

04

# 배치 – Row, Column

기본 위젯

# Row, Column

---

## 가로로 배치하기 - Row

- Row는 가로 방향으로 위젯을 배치

• 가로 방향으로 배치하기

```
Row(  
  children: [  
    Container(  
      width: 100,  
      height: 100,  
      color: Colors.red,  
    ),  
    Container(  
      width: 100,  
      height: 100,  
      color: Colors.blue,  
    ),  
  ],  
)
```

▶ 실행 결과

A screenshot of a mobile application interface. At the top, there is a status bar with the time 7:37 and various icons. Below the status bar, there is a blue header bar with the text 'Test'. The main content area of the app shows two colored squares side-by-side: a red square on the left and a blue square on the right. Both squares are 100x100 pixels in size.

# Row, Column

---

## 세로로 배치하기 - Column

- Column에 추가하면 다음처럼 세로로 배치

• 세로 방향으로 배치하기

```
Column(  
  children: [  
    Container(  
      width: 100,  
      height: 100,  
      color: Colors.red,  
    ),  
    Container(  
      width: 100,  
      height: 100,  
      color: Colors.blue,  
    ),  
  ],  
)
```

▶ 실행 결과



# Row, Column

## 레이아웃 중첩하기



그림 10-1 인스타그램 같은 화면

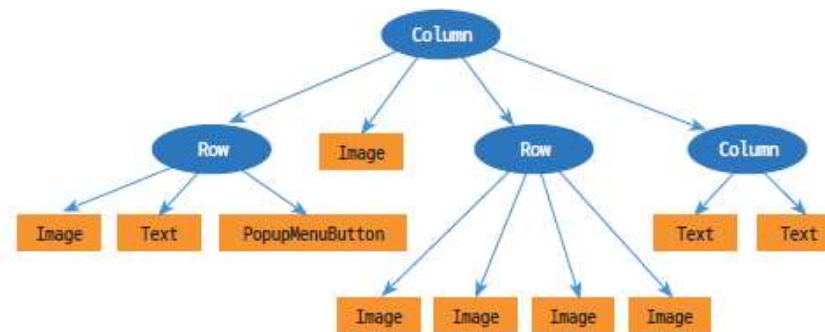


그림 10-2 레이아웃 중첩

# Row, Column

## 레이아웃 중첩하기

### • 레이아웃 중첩

```
Column(  
  children: [  
    Row(  
      children: [  
        Image(... (생략) ...),  
        Text(... (생략) ...),  
        PopupMenuButton(... (생략) ...)  
      ],  
    ),  
    Image(... (생략) ...),  
    Row(  
      children: [  
        Image(... (생략) ...),  
        Image(... (생략) ...),  
        Image(... (생략) ...),  
        Image(... (생략) ...),  
      ],  
    ),  
    Column(  
      children: [  
        Text(... (생략) ...),  
        Text(... (생략) ...),  
      ],  
    ),  
  ],  
),
```

# Row, Column

## 크기 설정하기 - `mainAxisSize`

- Row와 Column은 기본축main axis과 교차축cross axis이라는 2개의 축
- Row는 위젯을 가로로 배치하므로 가로 방향이 기본축이며 세로 방향이 교차축
- Column은 위젯을 세로로 배치하므로 세로 방향이 기본축이며 가로 방향이 교차축

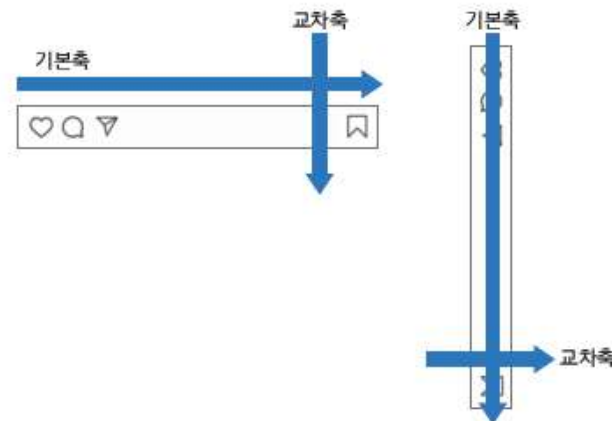


그림 10-3 기본축과 교차축

# Row, Column

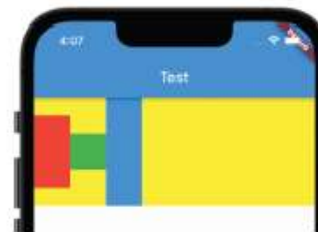
## 크기 설정하기 - `mainAxisSize`

- Row나 Column에 크기를 설정하지 않으면 기본축 크기는 차지할 수 있는 최대 크기로 설정
- 교차축은 추가된 위젯을 출력할 최소 크기로 설정

• 영역 확인하기

```
Container(  
  color: Colors.yellow,  
  child: Row(  
    children: [  
      Container(  
        width: 50, height: 100, color: Colors.red,  
      ),  
      Container(  
        width: 50, height: 50, color: Colors.green,  
      ),  
      Container(  
        width: 50, height: 150, color: Colors.blue,  
      ),  
    ],  
  ),  
)
```

▶ 실행 결과



# Row, Column

---

## 크기 설정하기 - `mainAxisSize`

- `mainAxisSize`에는 `MainAxisSize.max`나 `MainAxisSize.min`값을 설정





# Row, Column

---

## 배치 설정하기 - Alignment

- 배치를 설정하지 않으면 기본축은 start이며 교차축은 center
- MainAxisAlignment 클래스의 상수
  - •center: 중앙에 배치
  - •end: 끝에 배치
  - •start: 시작에 배치
  - •spaceAround: 각 위젯의 앞뒤 공백을 균등하게 배치
  - •spaceBetween: 위젯 간 공백을 균등하게 배치
  - •spaceEvenly: 앞뒤 그리고 각 위젯 간 공백을 균등하게 배치

# Row, Column

## 배치 설정하기 - Alignment

- CrossAxisAlignment 클래스의 상수
  - •baseline: 기준선에 맞춰 배치
  - •center: 가운데에 배치
  - •end: 끝에 배치
  - •start: 시작에 배치
  - •stretch: 교차축을 모두 차지하게 배치

### • 기본축과 교차축 배치 설정하기

```
Container(  
  color: Colors.yellow,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [  
      Container(width: 50, height: 100, color: Colors.red,),  
      Container(width: 50, height: 50, color: Colors.green,),  
      Container(width: 50, height: 150, color: Colors.blue,),  
    ],  
  ),  
)
```

▶ 실행 결과



# Row, Column

## 배치 설정하기 - Alignment

- spaceBetween은 각 위젯 사이의 공백을 균등한 크기로 설정
- spaceAround는 각 위젯의 앞뒤 공백을 균등한 크기로 설정
- spaceEvenly는 전체 위젯의 앞뒤 공백과 각 위젯 사이의 공백을 균등한 크기로 설정

### • 기본축 배치 설정하기

```
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceBetween,  
  children: [  
    Container(width: 50, height: 50, color: Colors.red,),  
    Container(width: 50, height: 50, color: Colors.green,),  
    Container(width: 50, height: 50, color: Colors.blue,),  
  ],  
)  
  
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceAround,  
  children: [  
    Container(width: 50, height: 50, color: Colors.red,),  
    Container(width: 50, height: 50, color: Colors.green,),  
    Container(width: 50, height: 50, color: Colors.blue,),  
  ],  
)  
  
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  children: [  
    Container(width: 50, height: 50, color: Colors.red,),  
    Container(width: 50, height: 50, color: Colors.green,),  
    Container(width: 50, height: 50, color: Colors.blue,),  
  ],  
)
```

▶ 실행 결과



# Row, Column

## 배치 설정하기 - Alignment

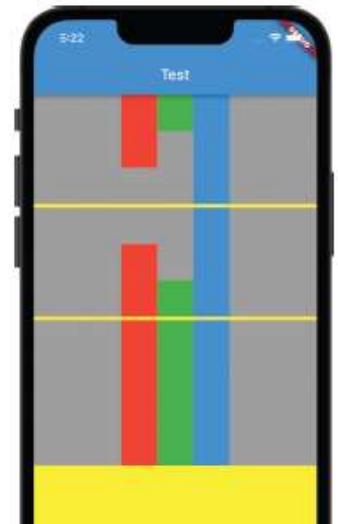
- `CrossAxisAlignment.start`를 설정하면 위젯을 교차축의 시작 위치에 배치
- `CrossAxisAlignment.end`를 설정하면 끝 위치에 배치
- `CrossAxisAlignment.stretch`는 각 위젯에 설정된 교차축 크기를 무시하고 전체 공간을 차지

### • 교차축 배치 설정하기

```
Container(  
  margin: EdgeInsets.only(bottom: 5),  
  color: Colors.grey,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    crossAxisAlignment: CrossAxisAlignment.start,  
    children: [  
      ... (생략) ...  
    ],  
  ),  
)  
  
Container(  
  color: Colors.grey,  
  margin: EdgeInsets.only(bottom: 5),  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    crossAxisAlignment: CrossAxisAlignment.end,
```

```
    children: [  
      ... (생략) ...  
    ],  
  ),  
)  
  
Container(  
  color: Colors.grey,  
  margin: EdgeInsets.only(bottom: 5),  
  height: 200,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    crossAxisAlignment: CrossAxisAlignment.stretch,  
    children: [  
      ... (생략) ...  
    ],  
  ),  
)
```

### ▶ 실행 결과





# 감사합니다

단단히 마음먹고 떠난 사람은  
산꼭대기에 도착할 수 있다.  
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어  
William Shakespeare