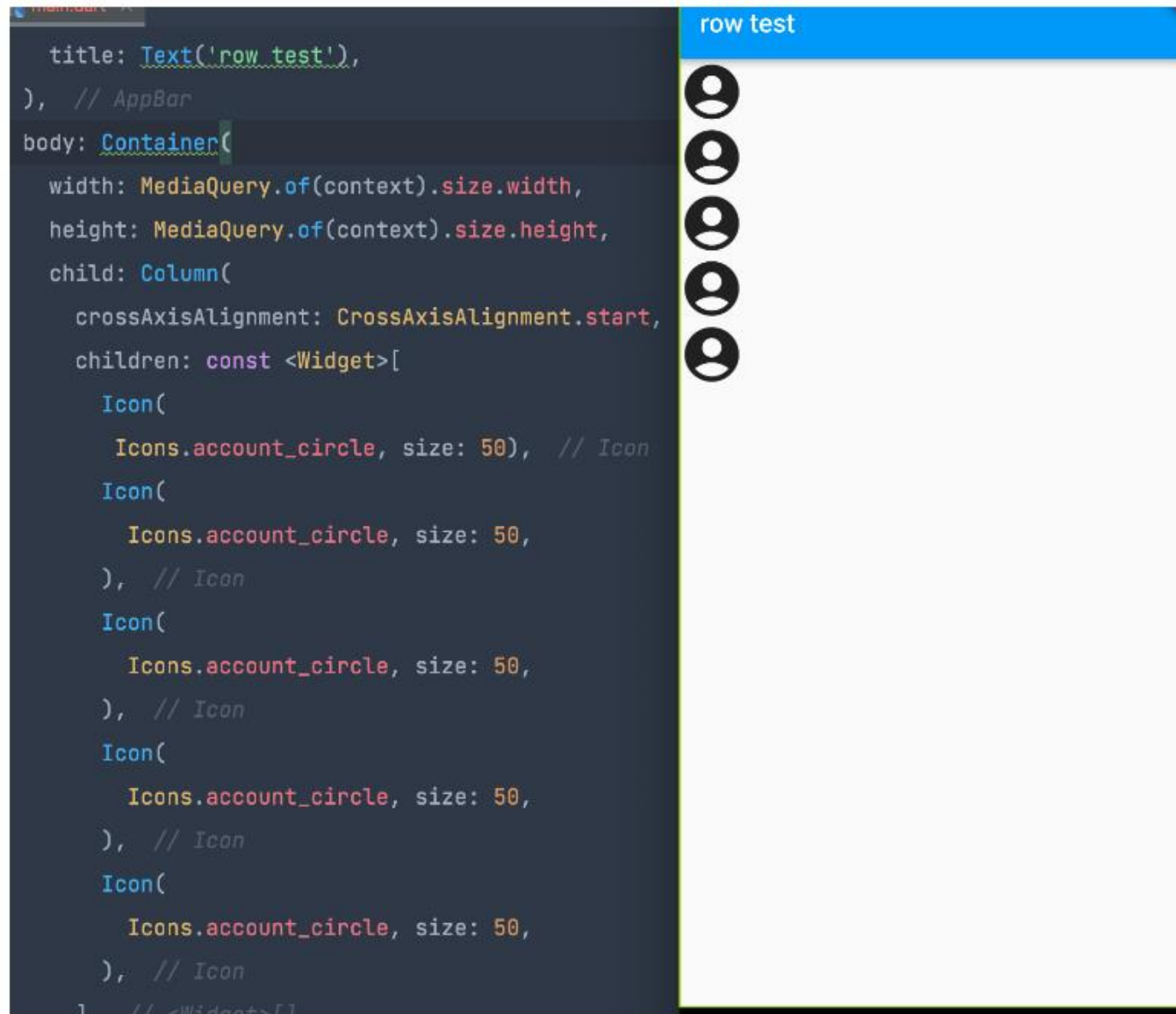
CrossAxisAlignment.start

Row(행)



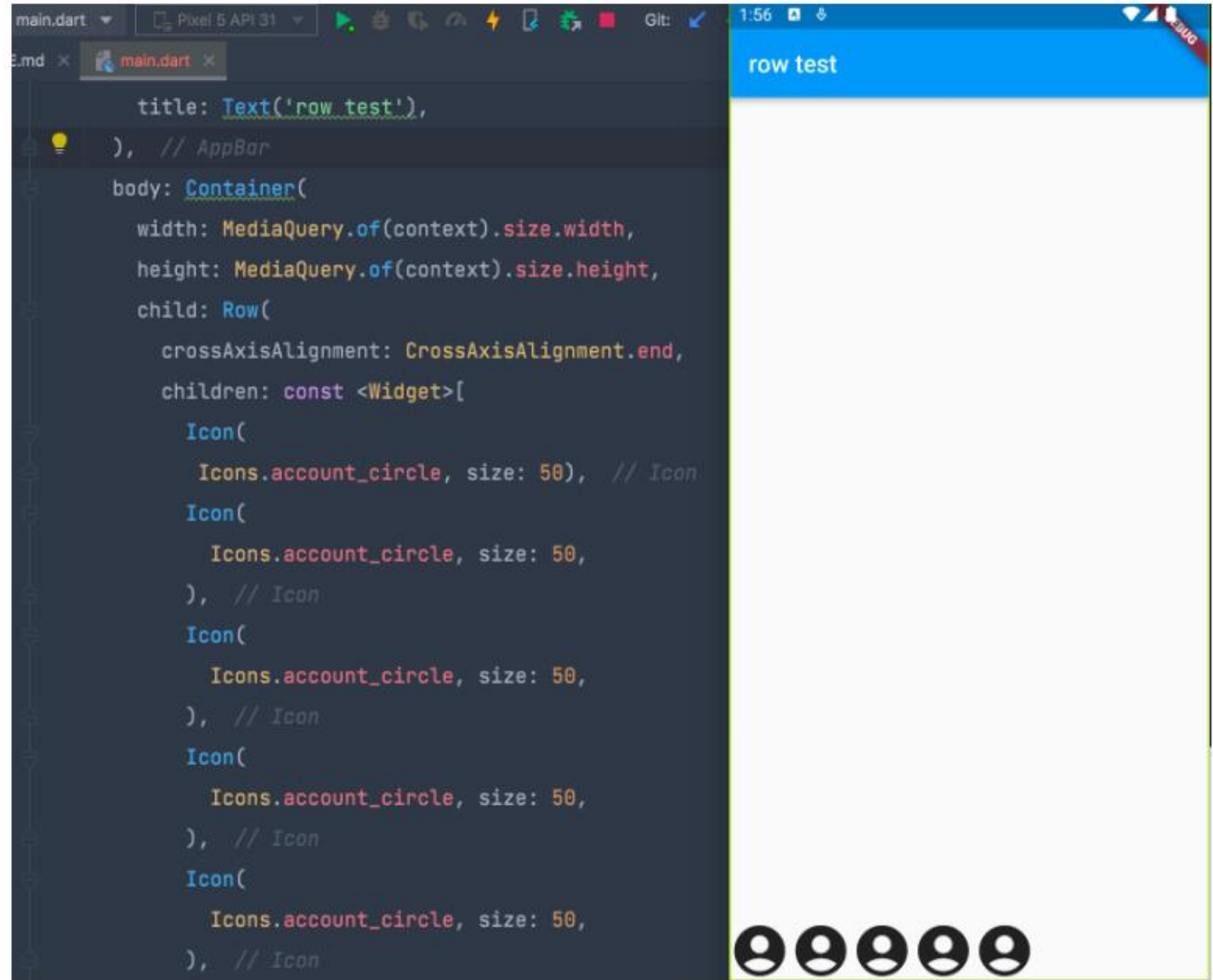
CrossAxisAlignment.start

column(열)



CrossAxisAlignment.end

Row(행)



CrossAxisAlignment.end

column(열)

```
main.dart x
title: Text('row test'),
), // AppBar
body: Container(
  width: MediaQuery.of(context).size.width,
  height: MediaQuery.of(context).size.height,
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.end,
    children: const <Widget>[
      Icon(
        Icons.account_circle, size: 50), // Icon
      Icon(
        Icons.account_circle, size: 50,
      ), // Icon
      Icon(
        Icons.account_circle, size: 50,
      ), // Icon
      Icon(
        Icons.account_circle, size: 50,
      ), // Icon
      Icon(
        Icons.account_circle, size: 50,
```

row test



2. 문제

③ 3번 문제

강의 #3.0 부터 #3.5까지 따라오셨다면, 완성되는 파일이 있을거예요!

자신이 작성한 flutter 파일의 lib 폴더를 github 자신의 이름에 해당하는 폴더에 upload 해주시면 됩니다

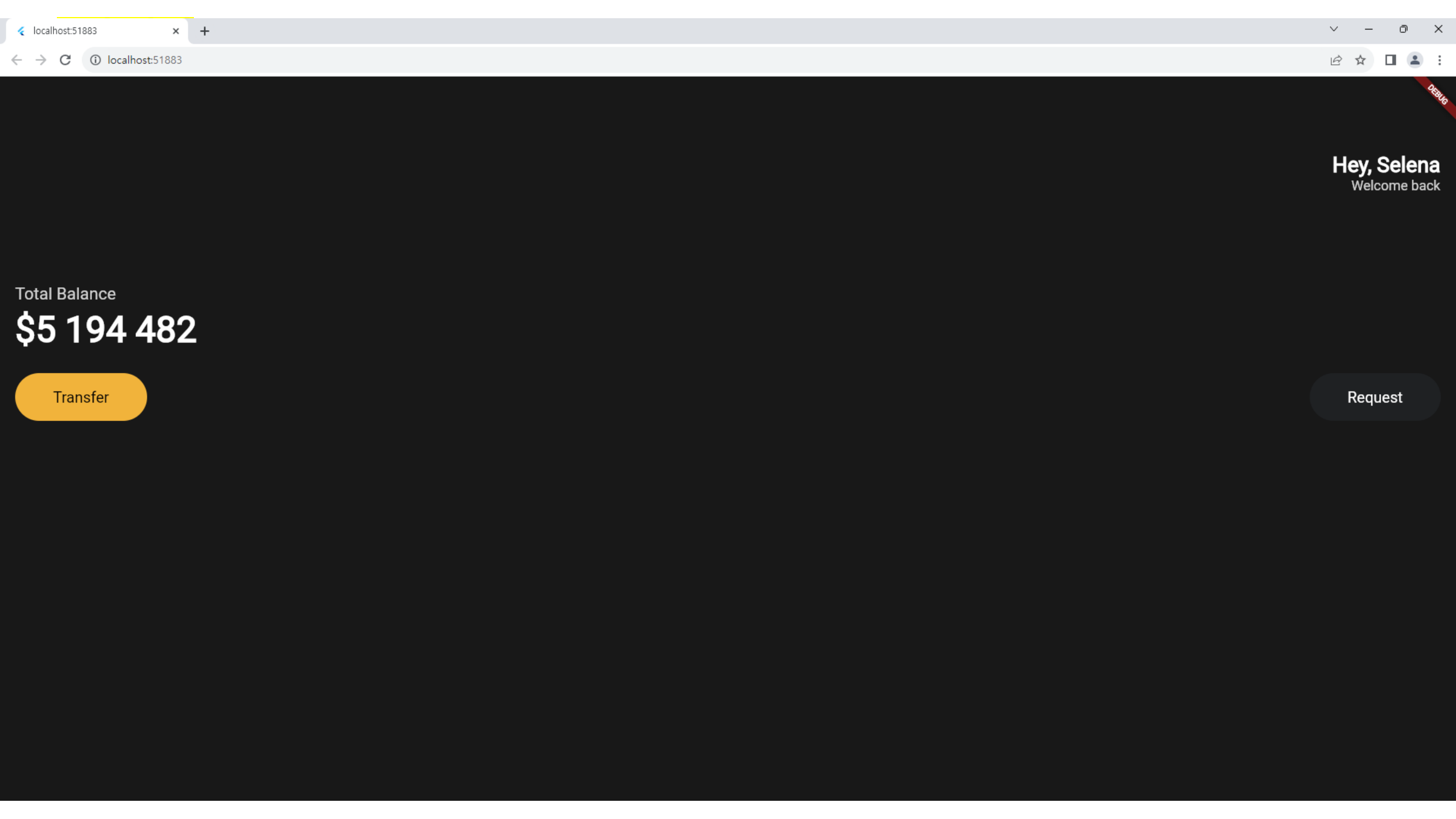
③ 3번 답변

```
main.dart X
C: > flutter > toonflix > lib > main.dart
1 import 'package:flutter/material.dart';
2 import 'package:toonflix/widgets/button.dart';
3
4 void main() {
5   runApp(App());
6 }
7
8 class App extends StatelessWidget {
9   @override
10  Widget build(BuildContext context) {
11    return MaterialApp(
12      home: Scaffold(
13        backgroundColor: const Color(0xFF181818),
14        body: Padding(
15          padding: const EdgeInsets.symmetric(
16            horizontal: 20, // 중복된 horizontal 속성을 합침
17            vertical: 20, // 추가된 vertical 속성
18          ),
19          child: Column(
20            crossAxisAlignment: CrossAxisAlignment.start,
21            children: [
22              const SizedBox(
23                height: 80,
24              ),
25              Row(
26                mainAxisAlignment: MainAxisAlignment.end,
27                children: [
28                  Column(
29                    crossAxisAlignment: CrossAxisAlignment.end,
30                    children: const [
31                      Text(
32                        'Hey, Selena',
33                        style: TextStyle(
34                          color: Colors.white,
35                          fontSize: 28,
36                          fontWeight: FontWeight.w800,
37                        ),
38                      ),
39                      Text(
40                        'Welcome back',
41                        style: TextStyle(
42                          color: Color.fromRGBO(255, 255, 255, 0.8),
43                          fontSize: 18,
44                        ),
45                      ),
46                    ],
47                  ),
48                ],
49              ),
50            ],
51          ),
52        ),
53      ),
54    );
55  }
56 }
```

```
main.dart X
C: > flutter > toonflix > lib > main.dart > App > build
47 ], // Column
48 ],
49 ), // Row
50 const SizedBox(
51   height: 120,
52 ), // SizedBox
53 Text(
54   'Total Balance',
55   style: TextStyle(
56     fontSize: 22,
57     color: Colors.white.withOpacity(0.8),
58   ), // TextStyle
59 ), // Text
60 const SizedBox(
61   height: 5,
62 ), // SizedBox
63 const Text(
64   '\$5 194 482',
65   style: TextStyle(
66     fontSize: 48,
67     fontWeight: FontWeight.w600,
68     color: Colors.white,
69   ), // TextStyle
70 ), // Text
71 const SizedBox(
72   height: 30,
73 ), // SizedBox
74 Row(
75   mainAxisAlignment: MainAxisAlignment.spaceBetween,
76   children: [
77     Button(
78       text: 'Transfer',
79       bgColor: const Color(0xFFFF1B3B),
80       textColor: Colors.black,
81     ), // Button
82     Button(
83       text: 'Request',
84       bgColor: const Color(0xFF1F2123),
85       textColor: Colors.white,
86     ), // Button
87   ],
88 ), // Row
89 ],
90 ), // Column
91 ), // Padding
92 ), // Scaffold
93 ); // MaterialApp
94 }
```

③ 3번 답변

```
main.dart button.dart X
C: > flutter > toonflix > lib > widgets > button.dart > ...
1 import 'package:flutter/material.dart';
2
3 class Button extends StatelessWidget {
4   final String text;
5   final Color bgColor;
6   final Color textColor;
7
8   const Button({
9     super.key,
10    required this.text,
11    required this.bgColor,
12    required this.textColor,
13  });
14
15  @override
16  Widget build(BuildContext context) {
17    return Container(
18      decoration: BoxDecoration(
19        color: bgColor,
20        borderRadius: BorderRadius.circular(45),
21      ), // BoxDecoration
22      child: Padding(
23        padding: const EdgeInsets.symmetric(
24          vertical: 20,
25          horizontal: 50,
26        ), // EdgeInsets.symmetric
27        child: Text(
28          text,
29          style: TextStyle(
30            color: textColor,
31            fontSize: 20,
32          ), // TextStyle
33        ), // Text
34      ), // Padding
35    ); // Container
36  }
37 }
```



Hey, Selena
Welcome back

Total Balance
\$5 194 482

Transfer

Request

Debug

감사합니다

T h a n k y o u