

안드로이드 프로그래밍

1주차

소개

- 김준우
- 안드로이드/iOS 앱, Front-end, Back-end, 윈도우/리눅스 앱 등을 개발
- 2014년, 신라대학교 사범대학 컴퓨터 교육과 졸업
- 2008년, 부일전자디자인고등학교 졸업
- 물어 볼 것이 있다면 아래 메일 주소로 메일을 보내주세요.

me@kimjunu.com

일정

일자	내용
9/5	Kotlin, Android 프로그래밍 기본, 시작 화면 UI 구현하기
9/12	To Do 앱 메인 화면 UI 구현하기
9/19	데이터베이스, 할 일 등록 기능 구현
9/26	할 일 조회, 수정(완료, 북마크 설정), 삭제 기능 구현
10/10	알림, 테마(Light / Dark 중 선택), 고급 기능 구현
10/17	앱의 완성도 올리기, 구글 플레이 스토어 등록 방법 알아보기, One more thing...

최종 목표

- 바쁘다 바빠 현대사회 알쏭달쏭 디지털 세상을 살아가는 현대인들의 자기 관리 필수품인 **To Do** 앱을 만드는 방법을 알 수 있음
- 코틀린(Kotlin) 언어로 안드로이드 앱을 개발하는 방법을 알 수 있음
- 안드로이드 앱에서 데이터 베이스에 접근하는 방법을 알 수 있음
- 안드로이드 앱에 테마 기능을 구현할 수 있음
- 안드로이드 앱을 구글 플레이 스토어에 등록하는 방법을 알 수 있음

수업 내용

- 우리는 무엇을 만드는가?
- 안드로이드 앱을 만드는 다양한 방법
- 우리는 왜 코틀린으로 안드로이드 앱을 만드는가?
- 생성, 빌드, 실행
- 시작 화면을 만들어보자

우리는 무엇을 만드는가?

- 내가 할 일을 잊어버리지않게 기록하고 확인할 수 있는 앱

안드로이드 앱을 만드는 다양한 방법

네이티브 도구를 이용한 방법

- Java, Kotlin을 이용한 앱 개발
- 안드로이드 스튜디오에서 개발할 수 있음
- 다른 방법에 비해서 성능이 제일 좋음
- 안드로이드 시스템이 제공하는 모든 기능 사용 가능

안드로이드 앱을 만드는 다양한 방법

웹 기반 기술을 이용한 방법

- HTML, CSS, Javascript를 이용한 웹 페이지 개발
- 모바일 기기에서 웹 페이지에 접속해서 사용
- 앱 설치가 필요 없고, 인터넷에 연결되면 언제 어디서든 사용 가능
- 하지만 성능이 좋지 않고 안드로이드의 일부 기능을 사용 못 함

안드로이드 앱을 만드는 다양한 방법

크로스 플랫폼 (혹은 하이브리드) 도구를 이용한 방법

- React Native, Xamarin, PhoneGap, Qt 등을 이용한 앱 개발
- 하나의 소스코드로 안드로이드, iOS, Windows 앱을 개발할 수 있음
 - 실제로는 iOS, Windows 용 소스코드를 추가로 작성해야 함
- 네이티브와 웹 기반 기술 개발의 중간 개념
- 네이티브 도구로 개발한 앱과 성능 차이가 거의 없음
- 웹 페이지에서 사용하지 못하는 기능 사용 가능

우리는 왜 코틀린으로 안드로이드 앱을 만드는가?

- 구글에서 Kotlin 을 Android 개발 언어로 밀고있음

Android에서 Kotlin을 최우선 지원 언어로 만든 이유는 무엇인가요?

Kotlin은 간결하고 표현력이 뛰어나며, 예기치 않은 유형 오류와 null이 발생하지 않도록 설계된 Android 호환 언어입니다. Kotlin은 자바 언어와 완전히 호환되므로, 자바 언어를 즐겨 사용하는 개발자는 자바 언어를 계속 사용하는 동시에 간편하게 Kotlin 코드를 점진적으로 추가하고 Kotlin 라이브러리를 활용할 수 있습니다. 또한 많은 Android 개발자가 이미 Kotlin을 사용하면 더 재밌고 빠르게 개발할 수 있다는 사실을 깨달았기 때문에 Google에서는 이러한 Kotlin 사용자를 위한 지원을 강화하고자 합니다. 자세한 내용은 [Kotlin과 Android](#)를 참조하세요.

우리는 왜 코틀린으로 안드로이드 앱을 만드는가?

Hello world

Java

```
public static void main(final String[] args) {  
    System.out.println("Hello world!")  
}
```

Kotlin

```
fun main() {  
    println("Hello world!")  
}
```

Variables I

Java

```
final int x;  
final int y = 1;
```

Kotlin

```
val x: Int  
val y = 1
```

Variables II

Java

```
int w;  
int z = 2;  
z = 3;  
w = 1;
```

Kotlin

```
var w: Int  
var z = 2  
z = 3  
w = 1
```

우리는 왜 코틀린으로 안드로이드 앱을 만드는가?

Book.java

```
1 public class Book {
2
3     private String isbn;
4     private String name;
5
6     public Book(String isbn, String name) {
7         this.isbn = isbn;
8         this.name = name;
9     }
10
11     public String getIsbn() {
12         return isbn;
13     }
14
15     public void setIsbn(String isbn) {
16         this.isbn = isbn;
17     }
18
19     public String getName() {
20         return name;
21     }
22
23     public void setName(String name) {
24         this.name = name;
25     }
26 }
27
```

Book.kt ▶ ...

```
1 class Book(var isbn: String, var name: String)
2
```

우리는 왜 코틀린으로 안드로이드 앱을 만드는가?

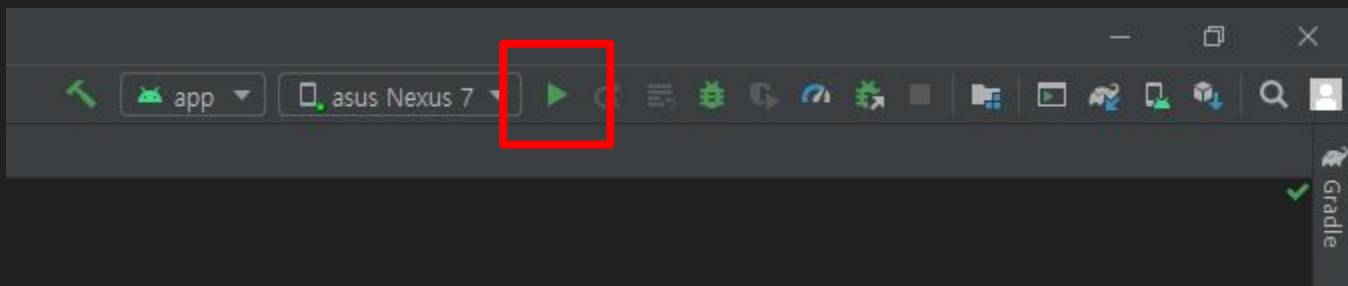
- 처음 프로그래밍을 한다면 읽기 쉽고, 만들기 쉬운 언어로 개발하자
- 안드로이드 스튜디오에서 코틀린 언어를 지원하기 때문에 개발이 쉬움

생성, 빌드, 실행

- 프로젝트 생성
 - Start a new Android project 클릭
 - Project Template은 Empty Activity 선택 후 Next 클릭
 - Name에 Today I am Done 입력 후 Finish 클릭

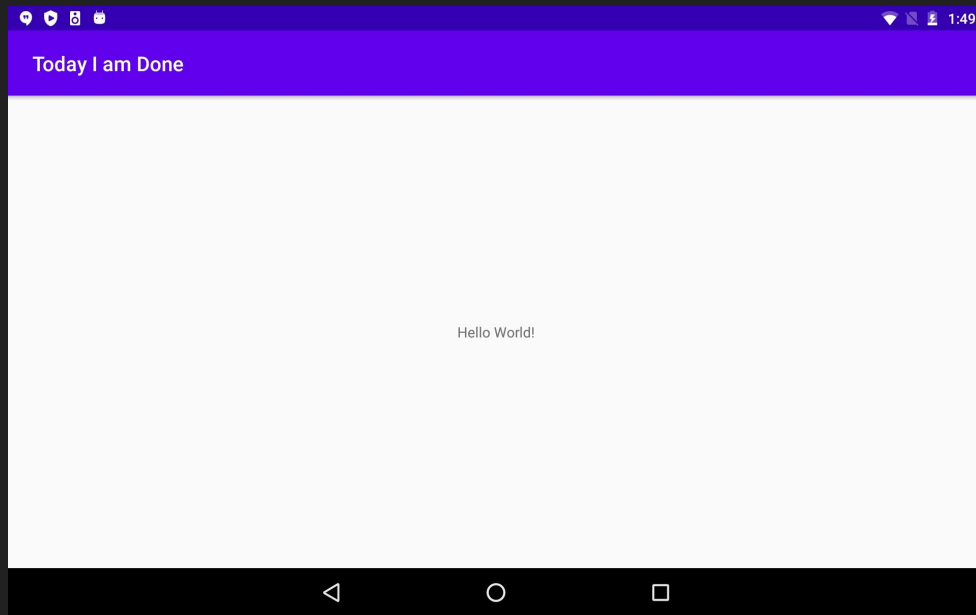
생성, 빌드, 실행

- 앱 빌드 및 설치



생성, 빌드, 실행

- 실행

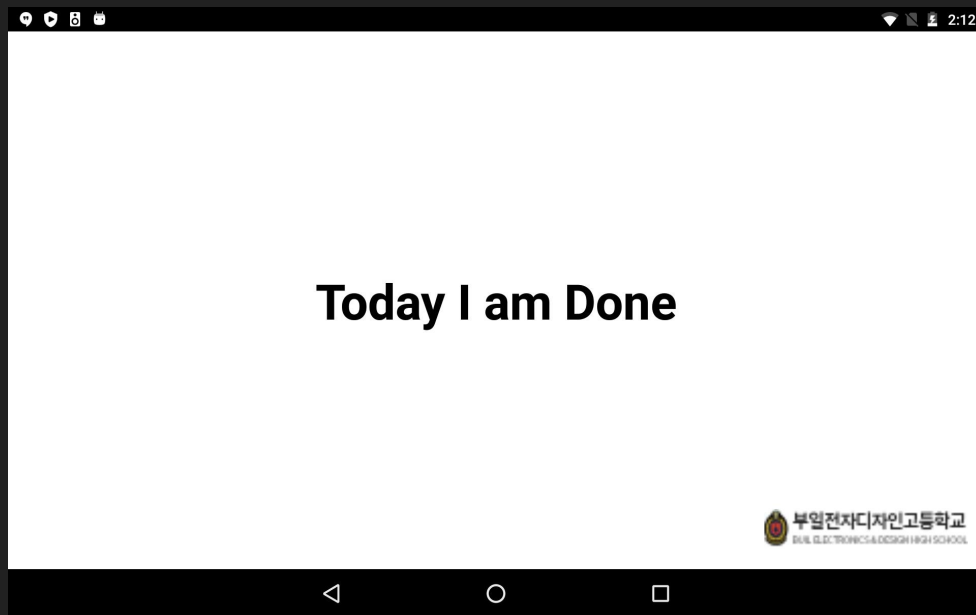


생성, 빌드, 실행

아주 개인적인 프로그램을 개발하는 방법

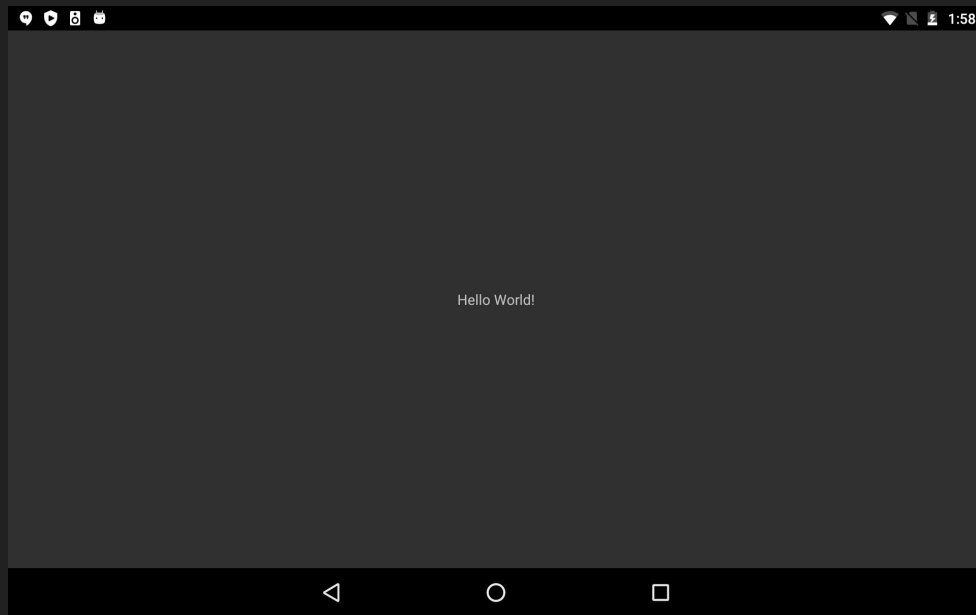
1. 프로그램의 모습을 확인 할 수 있는 **UI** 먼저 구현한다.
 - 가짜 데이터를 넣어서 모습 확인
2. 데이터를 담은 모델을 구현한다.
3. 모델과 **UI**를 연동하여 처리하는 클래스 및 함수들을 구현한다.
4. 테스트, 테스트, 버그 수정, 테스트, 테스트, 또 테스트
5. 완성!

오늘 목표



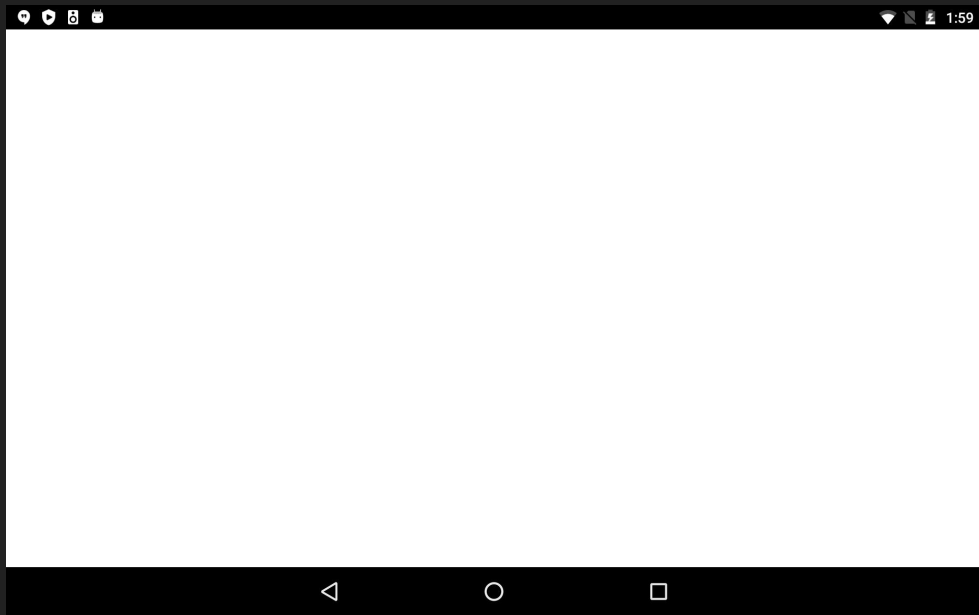
시작 화면을 만들어보자

- Action Bar 없애기



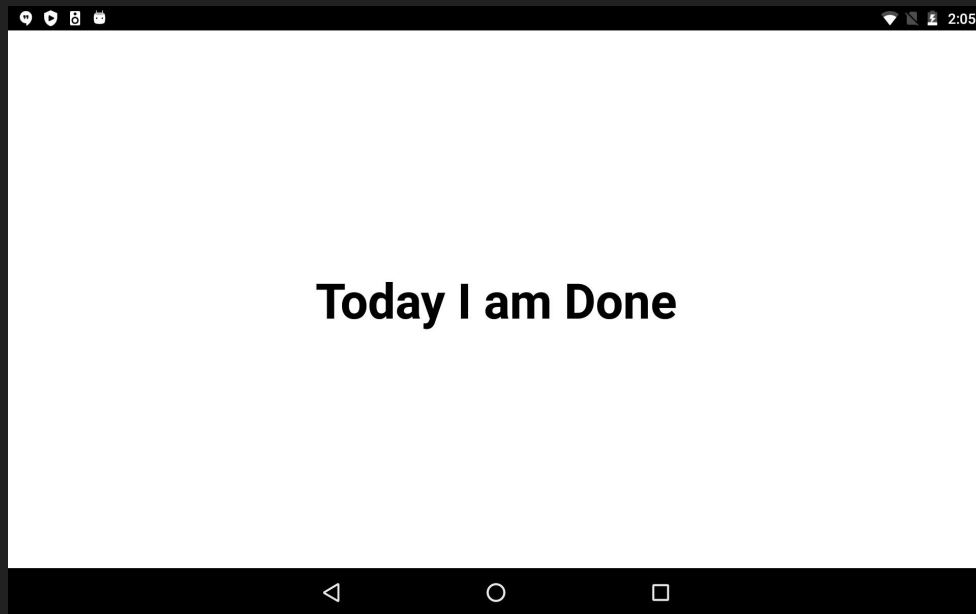
시작 화면을 만들어보자

- 화면 배경 색 변경: `background`



시작 화면을 만들어보자

- TextView 의 속성 변경: `text`, `textColor`, `textSize`, `textStyle`



시작 화면을 만들어보자

- 이미지(학교 로고) 추가
 - url
 - 파일을 다운 받아서 각자 컴퓨터의 TodaylamDone\app\src\main\res\mipmap-xxxhdpi 에 저장

시작 화면을 만들어보자

- ImageView 에 이미지 불러오기 및 속성 변경

