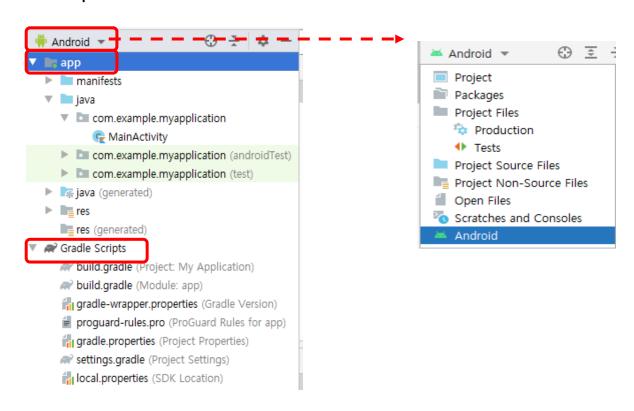
안드로이드 프로젝트 구조

프로젝트의 구성

- Android 표시 방식
 - app : 앱의 구조 및 소스와 관련된 파일이 위치하는 곳
 - Gradle Scripts : 빌드에 필요한 파일들이 위치하는 곳



AndroidManifest.xml

• 안드로이드 시스템이 앱을 실행할 때 필요한 기본적인 정보를 제공하는 역할

```
</ml version="1.0" encoding="utf-8"?>
   <manifest xmlns:android="http://schemas.android.com/apk/res/android"</pre>
       package="com.example.myapplication">→ 앱의 패키지 이름
       <application
                                            안드로이드 시스템에서 제공하는 백업 기능에 이 앱을 포함할지 여부
           android:allowBackup="true"
                                            True: 앱을 삭제하고 다시 설치할 때 기존 사용하던 정보를 복원
런처 아이콘 ▼android:icon="@mipmap/ic_launcher"
                                            False: 항상 새로 설치되고, 초기화된 상태로 실행
          _android:label="@string/app_name"
런처 라벨
          android:roundlcon="@mipmap/ic_launcher_round"
앱 이름
          android:supportsRt |="true" → 이슬람 문화권의 Right To Left(RTL)을 지원하는지 여부
          android:theme="@style/Theme.MyApplication">
<activity android:name=".MainActivity" android:exported=기본 (생상과 모양 결정
              <intent-filter>
                  <action android:name="android.intent.action.MAIN" />
  첫화면
                  <category android:name="android.intent.category.LAUNCHER" />
               </ri>
                               </activity>
       </application>
   </manifest>
```

Java 폴더

안드로이드 앱을 구성하는 java 소스 파일들이 패키지 명으로 분류되어 있음

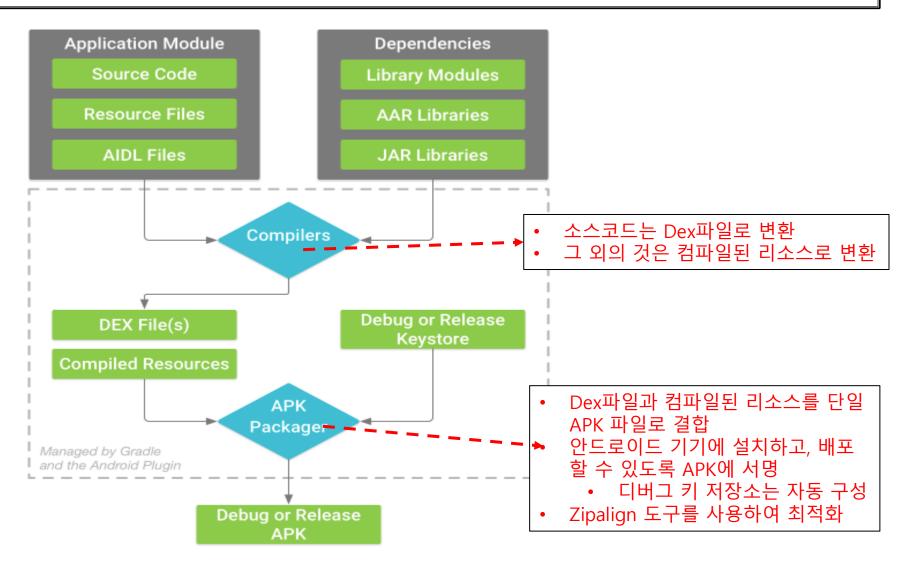


- com.example.myapplication
 - ু MainActivity ১০ মণ্ড
- ▶ com.example.myapplication (androidTest) ৩০০০০ লাবে আবার
- 🕨 🖿 com.example.myapplication (test) 🛮 🖽 এচ

res 폴더

- Res는 resource의 약자로, 소스 코드를 제외한 기타 파일들이 위치
 - Drawable
 - 이미지(.png, .jpg, .gif), 나인패치 이미지(.9.png), 또는 XML
 - R.drawable.파일이름 or @drawable/파일이름 다녀응수 ×
 - Layout
 - 액티비티 레이아웃을 정의하는 XML
 - R.layout.파일이름
 - Mipmap
 - 각기 다른 런처 아이콘 해상도에 대한 이미지 파일
 - R.mipmap.파일이름 or @mipmap/파일이름
 - Values
 - 코드와 레이아웃 모두에서 사용할 수 있는 공통 리소스를 정의한 XML
 - Colors.xml : 컬러값 정의 → @color/name
 - Strings.xml : 문자열 정의 → @string/name
 - Styles.xml : 스타일 정의 → @style/name

안드로이드 빌드 과정



https://developer.android.com/studio/build

Gradle Scripts

- Gradle
 - 복잡한 빌드 과정을 쉽게 적용하기 위한 안드로이드 공식 빌드 툴
 - 자신이 원하는 버전의 APK를 쉽게 생성하고, 여러가지 옵션을 담 아서 APK를 만들 수 있음
- Settings.gradle, Build.gradle
 - Gradle이 빌드를 시작하기 위해서 필요한 설정 정보

▼ # Gradle Scripts

- build.gradle (Project: My Application)
- w build.gradle (Module: app)
- 🚮 gradle-wrapper.properties (Gradle Version)
- proguard-rules.pro (ProGuard Rules for app)
- gradle.properties (Project Properties)
- settings.gradle (Project Settings)
- local.properties (SDK Location)

Settings.gradle

• 프로젝트 수준 저장소 설정 정의 및 앱을 빌드할 때 포함 해야 하는모듈을 Gradle에 알려줌

```
pluginManagement {
  repositories {
                         Gradle이 Gradle 플러그인 및 관련된 dependencies
    gradlePluginPortal()
                         를 탐색하거나 다운로드하기 위한 저장소
    google()
    mavenCentral()
dependencyResolutionManagement {
  repositoriesMode.set(RepositoriesMode.FAIL ON PROJECT REPOS)
  repositories {
    google()
                     프로젝트의 모든 모듈에서 사용할 저장소 및 dependencies
    mavenCentral()
rootProject.name = "My Application"
include ':app'
```

Build.gradle (Project)

Gradle Scripts

build.qradle (Project: My_Application)

프로젝트의 모든 모듈에 적용되는 종속 항목 정의

```
build.gradle (Module: My_Application.
plugins {
                                                                                                      📊 gradle-wrapper.properties (Gradle Vers
                                                                                                      proquard-rules.pro (ProGuard Rules fo
   id 'com.android.application' version '7.1.2' apply false
                                                                                                      gradle.properties (Project Properties)
                                                                                                      settings.gradle (Project Settings)
                                                                                                      local.properties (SDK Location)
   id 'com.android.library' version '7.1.2' apply false
   id 'org.jetbrains.kotlin.android' version '1.5.30' apply false
task clean(type: Delete) {
   delete rootProject.buildDir
                                      이전에 만들어진 apk 를 포함한 기타 파일을 삭제하고, 새롭게 빌드
```

Build.gradle (Module)

• 모듈빌드 파일로, 앱에 직접 적용될 빌드 구성을 정의

```
android {
                            컴파일에 사용할 API 버전 지정
   compileSdkVersion 32
   defaultConfig {
                                            패키지 네임
      applicationId "com.example.myapplication"
                           최소 지원 API
      minSdkVersion 29
                           가장 최적화된 버전 → 가장 안정적으로 동작하도록 테스트
      targetSdkVersion 32
      versionCode 1
                           앱의 버전 정보, 업데이트되면 값을 올려서 정보 표시
      versionName "1.0"
      testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
   buildTypes {
                  배포용 apk를 생성할 때 적용하는 옵션
      release {
          minifyEnabled false
          proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
                   난독화 적용 여부
                   난독화는 내부 코드를 쉽게 분석할 수 없게 만들어 주는 기능
```

Build.gradle (Module)

• 모듈빌드 파일로, 앱에 직접 적용될 빌드 구성을 정의

프로젝트에서 쓰이는 종속성을 정의

```
dependencies {
  implementation 'androidx.core:core-ktx:1.7.0'
  implementation 'androidx.appcompat:appcompat:1.6.1'
  implementation 'com.google.android.material:material:1.8.0'
  implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
  testImplementation 'junit:junit:4.13.2'
  androidTestImplementation 'androidx.test.ext:junit:1.1.5'
  androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
```

기본 위젯

위젯(Widgets) & 뷰(View)

뷰

- 화면에 보이는 각각의 것들
- 콘트롤이나 위젯이라고도 불리는 UI 구성 요소
- 버튼, 텍스트 필드, 체크 박스 등





Button

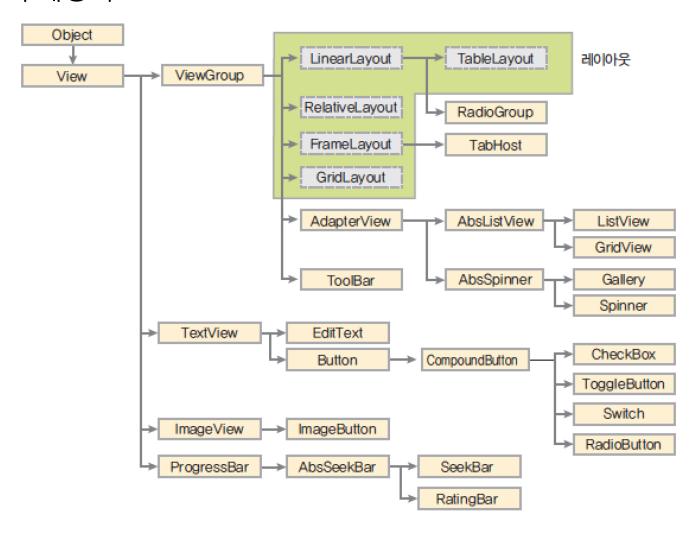
Checkbox

뷰그룹

- 뷰 들을 여러 개 담을 수 있는 컨테이너 기능
- 뷰 그룹도 뷰에서 상속받은 뷰임
- 레이아웃(Layout)이 해당됨

뷰(View)

• 뷰의 계층 구조



뷰

- View 클래스는 모든 뷰들의 부모 클래스
- 뷰의 필드와 메소드
 - id : 뷰의 식별자
 - Xml 레이아웃에 정의된 정보를 메모리 상에 객체로 만드는inflation 과정을 거쳐, 메모리 상에 만들어진 객체를 참조하기 위한 식별자

뷰의 위치와 크기

- 뷰의 위치와 크기
 - match_parent : 부모의 크기를 꽉 채운다
 - wrap_content : 뷰가 나타내는 내용물의 크기에 맞춘다
 - 숫자 : 크기를 정확히 지정한다

<Button

```
android:id="@+id/button1"
```

android:layout_width="wrap_content"

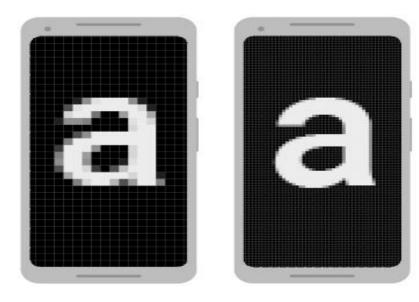
android:layout_height="wrap_content"

android:text="@string/button_text" />

버튼의 위치와 크기를 경정한다.

뷰의 크기 단위

- 뷰의 위치나 크기는 dp 사용
 - dp는 밀도 독립형 픽셀 (density independent pixel)을 측정 단위로 하는 가상 픽셀
 - px (pixels) : 화면 픽셀은 장치마다 화면의 밀도가 다르므로 권장되지 않음



- 텍스트의 크기는 sp를 사용
 - sp는 scalabel pixel로 기본적으로는 dp와 같지만 사용자가 선호하는 텍스트의 크기에 따라 크기가 조절됨

화면의 밀도

- 화면 밀도
 - 화면의 밀도 : Idpi, mdpi, hdpi, xhdpi, xxhdpi, xxxhdpi
 - 화면 밀도에 따른 이미지 제공되어야 함

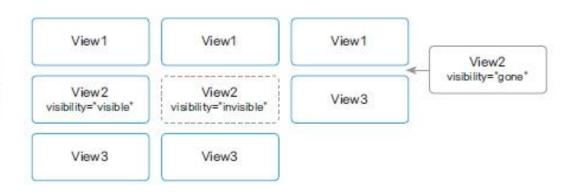
Qualifier	DPI	Scaling factor	Launcher icon	Action bar, tab icon	Small / Contextual Icons	Notification Icons
mdpi Baseline 1DP=1PX=1SP	160	1.0	48 x 48	32 x 32 Full Asset 24 x 24 Optical Square	24 x 24 Full Asset 22 x 22 Optical Square	16 x 16 Full Asset 12 x 12 Optical Square
hdpi	240	1.5	72 x 72 64 x 64	48 x 48 36 x 36	36 x 36 33 x 33	24 x 38 24 x 24
xhdpi	320	2.0	96 x 96 84 x 84	64 x 64 48 x 48	48 x 48 44 x 44	32 x 50 32 x 32
xxhdpi	480	3.0	144 x 144 126 x 126	96 x 96 72 x 72	72 x 72 66 x 66	48 x 75 48 x 48
xxxhdpi	640	4.0				

화면에 보이기 속성

- 뷰를 화면에 보이기 속성
 - visible : 화면에 보이게 한다, 디폴트 값
 - invisible : 표시되지 않는다. 그러나 공간을 차지한다
 - gone : 완전히 숨겨진다. 공간을 차지하지 않는다

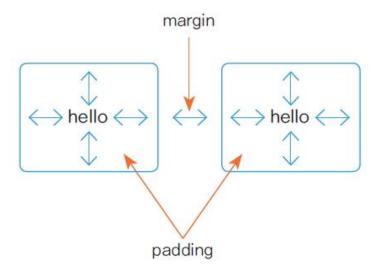
android:visibility="visible"

<Button android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="BUTTON2" android:visibility="invisible" />



뷰의 마진과 패딩

- 패딩 (Padding)
 - View 내부에서 View와 Content간의 여백
 - paddingLeft, paddingRight, paddingTop, paddingBottom
- 마진 (Margin)
 - View 외부 여백
 - layout_marginLeft, layout_marginRight, layout_marginTop, layout_marginBottom



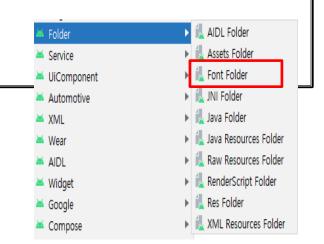
- 문자열을 출력하기 위한 뷰
- 주요 속성
 - android:text
 - 텍스트 뷰의 가장 중요한 속성으로 출력할 문자열을 지정
 - android:textColor
 - 문자열의 색상을 지정하며, 디폴트는 불투명한 밝은 회색
 - #RRGGBB나 #AARRGGBB 형식으로 각 색상 요소들의 강도를 지정
 - android:textSize
 - 텍스트의 폰트 크기를 지정
 - sp(scaled-pixel) 단위를 사용 : 예 15sp
 - android:textStyle
 - 폰트의 속성을 지정
 - normal, bold, italic 중 하나를 쓰거나 '|'로 묶어 두 개 이상의 상수 값을 지정 가능
 - ex: "bold|italic", "normal|italic"

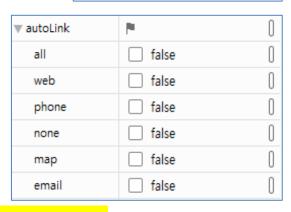
- 주요 속성
 - android:fontFamily
 - 화면에 출력할 문자열의 폰트 지정
 - 새로운 폰트 추가
 - res>>New>>Folder>>Font Folder 로 Font 폴더 생성 후 사용하고자
 하는 폰트를 복사해 넣음

 ▼ font
 - android:fontFamily ="@font/greenjoafont"
 - android:autoLink
 - Text안의 문자들 중에서 자동인식하여 Link화 함
 - android:autoLink="web|email" 등과 같이 사용
 - web, email, phone, map, all 등 사용가능
 - Phone의 경우 국가 코드 사용

This is a test.

Webpage address is at http://www.konkuk.ac.kr
Email me at im.greenjoa@gmail.com
Call me at +82-02-1234-1234





T greenjoafont.ttf

- 주요 속성
 - android:singleLine
 - 텍스트가 위젯의 폭보다 길 때 강제로 한 줄에 출력
 - 속성의 디폴트는 false로 폭보다 더 긴 줄은 자동으로 아래쪽으로 개행
 - Marquee 효과
 - 움직이는 글자로 singleLine과 같이 많이 사용하는 효과

```
android:singleLine="true"
android:ellipsize="marquee" *ellipsize:...로 표시
android:focusable="true"
android:focusableInTouchMode="true"
android:marqueeRepeatLimit="marquee_forever"
```

• 텍스트의 스타일 정의

</resources>

- res >> values >> styles.xml 파일에 스타일 추가
- TextView의 style에 스타일 이름 추가

```
<resources>
                                                                <TextView
                                                                      android:id="@+id/textView"
   <!-- Base application theme. -->
                                                                      style="@style/TextViewStyle "
   <style name="AppTheme" parent="Theme.AppCompat.Light.DarkAc</pre>
      <!-- Customize your theme here. -->
      <item name="colorPrimary">@color/colorPrimary</item>
      <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
      <item name="colorAccent">@color/colorAccent</item>
   </style>
   <!-- TextView Style -->
   <style name="TextViewStyle">
      <item name="android:textSize">20dp</item>
                                                                               TextView
      <item name="android:textColor">@color/colorPrimaryDark</item>
      <item name="android:padding">8dp</item>
   </style>
```

• Kotlin 코드에서는 TextView로 표현됨

```
val textView = findViewById<TextView>(R.id.textView)
```

EditText

• 문자열을 입력 받는 기능 수행하는 위젯

속성	설명
android:autoText	자동으로 <u>타이핑</u> 오류를 교정한다.
android:drawableBottom	텍스트의 아래에 표시되는 이미지 리소스이다.
android:drawableRight	텍스트의 오른쪽에 표시되는 이미지 리소스이다.
android:editable	편집가능
android:text	표시되는 텍스트이다.
android:singleLine	true이면 <u>한중만</u> 받음
android:inputType	입력의 종류
android:hint	입력 필드에 표시되는 힌트 메시지

EditText

• InputType 속성 → 다양한 형태의 EditText 존재

inputType	설명
none	편집이 불가능한 문자열
Text	일반적인 문자열
textMultiLine	여러 줄로 입력 가능
textPostalAddress	우편번호
textEmailAddress	이메일 주소
textPassword	패스워드
${ m textVisible Password}$	패스워드 화면에 보인다.
number	숫자
$\operatorname{numberSigned}$	부호가 붙은 숫자
${\bf number Decimal}$	소수점이 있는 숫자
phone	전화번호
datetime	시간

EditText

- 소프트 키보드에 레이아웃이 영향을 받게 됨
 - 액티비티에 이 모드를 설정할 수 있음

<activity

android:windowSoftInputMode="stateVisible|adjustResize" ... >

https://developer.android.com/guide/topics/manifest/activity-element.html#wsoft

- stateHidden
 - 소프트 키보드가 hidden 상태가 되게 한다.
- stateVisible
 - 소프트 키보드가 visible 상태가 되게 한다.
- adjustNothing
 - 레이아웃에 영향을 주지 않는다.
- adjustResize
 - 소프트 키보드에 맞춰서 화면이 resize 된다.
- adjustPan
 - EditText 부분이 화면에 보이도록 레이아웃이 위로 스크롤 된다.





Button

- 클릭할 수 있는 위젯
- 주요 속성

android:clickable="bool"	버튼의 클릭 여부 설정
android:id="@+id/ID"	자바 코드에서 사용하기 위한 ID
android:onClick="function"	클릭할 때 액티비티에서 호출할 함수
android:text="text"	버튼내의 텍스트

• Kotlin 코드에서는 Button으로 표현됨

val button = findViewById<Button>(R.id.button)

Button

• Button의 XML

```
<Button
android:id="@+id/basic_button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Basic Button"
/>
```

```
<RadioGroup
android:id="@+id/RadioGroup01"
android:layout_width="wrap_content"
android:layout_height="wrap_content">
<RadioButton
android:id="@+id/RadioButton01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Option 1"></RadioButton>
<RadioButton
android:id="@+id/RadioButton02"
android:layout_width="wrap_content"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Option 2"></RadioButton>
</RadioGroup>
```

```
<CheckBox
    android:id="@+id/checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Check me?"
/>
```

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/image_button"
    android:src="@drawable/droid">
</ImageButton>
```

```
<ToggleButton
    android:id="@+id/toggle_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Toggle"
    android:textOff="Disabled"
    android:textOn="Enabled"
/>
```

ImageButton

- 이미지가 출력되는 버튼
- 주요속성

android:clickable="bool"	버튼의 클릭 여부 설정	
android:id="@+id/ID"	자바 코드에서 사용하기 위한 ID	
android:onClick="function"	클릭할 때 액티비티에서 호출할 함수	
android:src="@drawable/img"	버튼에 출력될 이미지	
app:srcCompat ="@drawable/img"	Vector drawable 이용	

- Image Resource 설정
 - app/src/main/res/drawable 폴더에 이미지 파일을 넣어둠
 - /res/drawable-24: API 레벨 24 이상에서 쓰이는 이미지 폴더
 - 해상도에 따른 이미지 자동 생성
 - 해당하는 이미지가 존재하지 않을 경우 drawable 폴더 참조
 - test.png 파일→ @drawable/test 로 참조하면 됨
 - 파일 이름은 소문자와 숫자의 조합으로 나타냄

ImageButton

```
ImageButton
android:layout_width= "100dp"
android:layout_height= "50dp"
android:src="@android:drawable/sym_action_call" />
ImageButton
android:layout_width= "wrap_content"
android:layout_height= "wrap_content"
android:src="@android:drawable/sym_action_chat"/>
ImageButton
android:layout_width= "wrap_content"
android:layout_height= "wrap_content"
android:src="@android:drawable/sym_action_email" />
```

* Kotlin 코드에서 이미지 변경

imageButton.setImageResource(R.drawable.image)

이벤트 처리 방식

- 사용자 이벤트
 - 델리게이션 이벤트 모델 (Delegation Event Model)
 - 뷰에서 발생하는 이벤트를 처리하기 위한 모델
 - 이벤트의 목적을 보다 명확하게 처리하기 위함
 - 단순 터치로 인식하는 것이 아님, 예) checkbox 선택/해제 (CheckedChangeEvent)
 - 이벤트 소스와 이벤트 핸들러를 리스너(Listener)로 연결하여 처리함



- Hierarchy 이벤트 모델 (Hierarchy Event Model)
 - 액티비티가 화면에 출력되었을 때 발생하는 사용자의 키 이벤트와 화면 터치 이벤트
 를 처리하기 위한 모델
 - 이벤트 발생시 자동 호출되는 함수만 액티비티 내에 재정의하면 됨

Delegation 이벤트 구조

• Button객체의 ClickEvent 예제

```
Button basic_button = (Button) findViewById(R.id.basic_button);
basic_button.setOnClickListener ( new View.OnClickListener() {
    public void onClick(View v) {
        Toast.makeText(this, "Button clicked", Toast.LENGTH_SHORT).show();
    }
}
```

basic_button

setOnClickListener

OnClickListener

주요 Delegation 이벤트

이벤트	설명		
OnClickListener	뷰 클릭 시 발생하는 이벤트		
OnLongClickListener	뷰를 오래 클릭했을 때 발생하는 이벤트		
OnCheckedChangeListener	CheckBox의 상태 변경 이벤트		
OnItemClickListener	ListView의 항목 선택 이벤트		
OnDateSetListener	DatePicker의 날짜 선택 이벤트		
OnTimeSetListener	TimePicker의 시간 선택 이벤트		

Button: OnClickListener 1

- Button의 OnClickListener
 - Toast를 이용하여 화면에 표시하기

Java

```
Button basic_button = (Button) findViewById(R.id.basic_button);
basic_button.setOnClickListener ( new View.OnClickListener() {
    public void onClick(View v) {
        Toast.makeText(this, "Button clicked", Toast.LENGTH_SHORT).show();
    }
}
```

```
val basic_button = findViewById < Button > (R.id.basic_button) Kotlin
basic_button.setOnClickListener{
   Toast.makeText(this, "Button Click", Toast.LENGTH_LONG).show()
}
```

*메소드 하나만 있는 경우 → 람다가 메소드 대체함

```
public class FirstProjActivity extends Activity implements OnClickListener{
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
     Button bt = (Button)findViewById(R.id.start);
    Button bt1 = (Button)findViewById(R.id.start2);
     bt.setOnClickListener(this);
                                                               Java 코드
    bt1.setOnClickListener(this);
public void onClick(View v) {
    // TODO Auto-generated method stub
    switch (v.getId()){
    case R.id.start:
         Toast.makeText(this, R.string.start_string, Toast.LENGTH_SHORT).show();
         break;
    case R.id.start2:
         break;
    default:
         break;
```

Kotlin 코드

```
class MainActivity: AppCompatActivity(), View.OnClickListener {
   override fun onClick(v: View?) {
     when (v?.id){
                              ?. : 변수가 널이 아닐때 만 id 접근
        R.id.btn1 -> { }
        R.id.btn2 -> { }
  override fun onCreate(savedInstanceState: Bundle?) {
     super.onCreate(savedInstanceState)
     setContentView(R.layout.activity_main)
      btn1.setOnClickListener(this)
      btn2.setOnClickListener(this)
```

```
public class FirstProjActivity extends Activity{
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    Button bt = (Button)findViewById(R.id.start);
                                                             Java 코드
    Button bt1 = (Button)findViewById(R.id.start2);
    bt.setOnClickListener(new MyHandler());
class MyHandler implements OnClickListener
    public void onClick(View v) {
    // TODO Auto-generated method stub
```

• Kotlin 코드

```
class MyHandler : View.OnClickListener{
   override fun onClick(v: View?) {
      when (v?.id) {
        R.id.btn1 -> { }
        R.id.btn2-> { }
    }
}
btn1.setOnClickListener(MyHandler())
btn2.setOnClickListener(MyHandler())
```

```
public class FirstProjActivity extends Activity{
    public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
                                                   Java 코드
    public void fire(View v){
<Button
    android:id="@+id/test1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/test_string"
    android:onClick="fire" />
```

• Kotlin 코드

```
class MainActivity : AppCompatActivity(){
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity main)
    fun fire(v:View?):Unit{
<Button
   android:id="@+id/test1"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:text="@string/test_string"
   android:onClick="fire" />
```

Hierarchy 이벤트 모델

- 직접 터치 이벤트와 키 이벤트를 처리할 때 사용
- 액티비티 클래스내에 이벤트에 대한 콜백 함수를 정의하면 됨
 - Delegation 모델과 달리 이벤트소스와 핸들러를 연결하여 처리하는 구조가 아님

```
override fun onTouchEvent(event: MotionEvent?): Boolean {
}
```

- TouchEvent
 - MotionEvent
 - ACTION_DOWN : 화면에 터치된 순간의 이벤트
 - ACTION_UP : 터치를 떼는 순간의 이벤트
 - ACTION_MOVE : 터치한 후 이동하는 순간 이벤트
 - Event의 발생 위치
 - X / Y : 터치 이벤트가 발생한 뷰 내에서의 좌표값
 - rawX / rawY : 터치 이벤트가 발생한 화면에서의 좌표값

Hierarchy 이벤트 모델

```
var posX:Float=0.0f
override fun onTouchEvent(event: MotionEvent?): Boolean {
   when(event?.action){
       MotionEvent.ACT/ON_DOWN ->{
           posX = event. rawX
       MotionEvent.ACT/ON_UP ->{
           val distX = posX - event. rawX
           if(distX > 0){
               Toast.makeText(this, "왼쪽으로 밀었음",
                                       Toast.LENGTH_SHORT).show()
           }else if(distX <0){</pre>
               Toast.makeText(this, "오른쪽으로 밀었음",
                                      Toast. LENGTH_SHORT). show()
    return true
```

Hierarchy 이벤트 모델

- KeyEvent
 - 소프트 키보드(Soft Keyboard)가 아닌 하드웨어 키보드가 눌렸을 때의 키이벤트 (홈, 뒤로가기, 오버뷰 버튼)
 - 홈, 오버뷰 버튼은 이벤트로 제어할 수 없음
 - 주로 뒤로가기 버튼 처리할 때 사용함

```
var initTime:Long=0
override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {
  when(keyCode){
    KeyEvent.KEYCODE_BACK->{
        if((System.currentTimeMillis() - initTime) > 3000){
            Toast.makeText(this, "종료하려면 한번 더 누르세요", Toast.LENGTH_SHORT).show()
            initTime = System.currentTimeMillis()
        }else{
            finish()
        }
        return true
    }
}
return super.onKeyDown(keyCode, event)
}
```

Image

- ImageView
 - Image를 넣을 수 있는 TextView
- ImageButton
 - Image를 넣을 수 있는 Button

- Image를 넣는 방법
 - android:src / app:srcCompat 속성에 Image 파일을 지정
 - setImageResource(R.drawable.id) 로 이미지 변경

ImageView

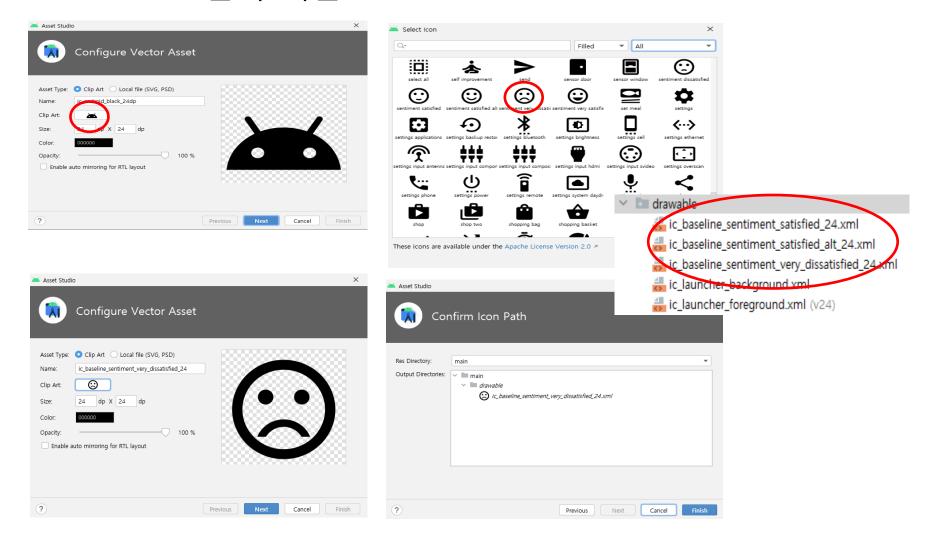
• 주요 속성

속성	설정 메소드	설명
android:adjustViewBounds	setAdjustViewBounds(boolean)	drawable의 종횡비를 유지하기 위하여 이미지 뷰의 가로, 세로를 조정
android:cropToPadding		true이면 패딩 안에 맞추어서 이미지를 자른다.
android:maxHeight	setMaxHeight(int)	이미지 뷰의 최대 높이
android:maxWidth	setMaxWidth(int)	이미지 뷰의 최대 너비
android:scaleType	setScaleType(ImageView. ScaleType)	이미지 뷰의 크기에 맞추어 어떻게 확대 나 축소할 것인지 방법 선택
android:src	setImageResource(int)	이미지 소스
android:tint	setColorFilter(int, PorterDuff, Mode)	이미지 배경 색상

^{*}scaleType: https://developer.android.com/reference/android/widget/ImageView.ScaleType.html

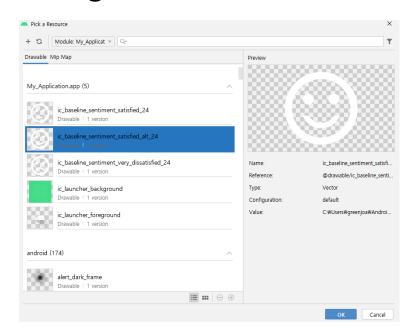
벡터 이미지 추가하기

• Res >> 오른쪽 버튼 >> New >> Vector Asset



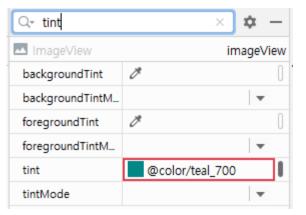
ImageView 추가 및 색 적용

ImageView 추가 >> 벡터 이미지 선택

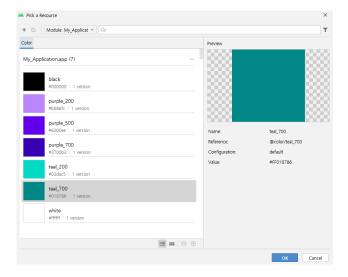


- ImageView의 tint 속성 변경
 - -. app:tint 속성 지정

app:tint="@color/teal_700"



app:tint 속성변경



토스트 (Toast)

• 화면에 잠깐 보이다가 사라지는 메시지

```
var t = Toast.makeText(this, "Hello World!", Toast.LENGTH_LONG)
t.show()
Toast.makeText(this, "Hello World!", Toast.LENGTH_LONG).show()
```

open static fun makeText(context: <u>Context!</u>, text: <u>CharSequence!</u>, duration: Int

): <u>Toast!</u>

- Context
 - Application과 관련된 정보
 - 시스템 함수
- Duration (임의의 값을 지정할 수 없음)

• LENGTH_LONG : 5초 • LENGTH_SHORT : 3초

View Binding

- 뷰와 상호작용하는 코드를 쉽게 작성하는 기능
 - 레이아웃에 ID가 있는 모든 뷰의 직접 참조가 가능하게 함
 - 뷰 객체를 참조하기 위해 findViewByID 사용했던 것 대체 함
 - Null safety : 유효하지 않은 뷰 ID로 인한 null 포인터 발생 위험 없음
 - Type safety: XML 파일의 뷰 타입과 일치하지 않는 위험 없음
- 모듈별 뷰 바인딩 설정 (build.gradle)

```
android {
...
buildFeatures {
viewBinding true
}
}
```

private lateinit var binding: ActivityMainBinding

```
override fun onCreate(savedInstanceState: Bundle) {
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    val view = binding.root
    setContentView(view)
}
```

RadioButton

- RadioGroup안에 RadioButton을 배치함
 - RadioButton의 Checked 속성을 true/false로 설정

```
<RadioGroup
android:id="@+id/RadioGroup01"
android:layout width="wrap content"
android:layout height="wrap content">
<RadioButton
android:id="@+id/RadioButton01"
android:layout_width="wrap_content"
android:layout height="wrap content"
android:text="Option 1"></RadioButton>
<RadioButton
android:id="@+id/RadioButton02"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Option 2"></RadioButton>
</RadioGroup>
```

RadioButton

RadioButton□ OnCheckedChangeListener

```
RadioGroup group = (RadioGroup)findViewById(R.id. RadioGroup01);
group.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
public void onCheckedChanged(RadioGroup group, int checkedId) {
         TextView tv = (TextView) findViewById(R.id. TextView01);
         if (checkedId != -1) {
                  RadioButton rb = (RadioButton) findViewById(checkedId);
                  if (rb != null) {
                            tv.setText("You chose: " + rb.getText());
         } else {
                                                                      Java 코드
                  tv.setText("Choose 1");
```

RadioButton

• RadioButton의 OnCheckedChangeListener
Kotlin 코드

```
val imgGroup = findViewById<RadioGroup>(R.id.imgGroup)
imgGroup.setOnCheckedChangeListener{
    __, checkedId ->
        when(checkedId){
        // do something
    }
}
```

* RadioGroup 인수 사용하지 않을 때 : _

Button

• ToggleButton의 **OnCheckedChangeListener** Java 코드

```
ToggleButton btn1 = (ToggleButton)findViewByld(R.id.btn1);
btn1.setOnCheckedChangeListener(
                   new CompoundButton.OnCheckedChangeListener() {
   @Override
  public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
     if(isChecked)
       Toast.makeText(getApplicationContext(), "Toggle On", Toast.LENGTH SHORT).show();
     else
       Toast.makeText(getApplicationContext(), "Toggle Off", Toast.LENGTH_SHORT).show();
```

Button

ToggleButton의 OnCheckedChangeListener

```
toggleButton.setOnCheckedChangeListener{
    _, isChecked ->
    if(isChecked)
        Toast.makeText(this, "Toggle On", Toast.LENGTH_SHORT).show();
    else
        Toast.makeText(this, "Toggle Off", Toast.LENGTH_SHORT).show();
}
```

Selector

• 클릭 이벤트가 사용되는 View에 적용가능

- android:state_enabled
 - 사용가능 상태
- android:state_pressed
 - 클릭 상태
- android:state_focused
 - 포커스 상태
- android:state_selected
 - 현재 선택된 경우
- android:state_checked
 - 체크된 상태

```
* Drawable폴더 >> New >> Drawable Resource file
```

/drawable/btnstyle.xml

```
</multiple>/> encoding="utf-8"?>
<selector
xmlns:android="http://schemas.android.com/apk/res/android">
 <item
     android:state checked="true"
     android:drawable="@drawable/toggle_button_on"/>
 <item
     android:state_checked="false"
     android:drawable="@drawable/toggle_button_off"/>
</selector>
<ToggleButton
    android:background="@drawable/btnstyle"
    android: layout_width="wrap_content"
    android: layout_height="wrap_content"
    android:text0n=""
    android:textOff="" | *속성 창:@null 사용
/>
```

수고하셨습니다.