

# Flutter

## 간단한 Splash Screen 만들기

### 1. Flutter로 메인 화면 로딩 UI 구성

- 로딩 화면(Splash Screen):
  - `MaterialApp` 과 `Scaffold` 를 사용해 기본 화면 구성.
  - `Container` 의 `BoxDecoration` 을 활용해 배경색 설정.
- 가운데 정렬:
  - `Row` 와 `Column` 을 조합하여 로고와 로딩 애니메이션을 수직 및 수평 중앙에 배치.
  - `mainAxisAlignment: MainAxisAlignment.center` 를 사용해 정렬 처리.

### 2. 위젯 간 간격 추가 및 `CircularProgressIndicator` 를 이용한 로딩 애니메이션

- 로고와 `CircularProgressIndicator` 간격:
  - `SizeBox(height: 20)` 를 사용해 로고와 로딩 애니메이션 사이에 마진 추가.
  - `SizeBox` 는 위젯 간 간격 조정을 간단하게 구현할 수 있음.

#### • `CircularProgressIndicator`

##### 1. 종류

- **Indeterminate:** 진행 상태를 알 수 없을 때 사용, 끝없이 회전.
- **Determinate:** 진행 상태를 나타낼 수 있을 때 사용, `value` (0.0~1.0)로 진행률 표시.

##### 2. 주요 속성

- `value` : 진행률(0.0~1.0, 없으면 Indeterminate).
- `valueColor` : 애니메이션 색상.
- `strokeWidth` : 원형 두께 설정.

##### 3. 크기 조정

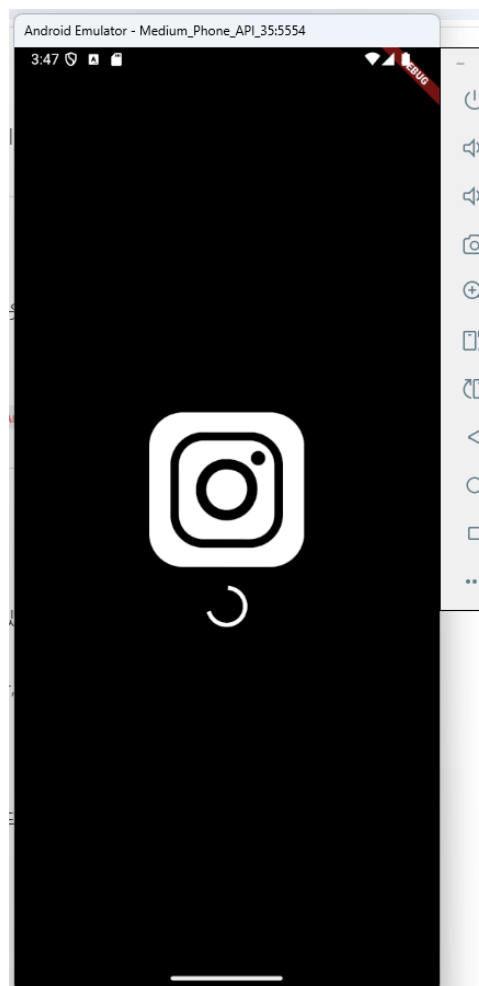
- `SizeBox` 또는 `Container` 로 크기 지정 가능.

#### 4. 활용

- 데이터 로딩, 다운로드 상태 표시 등에 사용.
- `LinearProgressIndicator` 는 선형 애니메이션으로 동일한 기능 제공.

### 3. 주요 코드 개념

- Flutter의 구조화된 레이아웃:
  - `Container` 로 스타일을 지정하고, `Row` 와 `Column` 으로 레이아웃을 정리.
  - 위젯 사이 간격 조정에는 `SizeBox` 를 활용.
- 로딩 애니메이션 색상 변경:
  - `CircularProgressIndicator` 의 `valueColor` 를 `AlwaysStoppedAnimation` 으로 설정해 색상을 커스터마이징.



## 회고

- 다음 목표:

- Flutter에서 상태 관리를 추가로 학습(BLoC 또는 Riverpod) 후 Bloc 과 Riverpod 중 무엇을 이용할 지 결정
- Splash Screen에 애니메이션 효과(페이드인/아웃 등) 추가 시도.
- Spotify 클론 코딩 시작하기 ( 개발 시작 전 클론 코딩으로 감각 익히기 연습 )