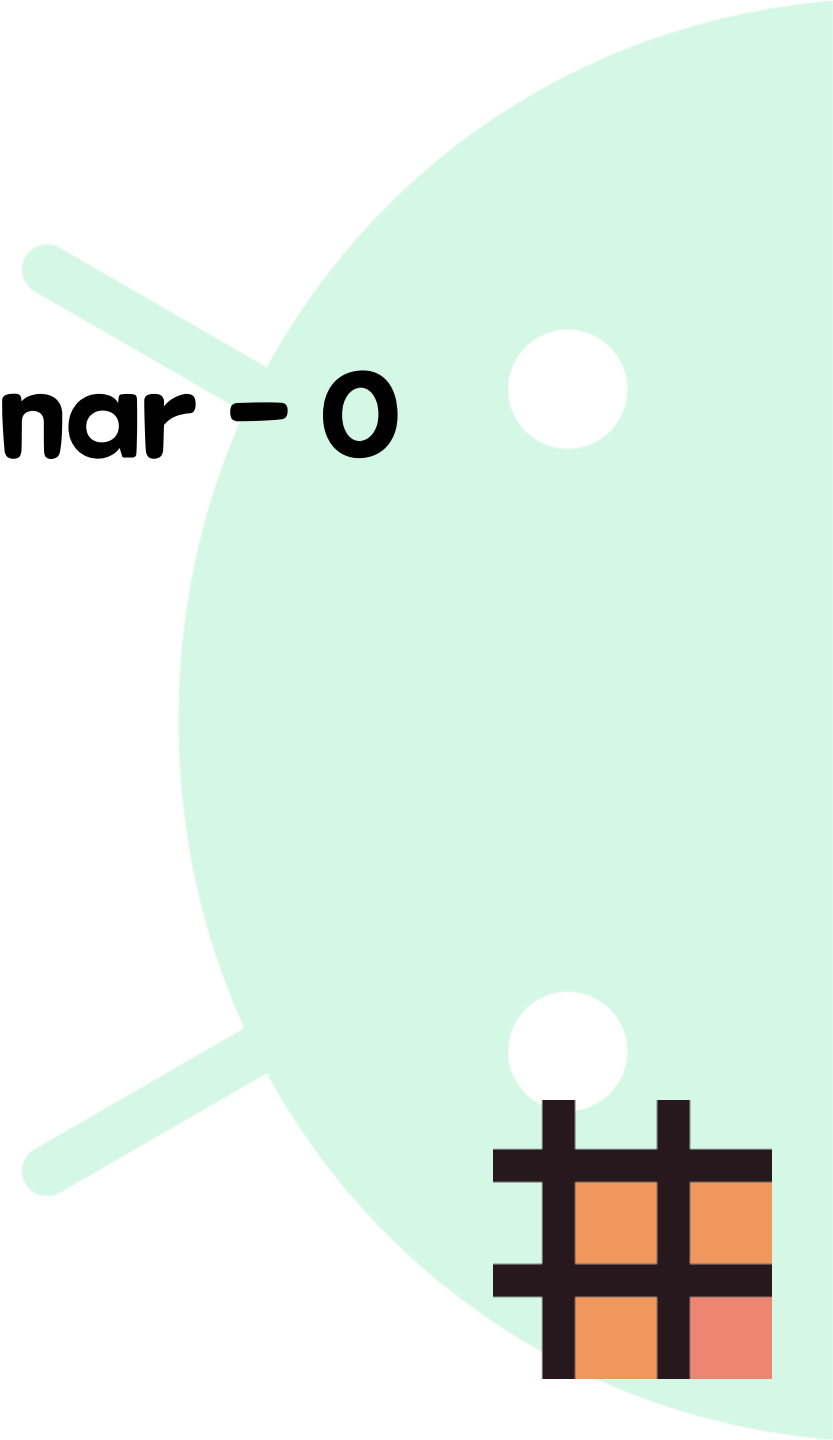


WaffleStudio Android Seminar - 0

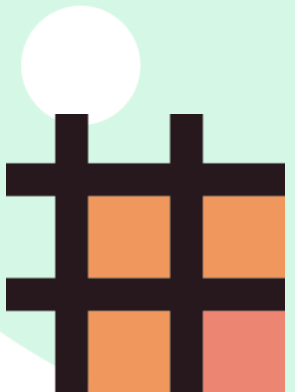
이승민 (안드로이드 세미나장)

2021.08.28.(토) 11:30 ~



세미나장 소개

- 이승민 (@veldic)
- Android Developer
- 컴퓨터공학부 19학번
- 컴퓨터공학부 전 부학생회장, 와플스튜디오 전 부와장
- ~~안드로이드 10년째 사용 중 (아이폰 사용 경험 X)~~



Android를 배우면...

1. 재밌다!

- 내가 짠 코드로 돌아가는 앱을 직접 실행시켜보며 눈으로 확인할 수 있다.
- 특히 심혈을 기울여 짠 버튼 하나를 눌렀을 때 버그 없이 실행되는 짜릿함이란...

2. 평소에 사용하던 앱 구조가 눈에 보인다!

- 세상을 보는 해상도가 높아지는 기분
 - 이 앱의 이 부분은 ViewPager2를 이용하여 만들어졌군...
 - 세상에 RecyclerView를 이렇게까지 쓴단 말이야?

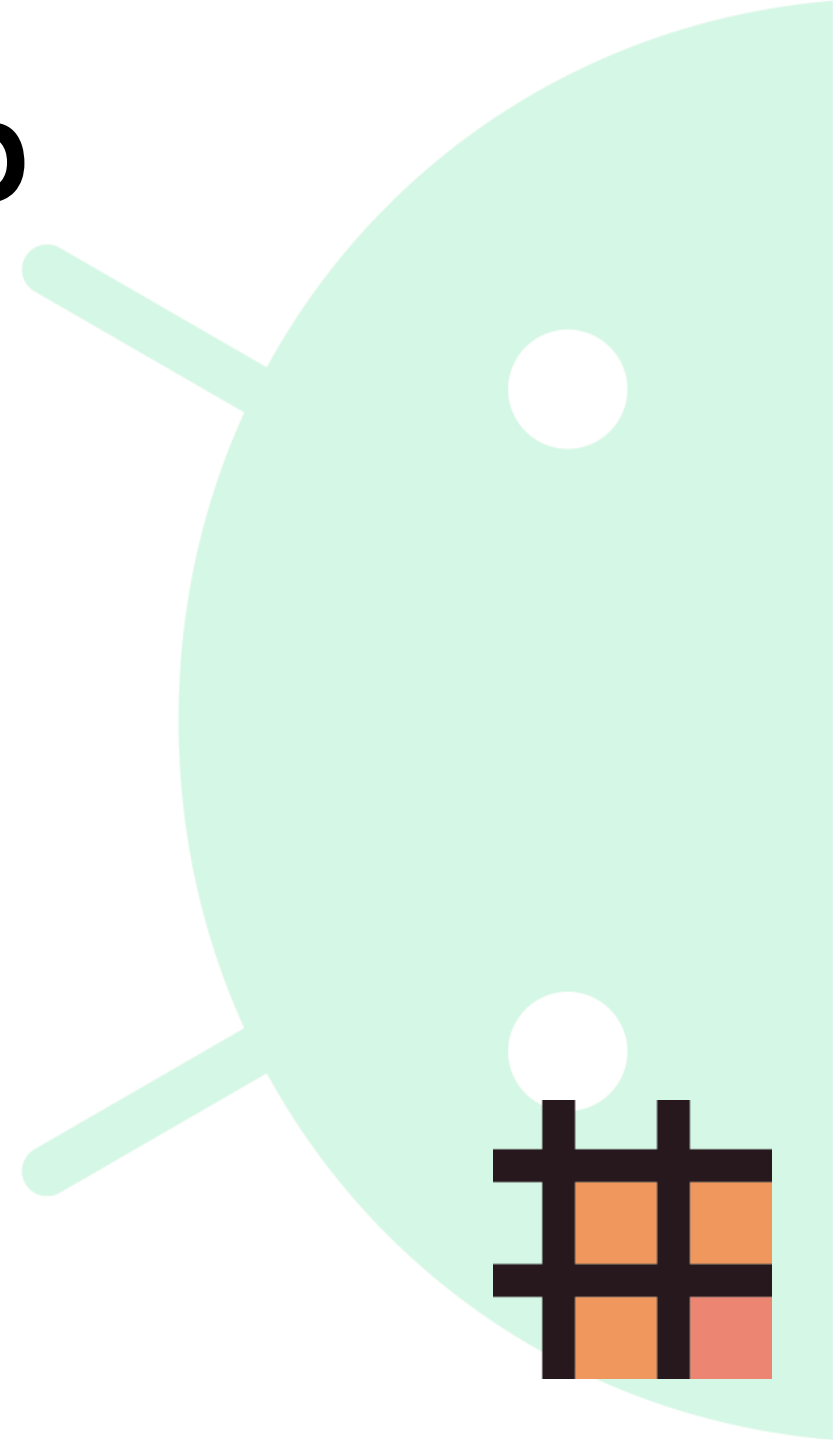
3. ~~앱에 디자인이 얼마나 필수인지 알 수 있다.~~

- ~~• 디자인 없는 앱 - 다이아몬드 원석 - 이렇게 내 앱이라고?~~
- ~~• 디자인 있는 앱 - 다이아몬드 보석 - 이게 앱이지.~~



What we will learn in Seminar 0

- Basic Kotlin
- What is Android?
- Start from the beginning
- About Activity



Basic Kotlin

JVM 위에서 동작함

Kotlin <-> Java 간 서로 참조 가능

OK!



```
class KotlinClass(private val a: Int) {  
    fun doSomething(b: Int): Int {  
        return a + b  
    }  
}
```



```
public class JavaClass {  
    public static void main(String[] args) {  
        int c = 1;  
        KotlinClass d = new KotlinClass(c);  
        System.out.println(d.doSomething(1));  
    }  
}
```



Basic Kotlin

Null 처리가 깔끔함

Compile Error

```
val a: Class? = null  
a.doSomething()
```



OK!

```
val a: Class? = null  
a?.doSomething()
```

Runtime Error
(NullPointerException)

```
Class a = null;  
a.doSomething();
```

? 외에도 !!, let 등 null 처리에 다양한 방식을 적용 가능!



Basic Kotlin

그 외 선언 방식에서의 차이... 등

Kotlin은 개인적으로 공부하셔야 원활하게 세미나를 따라오실 수 있습니다! (그리 어렵진 않아요!)

Kotlin Practice

<https://play.kotlinlang.org/byExample/overview>

▼ Introduction

Hello World

Functions

Variables

Null Safety

Classes

• Generics

Inheritance

▼ Control Flow

When

Loops

Ranges

Equality Checks

Conditional Expression

• : Optional

```
val a = 0
var b = 1

a = 3 // ERROR
b = 4 // OK
```



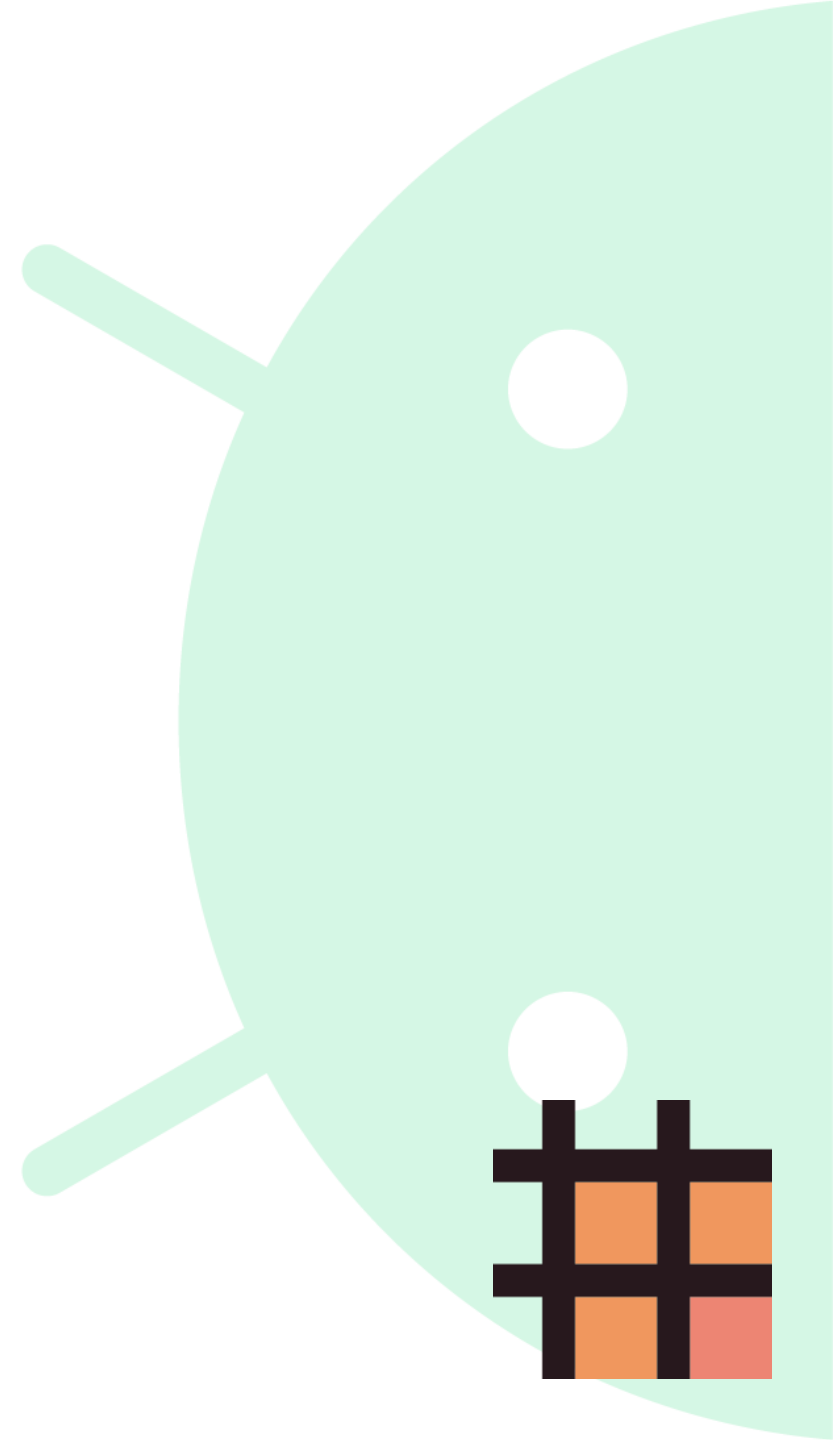
What is Android?

- 전 세계 점유율 72.21%의 Mobile OS
(StatCounter, July 2021)

- 오픈 소스 플랫폼

- 구글 개발 

But... 이렇게 중요한 것이 아니다...
그래서 개발할 때 중요한 것이 뭐가!



What is Android?

- OS
 - 하드웨어 자원의 관리
 - 여러 프로그램(프로세스) 들이 돌아갈 수 있는 기반을 제공



android 11



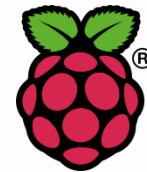
android 



macOS Big Sur

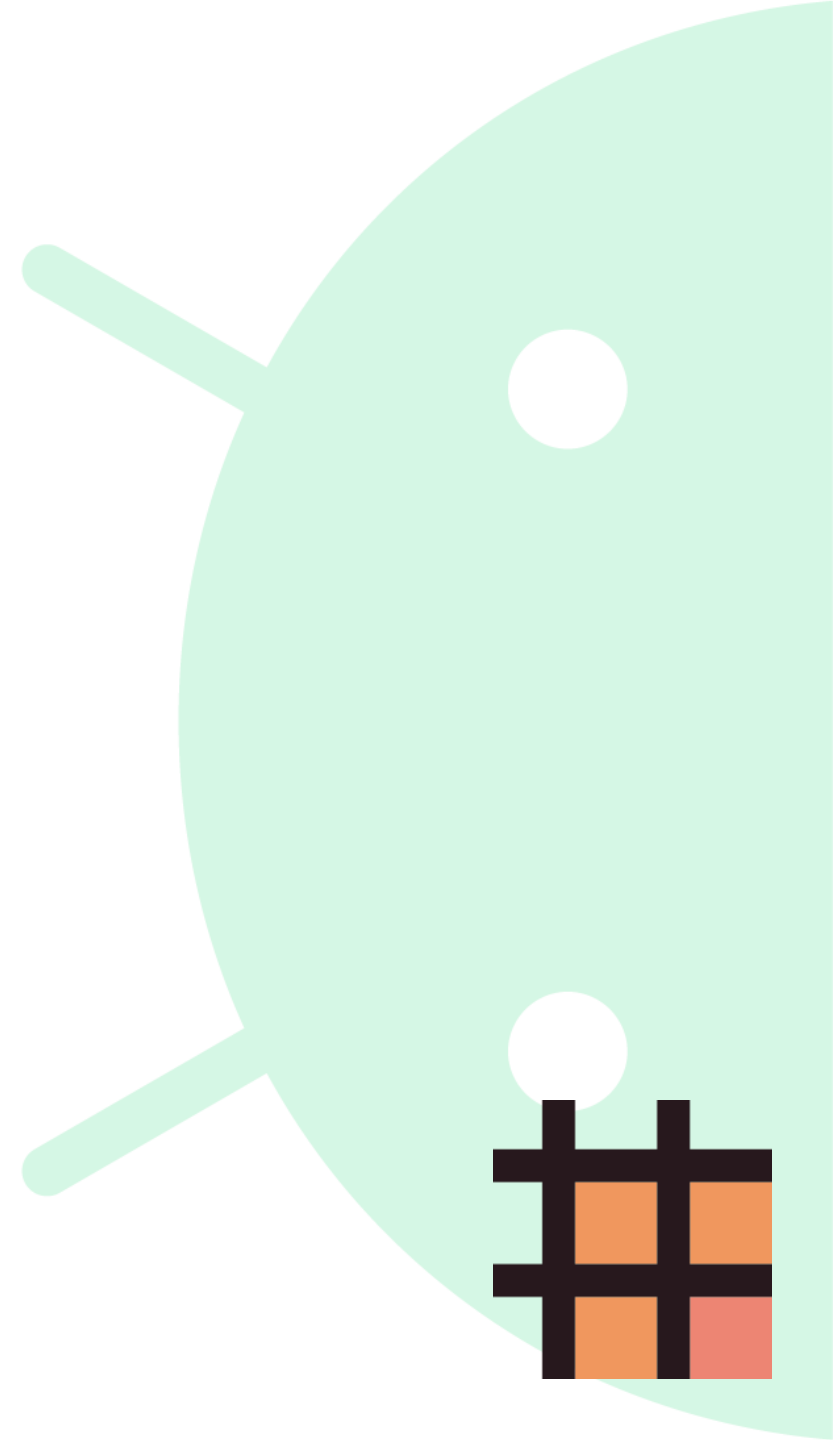


iOS



What is Android?

- Android OS가 관리하는 것들
 - 배터리 상태 유지 및 절전
 - 메모리 관리
 - 위치 정보, 비밀번호, 지문 등 보안 관리
 - 저장 공간 관리
 - UI 업데이트
 - 사용자 터치 감지 및 피드백
 - 블루투스 관리
 - 사운드 출력 관리
 - 등등...



What is Android?

- Android OS가 관리하는 것들
 - 배터리 상태 유지 및 절전
 - 메모리 관리
 - 위치 정보, 비밀번호, 지문 등 보안 관리
 - 저장 공간 관리
 - UI 업데이트
 - 사용자 터치 감지 및 피드백
 - 블루투스 관리
 - 사운드 출력 관리
 - 등등...

우리가 세미나에서 다룰 것과 관련된 부분!



Start from the beginning

- Android App Component (앱 기본 요소)
 - Activity
 - 앱이 사용자에게 보여지는 화면에 포함된 모든 요소들
 - Service
 - 앱의 백그라운드에서 실행되고 있는 다양한 요소들 (업로드, 다운로드, 타이머 등)
 - Broadcast Receiver
 - App과 App 또는 App 자기 자신과의 통신을 위한 코드 (인스타그램 공유, 카카오톡으로 공유 등)
 - Contents Provider
 - 다른 App의 데이터를 가져오기 위한 코드 (파일 시스템, 갤러리 등)



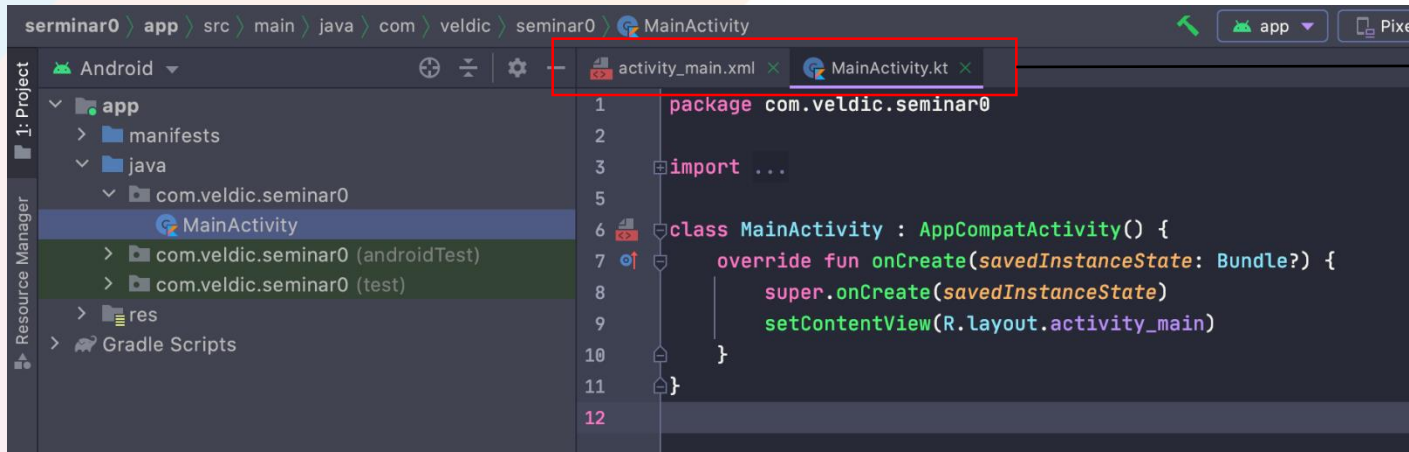
Start from the beginning

- Android App Component (앱 기본 요소)
 - Activity
 - 앱이 사용자에게 보여지는 화면에 포함된 모든 요소들
 - Service
 - 앱의 백그라운드에서 실행되고 있는 다양한 요소들 (업로드, 다운로드, 타이머 등)
 - Broadcast Receiver
 - App과 App 또는 App 자기 자신과의 통신을 위한 코드 (인스타그램 공유, 카카오톡으로 공유 등)
 - Contents Provider
 - 다른 App의 데이터를 가져오기 위한 코드 (파일 시스템, 갤러리 등)



About Activity

백문이 불여일견, 한 번 만들어보자!

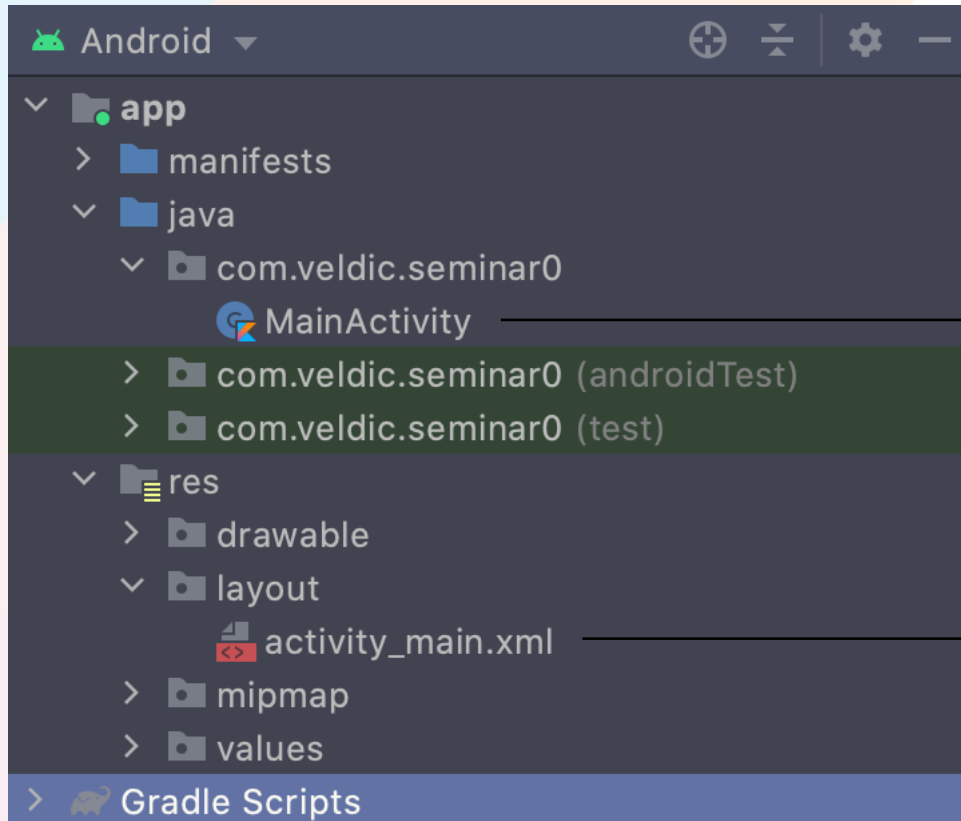


activity_main.xml?
MainActivity.kt?



About Activity

- Activity
 - 앱이 사용자에게 보여지는 화면에 포함된 모든 요소들



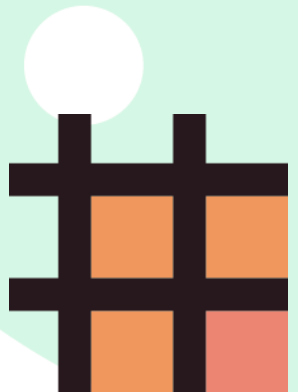
MainActivity의 이름에 걸맞게
사용자에게 main으로 보여지는
Activity의 요소들을 담은 파일

Similar as...

```
int main() {  
    return 0;  
}
```

in C

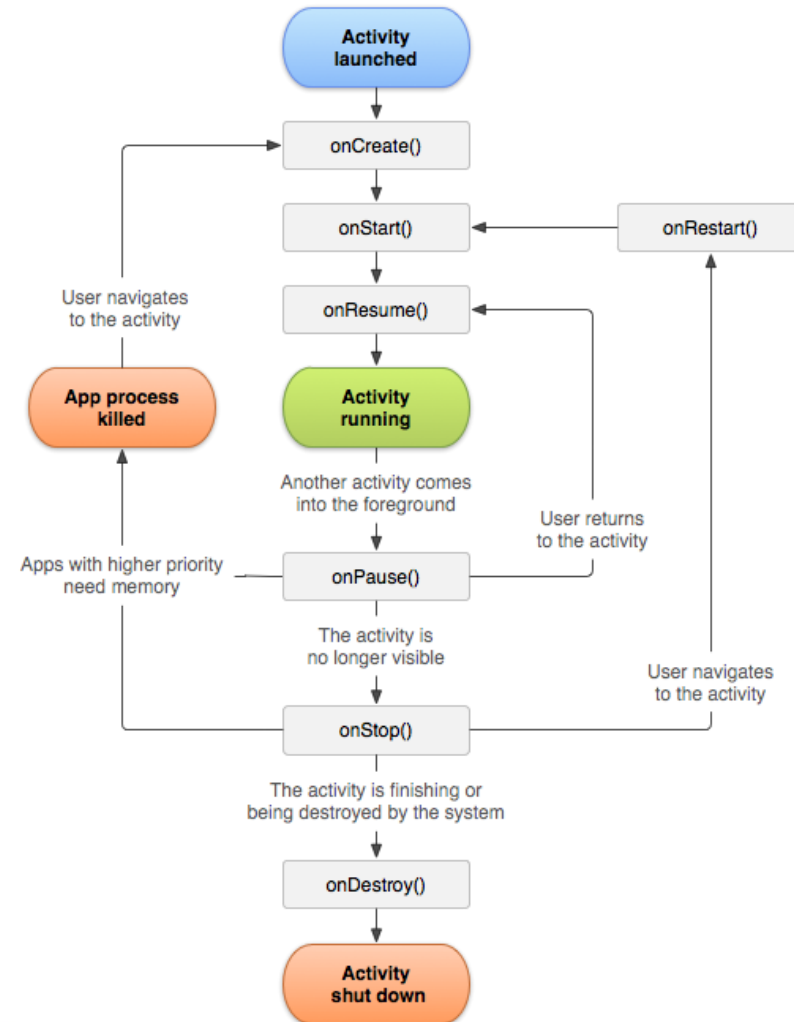
MainActivity의 디자인을
결정짓는 파일



About Activity

Activity Lifecycle

Activity Lifecycle을 이해해야
적절한 과정에서 적절한 작업을 실행할 수 있다!



About Activity

onCreate()

Activity가 만들어진다. (Rendering)

Activity의 onCreate 함수를 overriding해서 어떤 동작을 할 지 정해줄 수 있다.

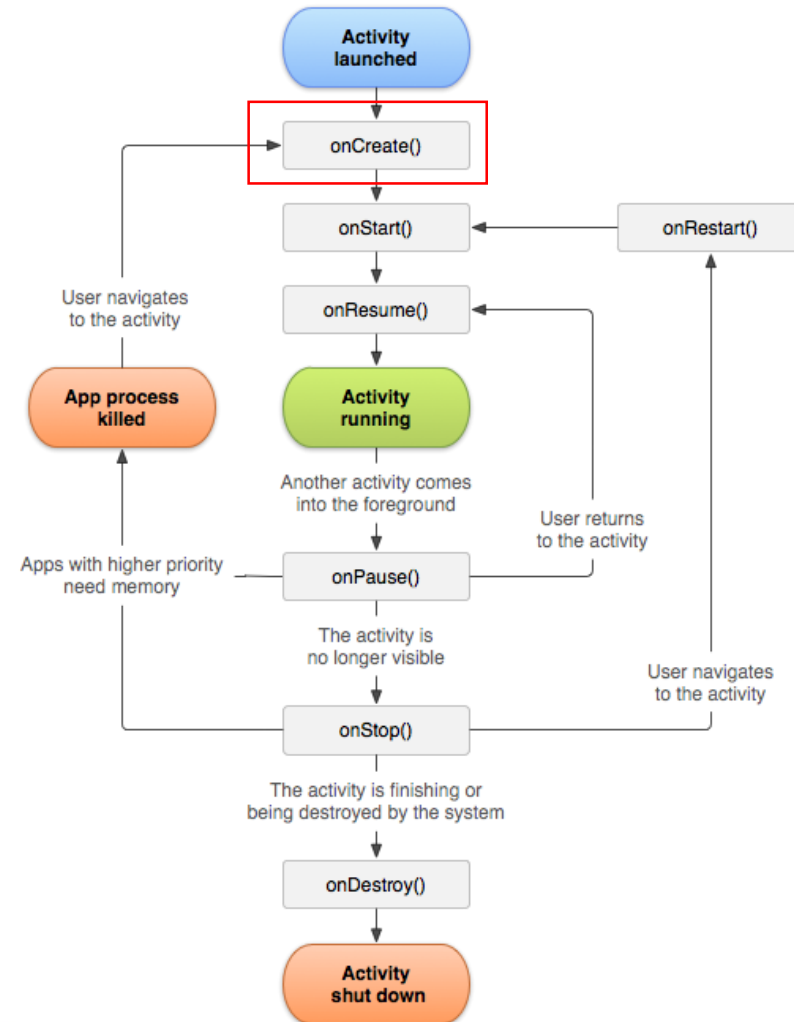
대부분의 UI 렌더링을 진행하기에 세미나에서 가장 사용할 일이 많은 부분

setContentView(R.layout.activity_oo)

해당 Activity가 어떤 UI (layout, ~~.xml) 파일을 사용하는 지 알려주어 해당 xml을 inflate한다.

```
6 class MainActivity : AppCompatActivity() {  
7     override fun onCreate(savedInstanceState: Bundle?) {  
8         super.onCreate(savedInstanceState)  
9         setContentView(R.layout.activity_main)  
10    }  
11 }
```

처음 실행했을 때 MainActivity에서 떠있던 코드



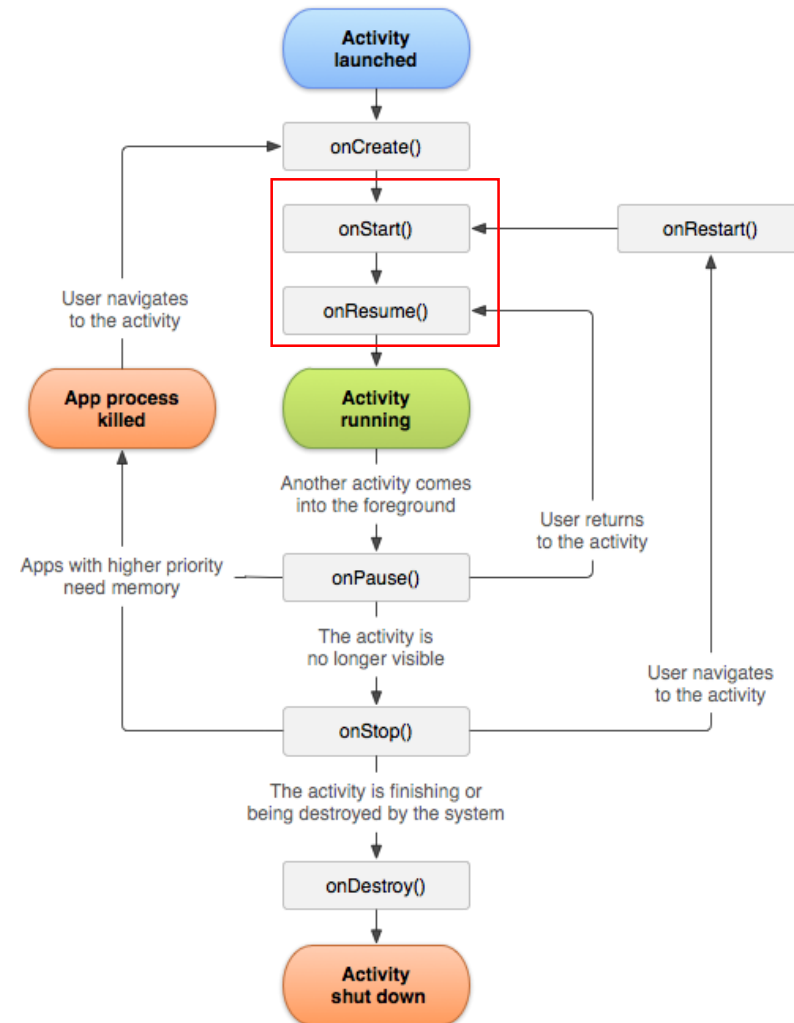
About Activity

onResume(), onStart()

Activity가 사용자에게 보인다.

onCreate()에서 Activity는 단지 만들어지기만 했을 뿐! onResume()과 onStart()에서 사용자에게 보여지기 시작한 시점에 실행시켜야 할 코드를 실행시킨다.

Ex) 카메라를 사용하는 경우 onResume()에서 카메라와의 상호작용을 시작할 수 있다.



About Activity

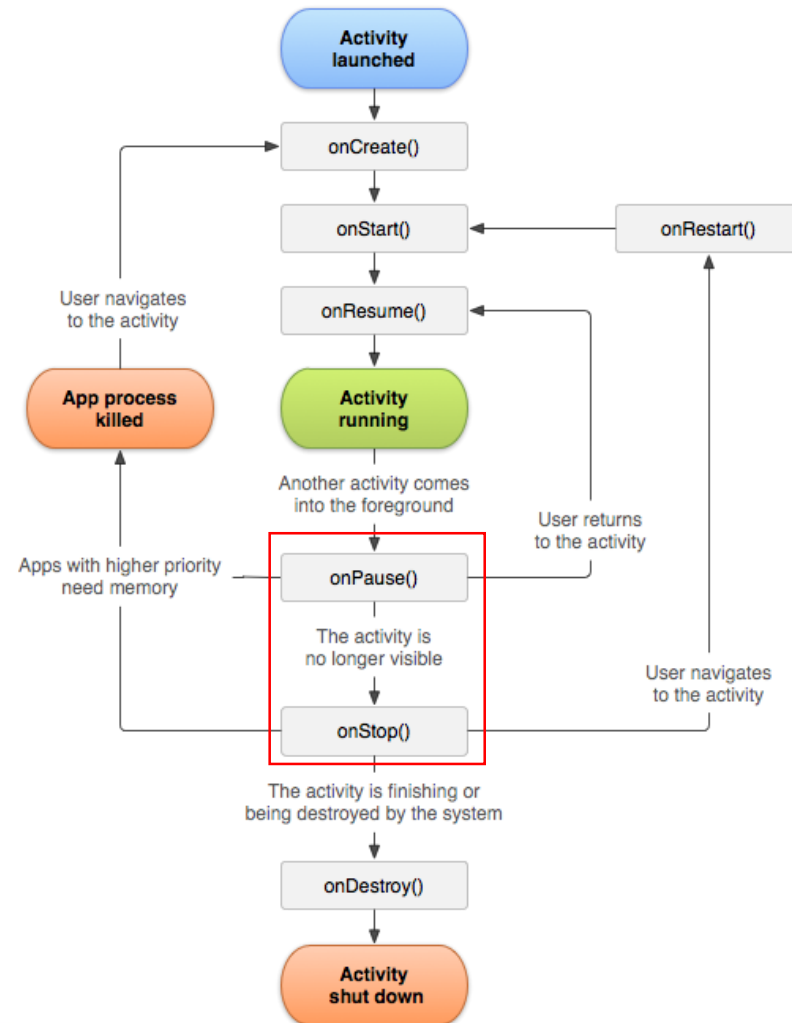
onPause(), onStop()

Activity가 사용자에게 보이지 않는다.

사용자가 다른 Activity를 실행시켜서 현재 Activity가 가려지거나, 오래 폰을 켜놓아서 화면이 꺼졌거나 등 다양한 이유로 Activity가 보이지 않으면 위 함수가 불리게 된다.

Ex) 카메라를 사용하는 경우 onPause()에 카메라 화면을 끌 수 있다.

(계속 켜놓으면 배터리나 메모리 측면에서 좋지않다!)



About Activity

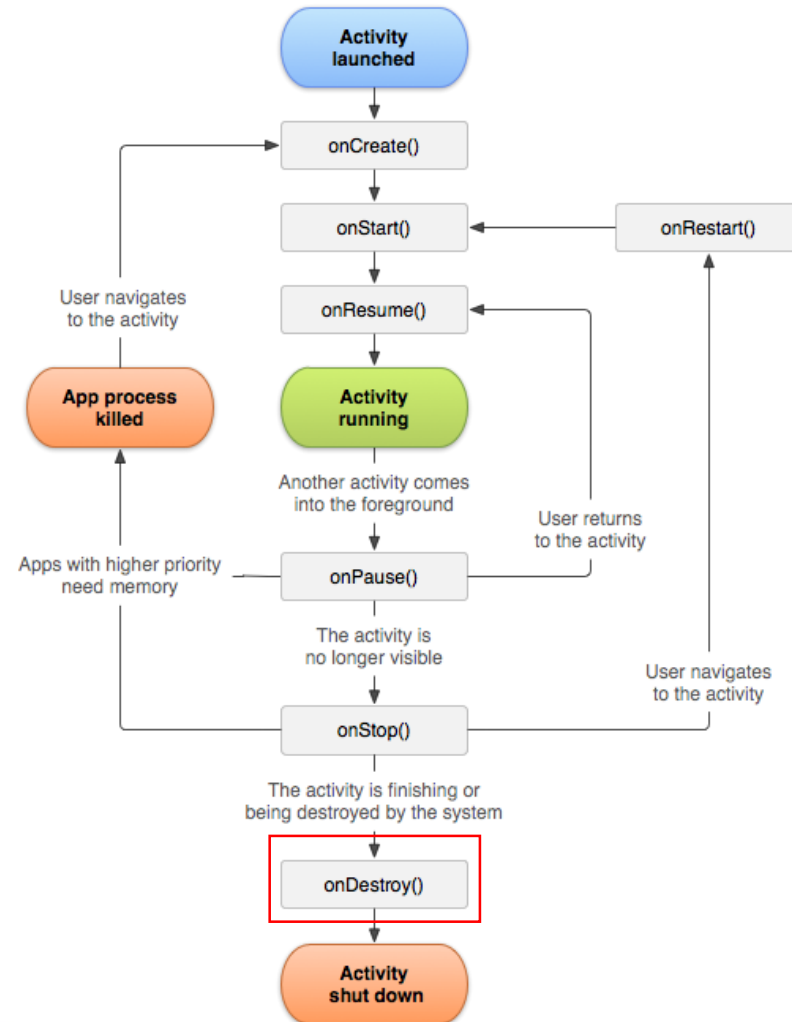
onDestroy()

Activity가 Android OS에 의해 완전히 종료되었다.

사용자가 back button으로 액티비티나 앱을 종료한 경우나 사용하지 않는 앱이 화면에 떠있지 않은 상태로 너무 오래 지속된 경우

할당되었던 자원을 해제한다.

(대부분 자동으로 진행되나, 가끔 자동으로 진행되지 않는 자원의 경우 수동으로 해제해주어야 한다.)



About Activity

Activity의 디자인 -> XML (Extensible Markup Language) file

- Android의 UI를 표현하는 방법
- HTML처럼 <tag></tag> 문법을 사용
- 항상 모든 코드를 짤 필요는 X 안드로이드 스튜디오를 활용할 수 있다!
 - ~~다만 레이아웃이 복잡해질수록 코드가 편하다...~~

간단한 Hello World!조차도 생각보다 많은 정보량

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello World!"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintLeft_toLeftOf="parent"
16         app:layout_constraintRight_toRightOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19 </androidx.constraintlayout.widget.ConstraintLayout>
```



About Activity

View

안드로이드 UI를 구성하는 요소들을 view라고 부른다.

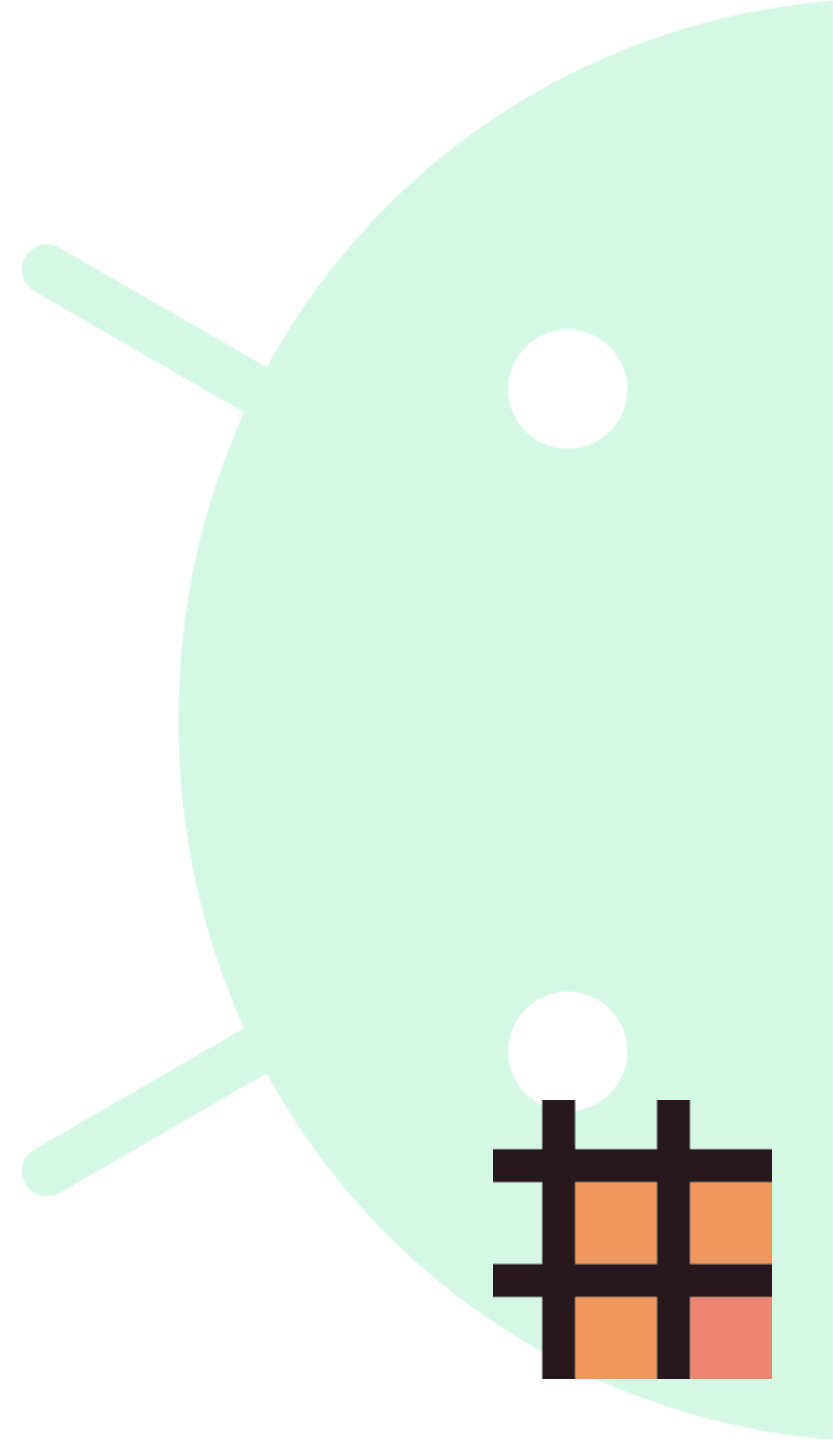
텍스트를 보여주는 TextView

특정 요소들의 리스트를 보여주는 ListView

사진 또는 그림을 보여주는 ImageView

사용자의 클릭을 받아주는 Button (이 또한 View의 일종이다.)

등...



About Activity

정보 파헤쳐보기

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      tools:context=".MainActivity">
9
10     <TextView
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Hello World!"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintLeft_toLeftOf="parent"
16         app:layout_constraintRight_toRightOf="parent"
17         app:layout_constraintTop_toTopOf="parent" />
18
19 </androidx.constraintlayout.widget.ConstraintLayout>
```

→ Layout 종류

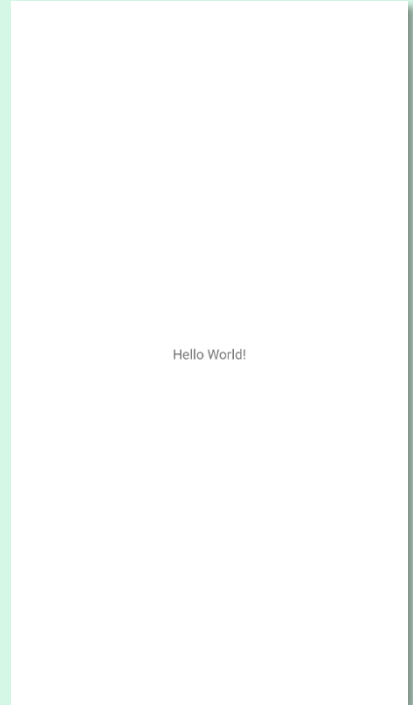
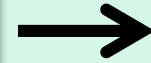
→ Import와 같은 역할

→ Layout의 가로 세로

→ TextView의 가로 세로

→ TextView의 text (내용)

→ TextView의 상대적 위치

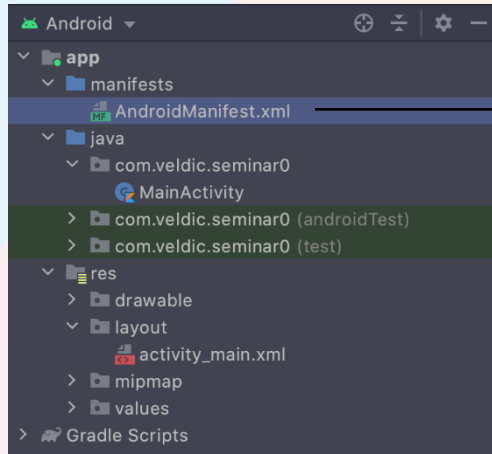


About Activity

그렇다면 Android는 이 MainActivity를 어떻게 실행시킬까?

Java에서는 자동으로 main function을
찾아 실행시킨다.
(물론 바꿀 수는 있다.)

```
public class AndroidSeminar0 {  
    public static void main(String[] args) {  
        // AUTOMATICALLY RUN  
    }  
}
```



```
1  <?xml version="1.0" encoding="utf-8"?>  
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"  
3      package="com.veldic.seminar0">  
4  
5      <application  
6          android:allowBackup="true"  
7          android:icon="@mipmap/ic_launcher"  
8          android:label="Seminar0"  
9          android:roundIcon="@mipmap/ic_launcher_round"  
10         android:supportRtl="true"  
11         android:theme="@style/Theme.Seminar0">  
12         <activity android:name=".MainActivity">  
13             <intent-filter>  
14                 <action android:name="android.intent.action.MAIN" />  
15  
16                 <category android:name="android.intent.category.LAUNCHER" />  
17             </intent-filter>  
18         </activity>  
19     </application>  
20  
21 </manifest>
```

Activity는 필수적으로
AndroidManifest.xml
에 명시되어 있어야 함

Intent를 이용!



About Activity

Intent?

-> 무언가를 하려는 의도

-> 새로운 Activity를 열려는 의도!

-> 그 외에도 Android App Component 중 BroadcastReceiver나 Service를 이용할 때도 사용할 수 있다.

Intent를 이용하여 한 Activity에서 다른 Activity로 이동할 수 있다.

```
val intent = Intent(context, LoginActivity::class.java)
startActivity(intent)
```

LoginActivity라는 Activity를 실행 및 이동하는 코드



Summary

Kotlin

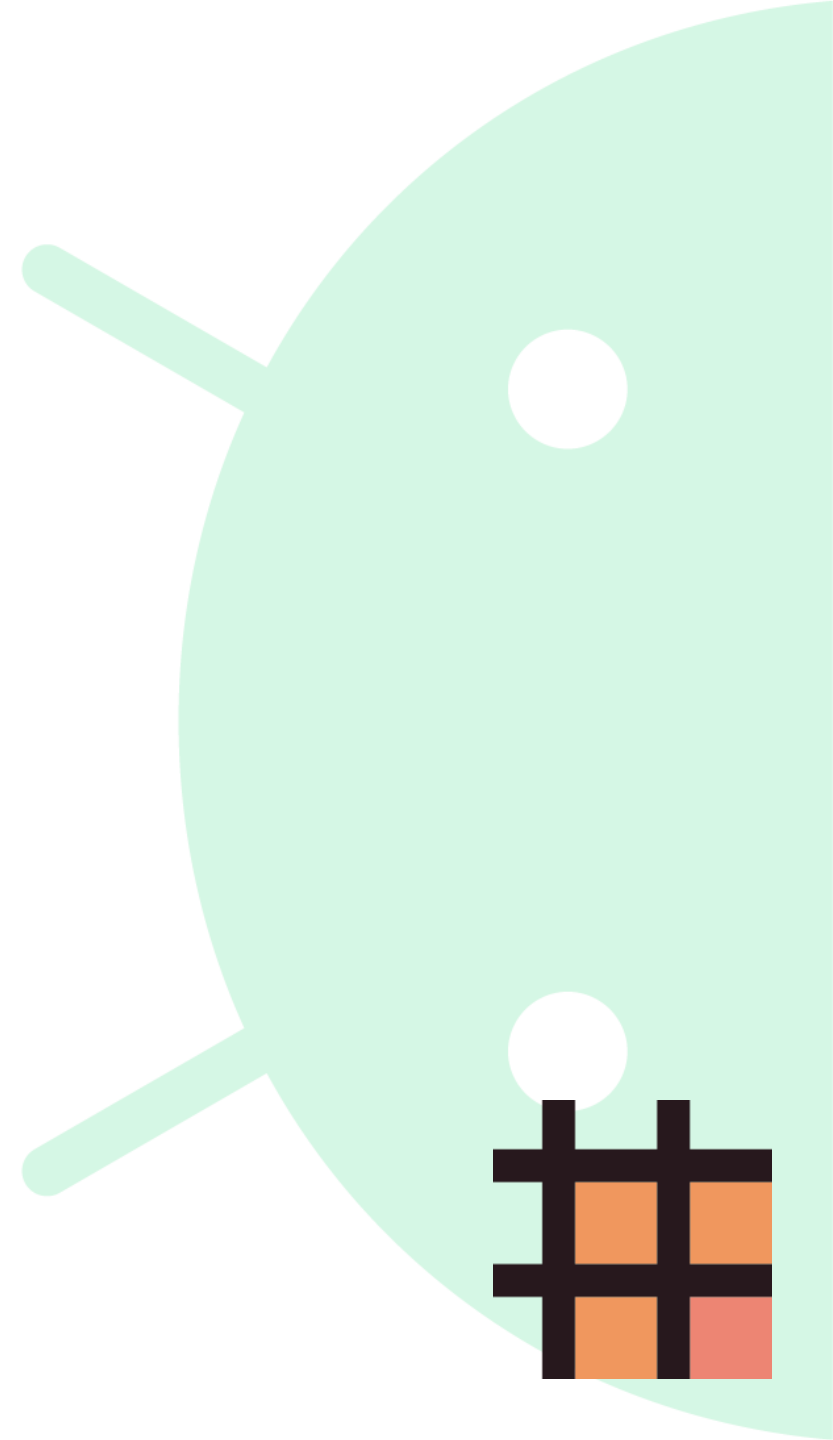
- 안어려워요! 기존에 프로그래밍 경험이 있다면 충분히 쉽게 따라오실 수 있습니다.

Android OS

- 모바일 기기를 동작시키기 위해 하드웨어 자원 관리 및 다양한 프로그램들이 돌아갈 수 있는 기반 제공

Android App Component

- Activity, Service, Broadcast Receiver, Contents Provider 이 있다.
- 하지만 세미나에선 Activity만 사용할 예정



Summary

Activity

- 앱이 사용자에게 보여지는 화면에 포함된 모든 요소들
- `OOOActivity.kt`과 `activity_ooo.xml` 파일로 구성
- Intent를 통해 다른 Activity를 실행 가능

Activity Lifecycle

- `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, `onDestroy()`로 구성
- 당장 주목해야할 부분은 `onCreate()` - view 생성의 대부분을 차지함

XML file

- Activity의 디자인을 결정짓는 파일
- View의 관계나 배치 형태를 확인 가능



Assignment 0

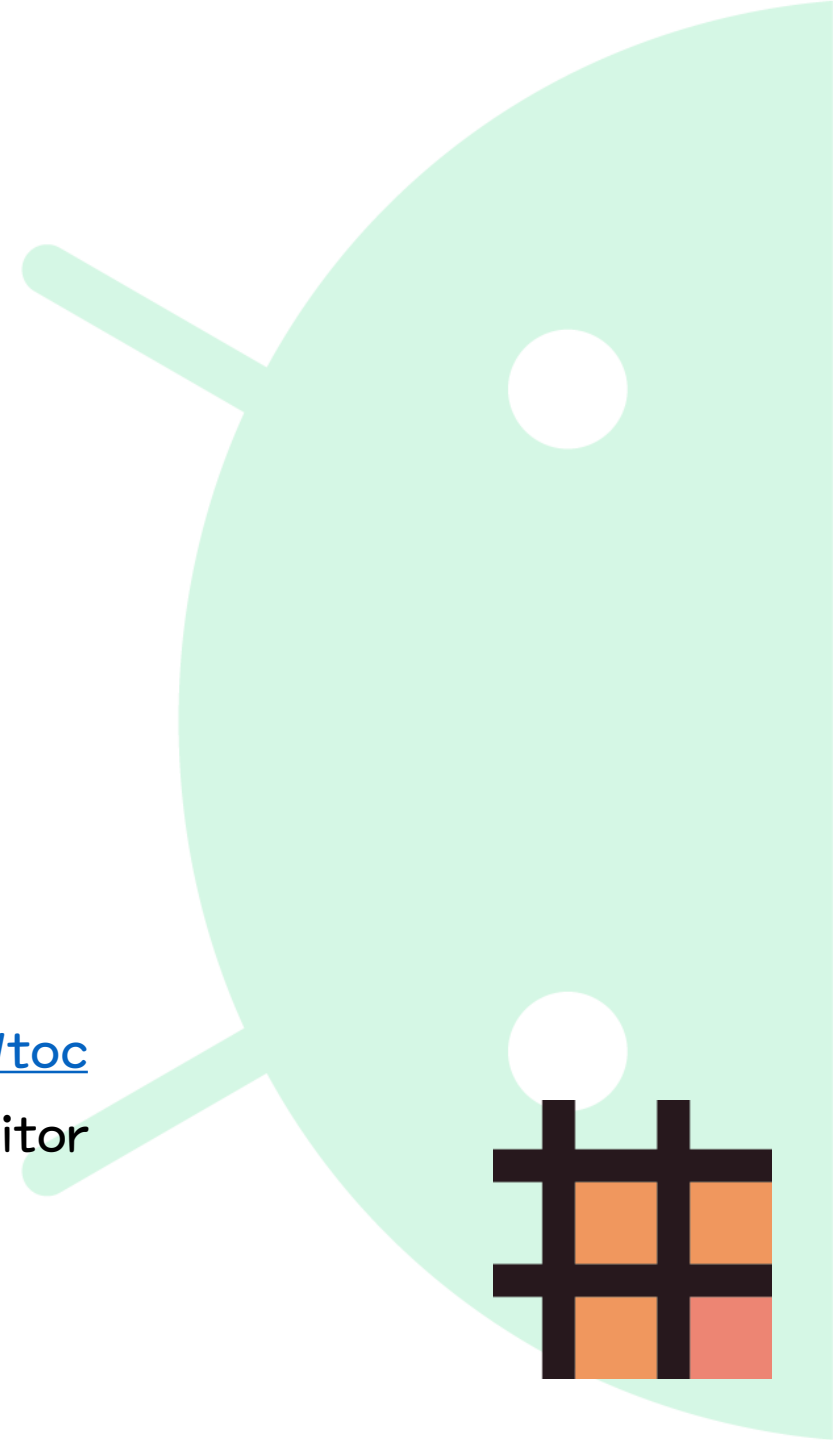
아직 개강 안하셨으니 심심하실까봐...
Kotlin & Android와 친해져보자!

Assignment Detail 및 제출 방법은
github의 assignment 0 README.md를 확인해주세요!

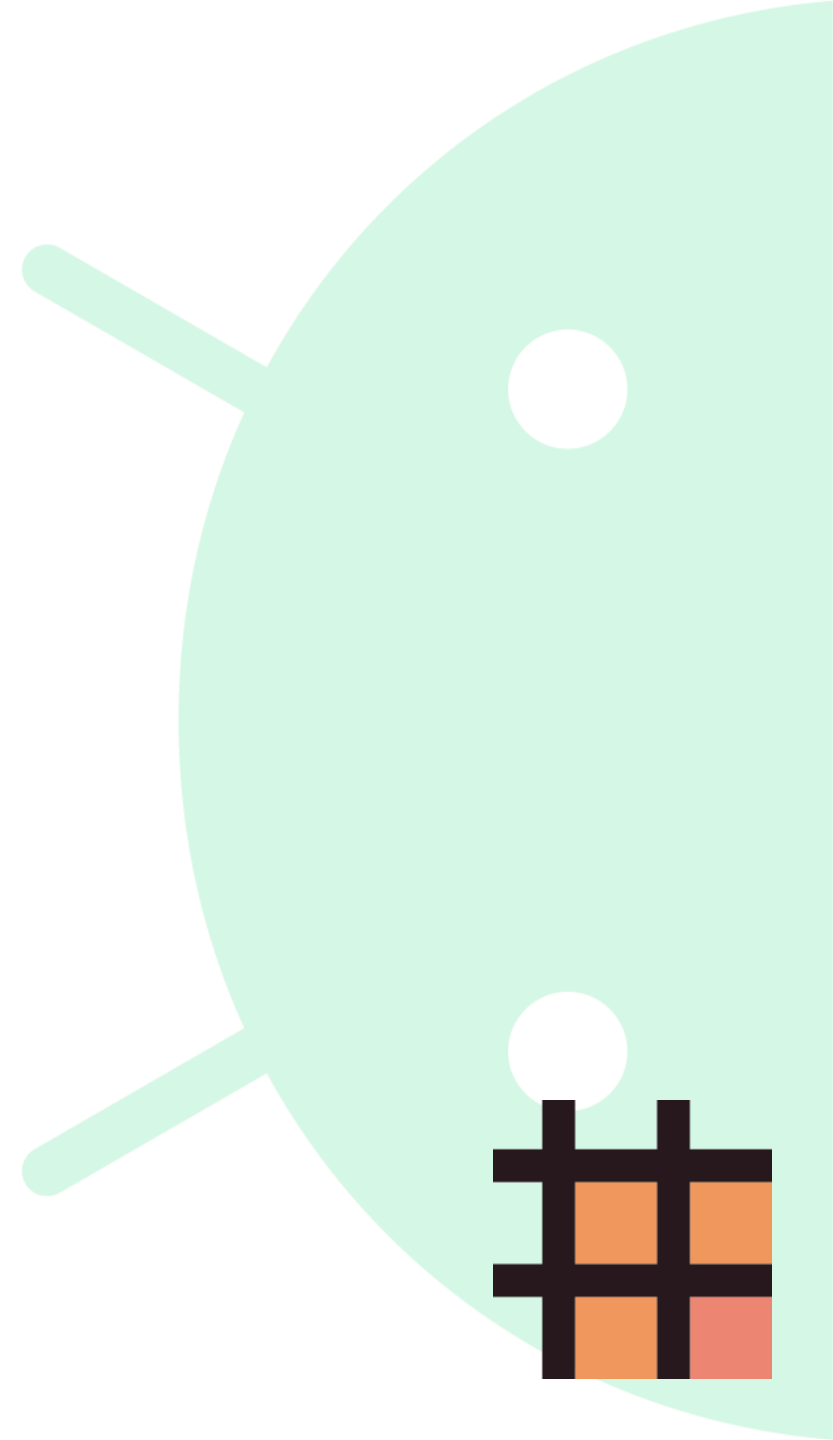
참고하면 매우매우매우 좋은 사이트

<https://developer.android.com/courses/kotlin-android-fundamentals/toc>

1.0 Install Android Studio ~ 2.3 Constraint layout using the Layout Editor



QA



감사합니다

Thanks to. 기존 세미나 자료를 참고할 수 있도록 해준 김상민(@sangggggg)

