

04

# 버튼과 이벤트

기본 위젯

# 아이콘과 아이콘 버튼

- 아이콘을 출력할 때는 Icon 위젯을 사용
- 아이콘을 지정해야 하는데 그 역할을 IconData 객체가 합니다
- 플러터 아이콘 모음: <https://api.flutter.dev/flutter/material/Icons-class.html>

```
static const IconData alarm = IconData(0xe072, fontFamily: 'MaterialIcons');
```

```
Icon(  
  IconData(0xe072, fontFamily: 'MaterialIcons'),  
  size: 100,  
  color: Colors.red,  
),
```

## • 알람 아이콘 출력하기

```
Icon(  
  Icons.alarm,  
  size: 100,  
  color: Colors.red  
),
```

## ▶ 실행 결과



## • Icon 생성자

```
const Icon(  
  IconData? icon,  
  { Key? key,  
    double? size,  
    Color? color,  
    String? semanticLabel,  
    TextDirection? textDirection }  
)
```

# 아이콘과 아이콘 버튼

---

## 폰트 어섬 아이콘 사용하기

- 폰트 어섬(fontawesome.com)에서 제공하는 아이콘은 FontAwesomelcons 패키지로 이용

```
• FontAwesomelcons 패키지 등록  
  
dependencies:  
  font_awesome_flutter: ^10.1.0
```

- FontAwesomelcons 패키지의 Falcon 위젯으로 사용

```
• 폰트어섬 아이콘 사용하기  
  
FaIcon(  
  FontAwesomeIcons.bell  
)
```

# 아이콘과 아이콘 버튼

---

## 아이콘으로 버튼 만들기 – IconButton

### • 아이콘 버튼 사용법

```
IconButton(  
  onPressed: onPressed,  
  icon: Icon(Icons.alarm)  
)
```

# 제스처 감지기와 엘리베이트 버튼

## 제스처 감지기 – GestureDetector

- GestureDetector 위젯은 사용자가 화면을 탭하거나 드래그하는 등의 행위를 감지해 특정 로직을 실행
- GestureDetector의 child에 화면 구성을 대입하고 이 화면에서 발생하는 사용자 이벤트를 처리

### • GestureDetector의 생성자

```
GestureDetector(  
  { Key? key,  
    Widget? child,  
    GestureTapDownCallback? onTapDown,  
    GestureTapUpCallback? onTapUp,  
    GestureTapCallback? onTap,  
    GestureTapCallback? onDoubleTap,  
    GestureLongPressCallback? onLongPress,  
    GestureDragStartCallback? onVerticalDragStart,  
    GestureDragEndCallback? onVerticalDragEnd,  
    GestureDragStartCallback? onHorizontalDragStart,  
    GestureDragEndCallback? onHorizontalDragEnd,  
    ... (생략) ...  
  }  
)
```

### • 사용자 이벤트 처리하기

```
GestureDetector(  
  child: Image.asset('images/icon/user.png'),  
  onTap: () {  
    print('image click...');  
  },  
)
```

# 제스처 감지기와 엘리베이트 버튼

## 제스처 감지기 – GestureDetector

- Drag와 관련된 이벤트인 onVerticalDragStart, onHorizontalDragStart 등의 이벤트 콜백 함수에는 매개변수로 DragStartDetails 객체가 전달
- DragStartDetails 클래스의 globalPosition 속성에는 단말기에서의 좌푯값이 전달되며 localPosition에는 위젯에서의 좌푯값이 전달

• 드래그 좌푯값 가져오기

```
GestureDetector(  
  child: Image.asset('images/icon/user.png'),  
  onVerticalDragStart: (DragStartDetails details) {  
    print(  
      'vertical drag start...global position : '  
        '${details.globalPosition.dx}, '  
        '${details.globalPosition.dy}');  
    print(  
      'vertical drag start...local position : '  
        '${details.localPosition.dx}, '  
        '${details.localPosition.dy}');  
  }  
)
```

# 제스처 감지기와 엘리베이트 버튼

---

## 엘리베이트 버튼 - ElevatedButton

- 버튼 모양을 제공하는 ElevatedButton

### • ElevatedButton 생성자

```
const ElevatedButton(  
  { Key? key,  
    required VoidCallback? onPressed,  
    ButtonStyle? style,  
    required Widget? Child  
    ... (생략) ...  
  }  
)
```

# 제스처 감지기와 엘리베이트 버튼

## 엘리베이트 버튼 - ElevatedButton

- onPressed 속성에 이벤트 콜백 함수를 등록하는데 만약 null이 대입되면 버튼이 비활성
- style 속성에 ButtonStyle을 적용해 배경색 등을 설정

• 이벤트 처리가 포함된 엘리베이트 버튼

```
ElevatedButton(  
  onPressed: () {  
    print('ElevatedButton click...');  
  },  
  child: Text('Click Me'),  
),  
ElevatedButton(  
  onPressed: null,  
  child: Text('Disabled Button'),  
),  
ElevatedButton(  
  onPressed: () {  
    print('ElevatedButton click...');  
  },  
  child: Text('Click Me'),  
  style: ButtonStyle(  
    backgroundColor: MaterialStateProperty.all<Color>(Colors.red)  
  ),  
)
```

▶ 실행 결과







# 감사합니다

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



윌리엄 셰익스피어  
William Shakespeare