

Do it! 플러터 앱 프로그래밍

11장 애니메이션 활용하기

목차

- 11-1 애니메이션 구현하기
- 11-2 애니메이션 세세하게 다루기

11-1 애니메이션 구현하기

플러터는 애니메이션을 구현하기 쉽게 구조화되어 있습니다

다양한 애니메이션 효과와 Curves를 제공합니다.

<https://api.flutter.dev/flutter/animation/Curves-class.html>

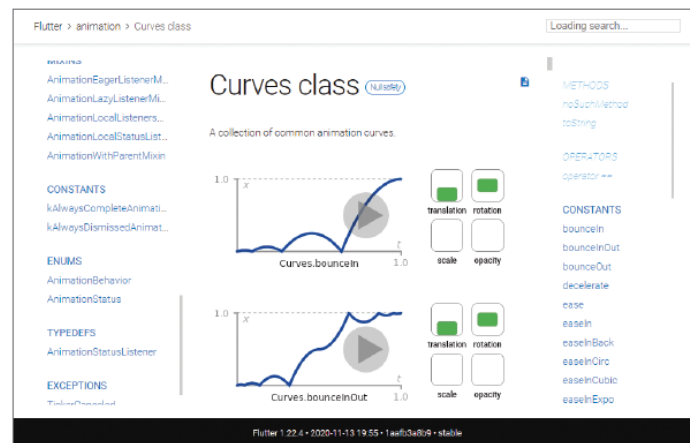
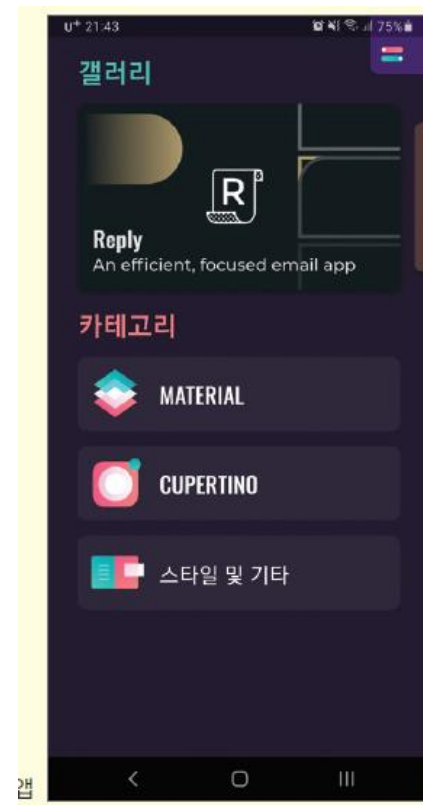


그림 11-1 Curves 클래스가 제공하는 다양한 애니메이션 모양



11-1 애니메이션 구현하기

AnimatedContainer 를 이용하여 애니메이션을 구현할수있음

Duration : 애니메이션 실행시간

Curve : 애니메이션 작동 계산식

```
AnimatedContainer(  
  duration: Duration(seconds: 2),  
  curve: Curves.bounceIn,  
  color: Colors.amber,  
  child: Text(  
    '키 ${peoples[current].height}',  
    textAlign: TextAlign.center,  
  ),  
  width: 50,  
  height: peoples[current].height,  
),
```

애니메이션 모양과
재생 시간 설정

11-1 애니메이션 구현하기

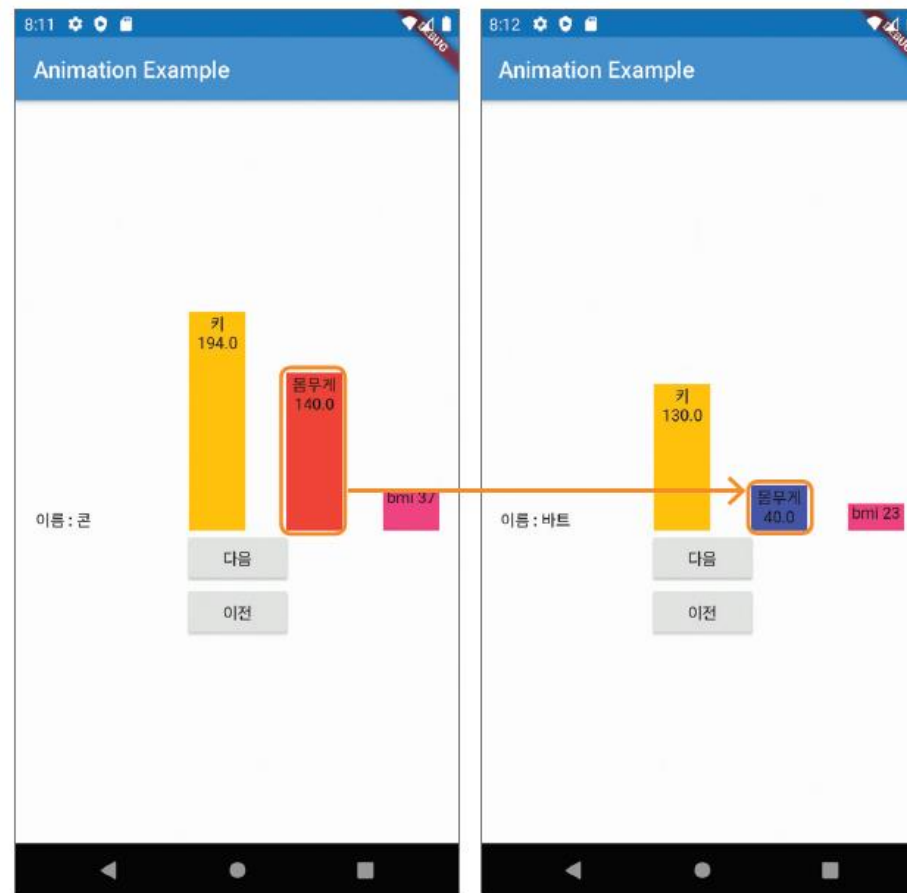
AnimatedContainer 를 이용하여 애니메이션을 구현할수있음

Color 를 변경하면 color 로 보간법을 이용하여 자연스럽게 색이 변함

• lib/main.dart

```
(...생략...)
AnimatedContainer(
  duration: Duration(seconds: 2),
  curve: Curves.easeInCubic,
  color: weightColor,
  child: Text(
    '몸무게 ${peoples[current].weight}',
    textAlign: TextAlign.center,
  ),
)(...생략...)
```

▶ 실행 결과



11-1 애니메이션 구현하기

AnimatedOpacity 를 이용하여 애니메이션을 구현할수있음 애니메이션에서 alpha 값을 이용한 투명도를 정할수있음

Duration : 애니메이션 실행시간

opacity : 위젯의 투명도 0 ~ 1 까지 값을 통해 결정

```
AnimatedOpacity(  
  opacity: _opacity,  
  duration: Duration(seconds: 1),  
  child: SizedBox(  
    child: Row(  
      children: <Widget>[ ...생략... ],  
      mainAxisAlignment: MainAxisAlignment.spaceAround,  
      crossAxisAlignment: CrossAxisAlignment.end,  
    ),  
    height: 200,  
  ),  
)
```

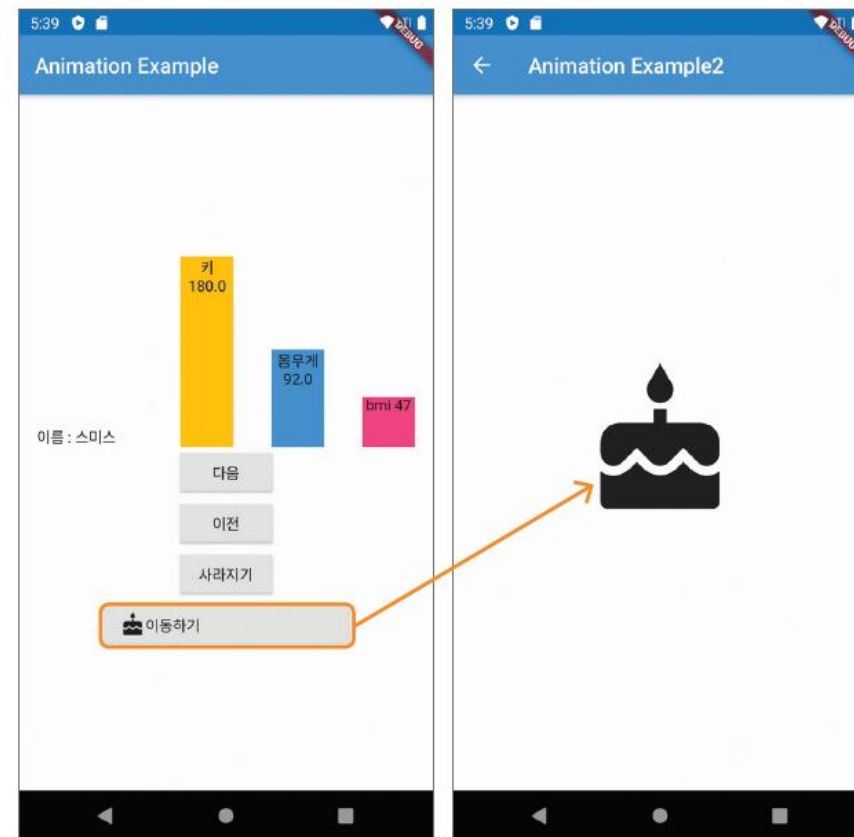
11-2 애니메이션 세세하게 다루기

페이지 이동시 사용할수있는 Hero 위젯

Hero 위젯의 tag를 똑같이 맞추면 child 에 원하는 위젯을 넣으면 Hero를 통해서 페이지 이동간에 widget을 연결되게 할 수 있음

```
child: Row(  
  children: <Widget>[  
    Hero(tag: 'detail', child: Icon(Icons.cake)),  
    Text('이동하기')  
  ],  
,),),
```

▶ 실행 결과



11-2 애니메이션 세세하게 다루기

실습하기

AnimationController를 이용하여 애니메이션을 미세하게 조정할 수 있음

Tween을 이용하여 애니메이션의 시작부터 끝까지의 값을 셋팅할 수 있음

애니메이션을 사용할 때
SingleTickerProviderStateMixin 을 상속받아야 함

```
class _SecondPage extends State<SecondPage>
  with SingleTickerProviderStateMixin {
    AnimationController? _animationController;
    Animation? _rotateAnimation;
    Animation? _scaleAnimation;
    Animation? _transAnimation;

    @override
    void initState() {
      super.initState();
      _animationController =
        AnimationController(duration: Duration(seconds: 5), vsync: this);
      _rotateAnimation =
        Tween<double>(begin: 0, end: pi * 10).animate(_animationController!);
      _scaleAnimation =
        Tween<double>(begin: 1, end: 0).animate(_animationController!);
      _transAnimation =
        Tween<Offset>(begin: Offset(0, 0), end: Offset(200, 200))
          .animate(_animationController!);
    }

    @override
    void dispose() {
      _animationController!.dispose();
      super.dispose();
    }
    (...생략...)
  }
```


11-2 애니메이션 세세하게 다루기

실습하기

애니메이션을 어떻게 넣느냐에 따라 앱의
퀄리티가 결정되어짐

▶ 실행 결과

