

Ch02. 처음 만드는 안드로이드 애플리케이션

모바일앱프로그래밍
김지심

학습목표



기본적인 안드로이드 애플리케이션을 작성할 수 있다.
안드로이드 프로젝트의 구성을 이해한다.

목차

1. 안드로이드 애플리케이션 작성
2. 프로젝트 구성 살펴보기
3. 간단한 앱 만들어보기



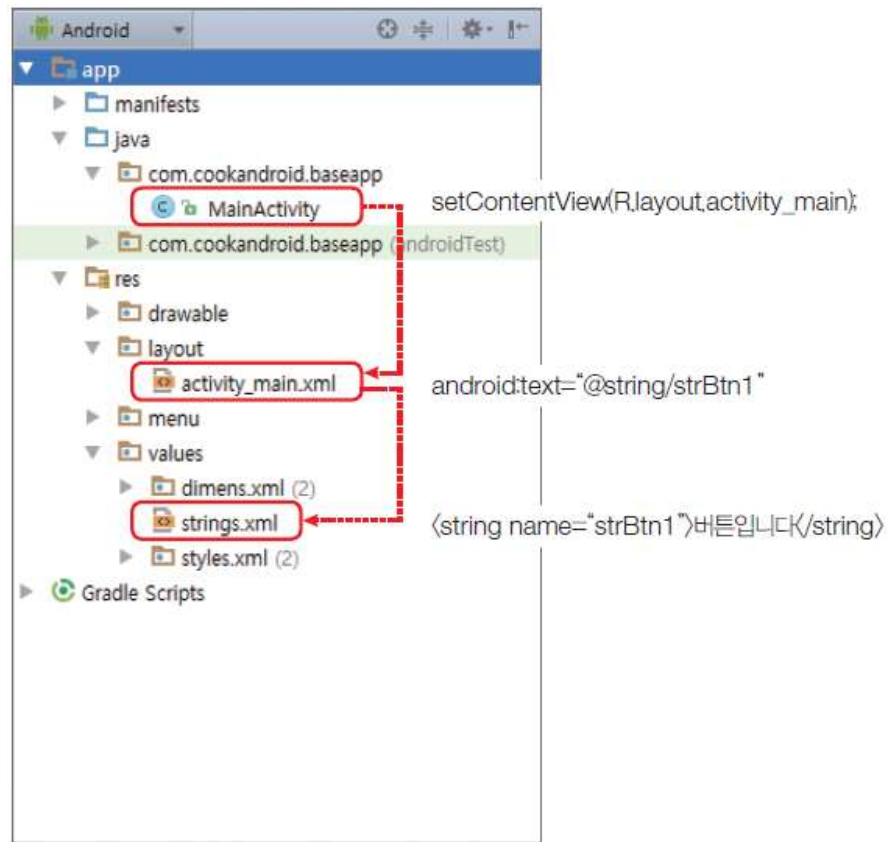
02



프로젝트 구성 살펴보기



2. 프로젝트 구성 살펴보기



[Android] 뷰

논리적 뷰

[Project] 뷰

탐색기와 동일한 물리적 뷰

그림 2-65 안드로이드 프로젝트 구성

2. 프로젝트 구성 살펴보기

프로젝트 구성

java 폴더

하위에 안드로이드 프로젝트를 생성할 때 입력한 패키지명 하에 MainActivity.java

res 폴더

앱 개발에 사용되는 이미지, 레이아웃, 문자열 등의 리소스가 저장되는 폴더

이미지 파일은 drawable 폴더에 넣음

layout 폴더: 액티비티(화면)을 구성하는 xml

values 폴더: styles.xml, 문자열을 저장하는 string.xml, 화면 구성 자원의 크기(여백, 글자크기)를 정의하는 dimens.xml 등

menu 폴더: 메뉴 xml

mipmap 폴더: 화면 밀도별 아이콘

manifests 폴더

AndroidManifest.xml 파일이 들어 있는데, 앱의 전반적인 정보를 담고 있음

2. 프로젝트 구성 살펴보기

Gradle Scripts 폴더

Gradle 빌드 방법이 정의된 스크립트

build.gradle (Project: 프로젝트명) : 모듈 전체에 적용되는 빌드 정보

build.gradle (Module: app) : 각 모듈에 적용되는 빌드파일

proguard-rules.pro: 코드 최적화, 난독화(obfuscate) 등 명시

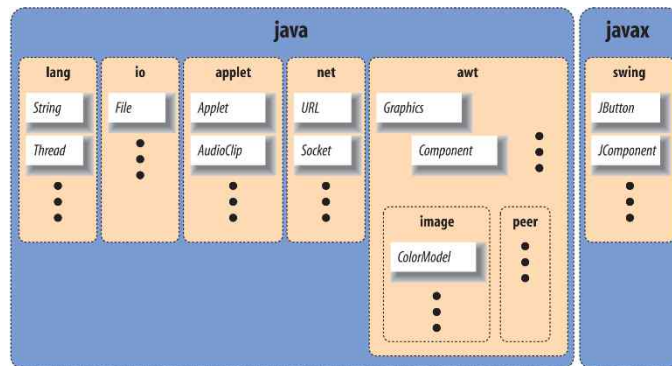
gradle.properties: 빌드 관련 속성

setting.gradle: 빌드 관련환경설정, 빌드될 하위 모듈 정보 포함

Local.properties: 빌드 시 필요한 환경변수 정보 명시(안드로이드 SDK 경로 등)

2. 프로젝트 구성 살펴보기

패키지(package): 클래스들을 보관하는 컨테이너
com.android.kjs.프로젝트명



AppCompatActivity

Summary
Meth

```
public class AppCompatActivity  
extends FragmentActivity implements AppCompatActivity, TaskStackB  
ActionBarDrawerToggle.DelegateProvider
```

[java.lang.Object](#)

↳ [android.content.Context](#)

↳ [android.content.ContextWrapper](#)

↳ [android.view.ContextThemeWrapper](#)

↳ [android.app.Activity](#)

↳ [android.support.v4.app.FragmentActivity](#)

↳ [android.support.v7.app.AppCompatActivity](#)

2. 프로젝트 구성 살펴보기

MainActivity.java

```
package com.android.kjs.a01_01_firstandroid;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

MainActivity: Activity 클래스로 생성됨. Activity: 화면을 가지는 하나의 작업
onCreate(): 액티비티가 생성되는 순간 한번 호출됨, 초기화, UI 설정 등 구현
savedInstanceState: 번들 타입의 파라미터로 애플리케이션이 이전에 실행되었던 상태를 전달
setContentView(): activity.xml(액티비티의 화면)을 화면에 출력
@Override: '@' 메소드가 재정의되었음을 컴파일러에게 전달하는 annotation 기호

2. 프로젝트 구성 살펴보기

리스너

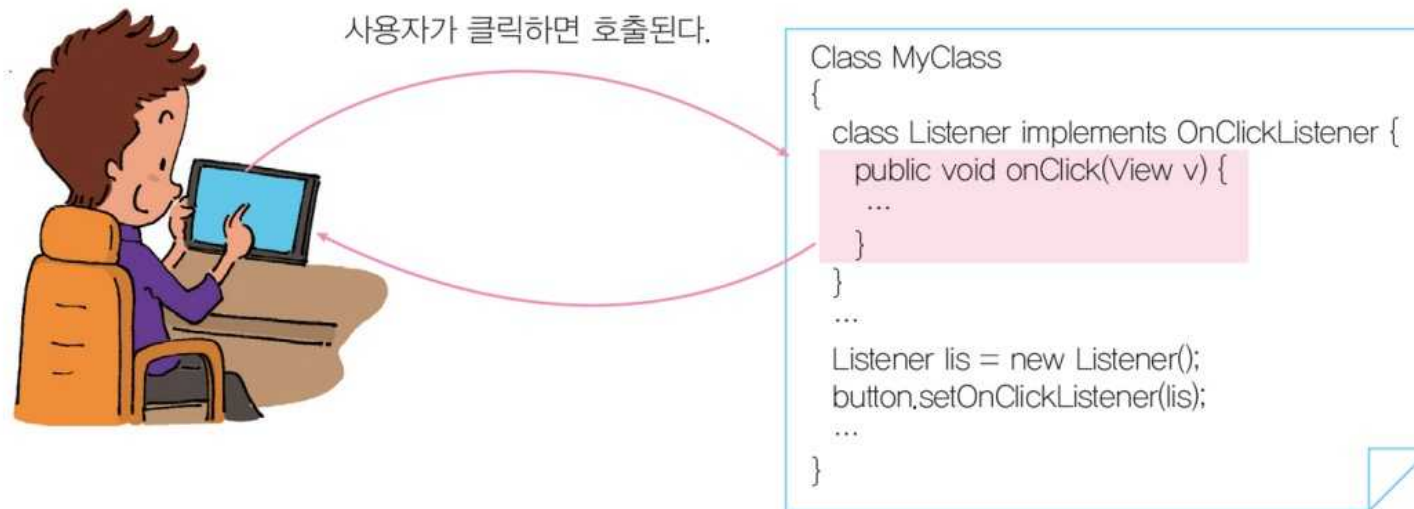
이벤트를 처리하는 메소드들이 정의된 인터페이스(event listener)

이벤트 리스너를 구현하는 클래스를 정의하고 이 클래스의 객체를 생성하여 뷰에 등록

이벤트가 발생하면 등록된 리스너 안의 콜백 메소드가 자동으로 호출됨

2. 프로젝트 구성 살펴보기

버튼에 붙은 리스너 객체가 이벤트를 처리함



2. 프로젝트 구성 살펴보기

리스너의 기본 정의

```
class MyClass
{
    class Listener implements OnClickListener {
        public void onClick(View v){
            ...
        }
    }
    ...
    Listener lis = new Listener();
    button.setOnClickListener(lis);
    ...
}
```

리스너 인터페이스를
구현한 클래스 정의

이벤트 리스너
객체 생성

버튼에 이벤트
리스너 객체를 등록

2. 프로젝트 구성 살펴보기

리스너의 기본 종류

리스너	콜백 메소드	설명
View.OnClickListener	<code>onClick()</code>	사용자가 어떤 항목을 터치하거나 내비게이션 키나 트랙볼로 항목으로 이동한 후에 엔터키를 눌러서 선택하면 호출된다.
View.OnLongClickListener	<code>onLongClick()</code>	사용자가 항목을 터치하여서 일정 시간 동안 그대로 누르고 있으면 발생한다.
View.OnFocusChangeListener	<code>onFocusChange()</code>	사용자가 하나의 항목에서 다른 항목으로 포커스를 이동할 때 호출된다.
View.OnKeyListener	<code>onKey()</code>	포커스를 가지고 있는 항목 위에서 키를 눌렀다가 놓았을 때 호출된다.
View.OnTouchListener	<code>onTouch()</code>	사용자가 터치 이벤트로 간주되는 동작을 한 경우에 호출된다.

2. 프로젝트 구성 살펴보기

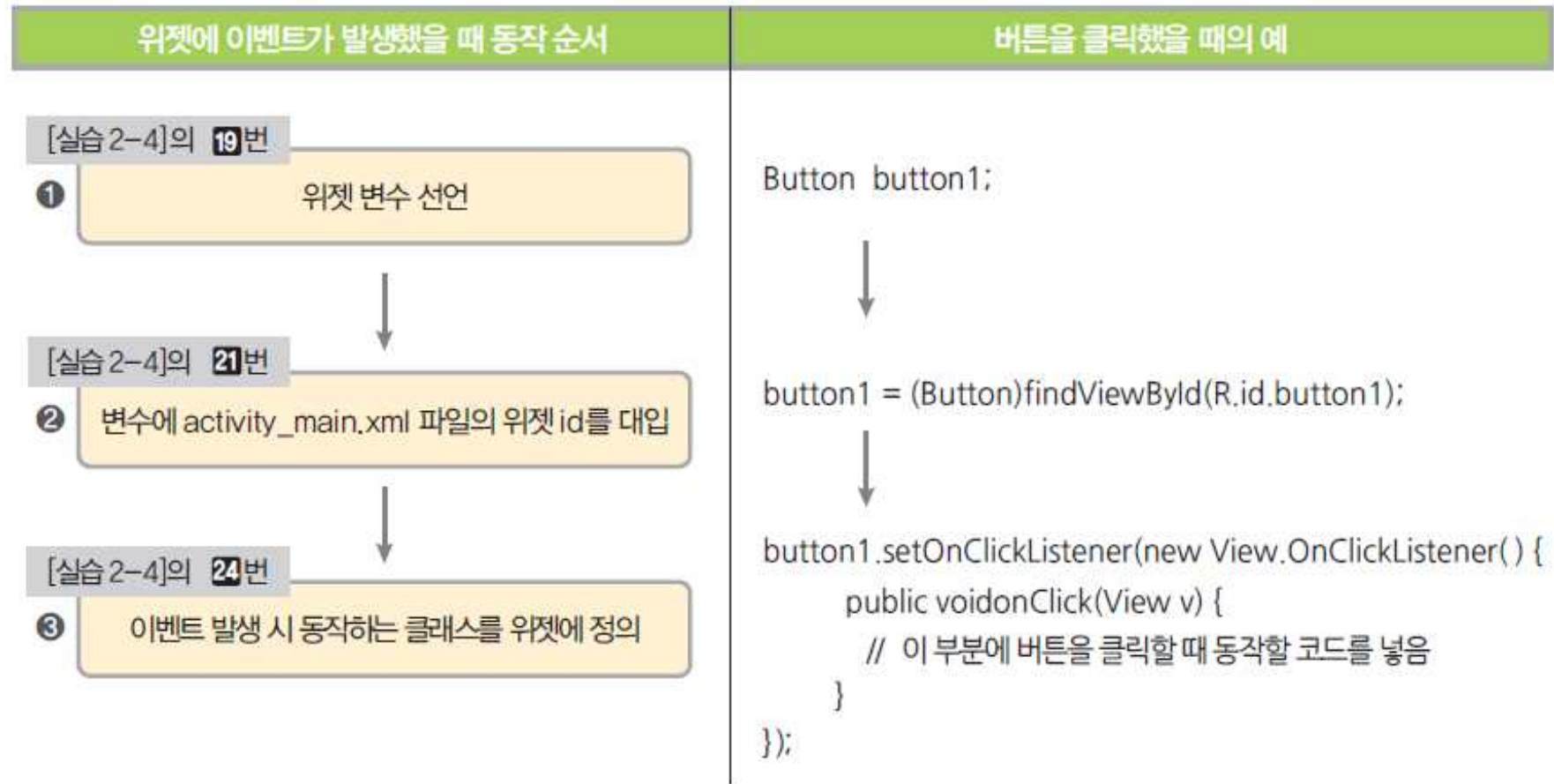


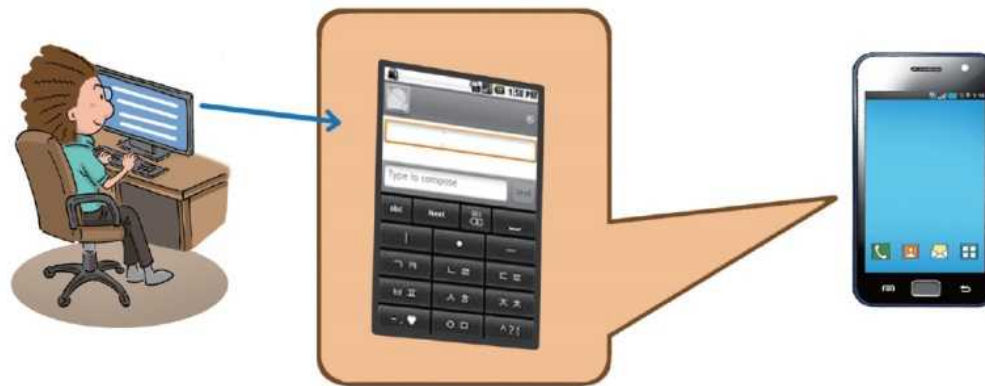
그림 2-63 위젯에 이벤트를 작동하기 위한 코딩 요약

2. 프로젝트 구성 살펴보기

클래스의 정의

Activity(AppCompatActivity)로부터 상속받았으므로 액티비티가 됨

액티비티: 안드로이드에서 애플리케이션을 구성하는 4가지의 컴포넌트 중의 하나



액티비티는 화면을 통하여 사용자와 상호작용하는 활동을 의미한다.

2. 프로젝트 구성 살펴보기

애플리케이션은 컴포넌트로 이루어짐

액티비티(Activity): 사용자 인터페이스 화면을 가지는 하나의 작업

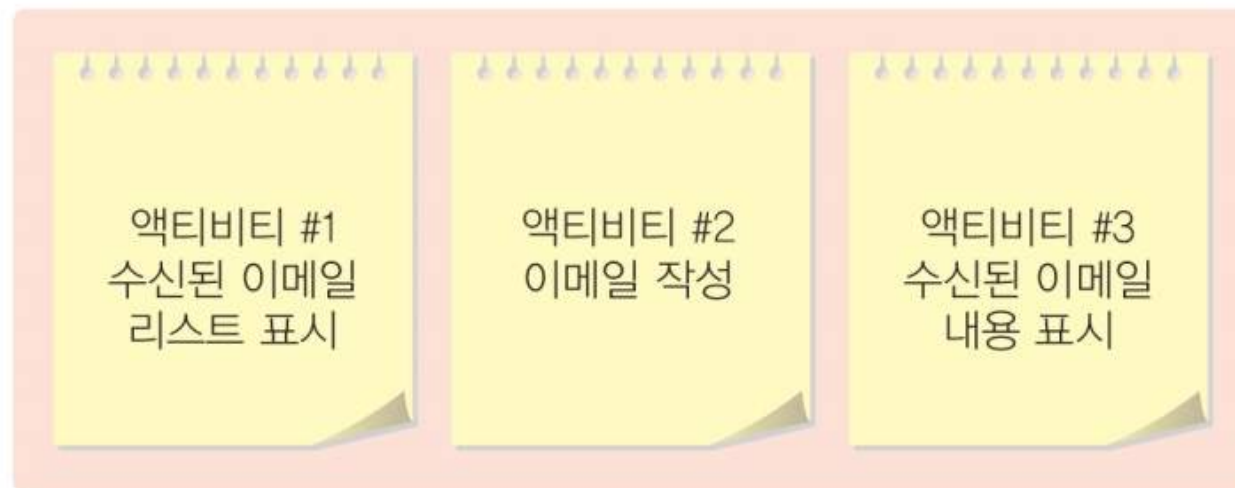
서비스(Service)

방송 수신자(BroadcastReceiver)

콘텐츠 제공자(ContentProvider)

2. 프로젝트 구성 살펴보기

액티비티들이 모여서 애플리케이션이 됨



이메일 애플리케이션

2. 프로젝트 구성 살펴보기

XML(Extensible Markup Language)

a simple text-based format for representing structured information: documents, data, configuration, books, transactions, invoices, and much more

(출처: <https://www.w3.org>)

2. 프로젝트 구성 살펴보기

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      xmlns:app="http://schemas.android.com/apk/res-auto"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:orientation="vertical"
9      android:gravity="center"
10     tools:context="kr.ac.mjc.kjs.ch02_baseapp.MainActivity">
11
12     <Button
13         android:id="@+id/button"
14         android:layout_width="match_parent"
15         android:layout_height="wrap_content"
16         android:text="@string/strBtn1"
17         tools:text="Click BTN" />
18
19 </LinearLayout>
```

2. 프로젝트 구성 살펴보기

XML 형식

요소(element): 시작 태그로 시작되어 종료 태그로 끝나는 논리적인 구성 요소

ex. `<Greeting>Hello, world.</Greeting>`

속성(attribute): 요소의 속성으로서 “이름=값”의 쌍으로 구성

ex. ``에서
img 태그(엘리먼트)는 src와 alt라는 2개의 속성을 가짐

2. 프로젝트 구성 살펴보기

접두어

namespace

xml 내에서 유일한 엘리먼트명이나 속성명을 제공하기 위한 이름공간

Prefix

xmlns:android

안드로이드 기본 SDK에 포함되어 있는 속성을 사용

xmlns:app

프로젝트에서 사용하는 외부 라이브러리에 포함되어 있는 속성을 사용

xmlns:tools

안드로이드 스튜디오의 디자이너 도구 등에서 화면에 보여줄 때 사용.
(앱이 실행될 때는 적용되지 않고 안드로이드 스튜디오에서만 적용됨)

2. 프로젝트 구성 살펴보기

xml은 기본적으로 뷰 엘리먼트와 속성으로 구성됨

<시작 태그

속성1="속성값1"

속성2="속성값2"

...

</끝 태그>

속성Prefix : 속성명1="속성값1"

android:layout_width="wrap_content"

app:layout_constraintLeft_toLeftOf="parent"

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Hello World!" />

<android.support.constraint.ConstraintLayout

android:layout_width="match_parent"

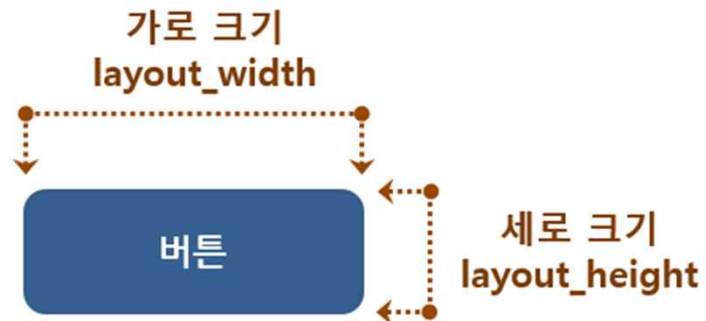
android:layout_height="match_parent"

...

</android.support.constraint.ConstraintLayout>

2. 프로젝트 구성 살펴보기

가로 크기와 세로 크기는 뷰의 필수 속성
속성 제거 시 에러가 표시됨



```
Test - [C:\Users\User\AndroidStudioProjects\Book1-4\Test] - [app] - ...\Wapp\src\main\res\layout\activity_main.xml - Android Stu...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Test app src main res layout activity_main.xml
activity_main.xml x
android.support.constraint.ConstraintLayout TextView
  xmlns:tools="http://schemas.android.com/tools"
  android:id="@+id/activity_main"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context="org.androidtown.test.MainActivity">
  <TextView
    layout_width' attribute should be defined more... (Ctrl+F1)
    android:layout_height="wrap_content"
    android:text="Hello World!"
    tools:layout_constraintTop_creator="1"
    tools:layout_constraintRight_creator="1"
    tools:layout_constraintBottom_creator="1"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    tools:layout_constraintLeft_creator="1"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    </pre>
```

2. 프로젝트 구성 살펴보기

기본 속성

java에서 참조하기 위해 id는 필수

android:id	TextView 요소에 유일한 아이디를 할당한다. 이 아이디를 이용하여서 소스 코드에서 이 텍스트 뷰를 참조할 수 있다.
android:layout_width	화면에서 얼마나 폭을 차지할 것인지를 정의한다. "match_parent"는 전체 화면의 폭을 다 차지하는 것을 의미한다.
android:layout_height	화면에서 길이를 얼마나 차지할 것인지를 정의한다.
android:text	화면에 표시하는 텍스트를 설정한다. 이 속성은 예제와 같이 하드코딩될 수도 있고 아니면 문자열 리소스의 개념을 사용할 수도 있다.

2. 프로젝트 구성 살펴보기

속성값

리소스 참조시

@[패키지명:]<resource_type>/<resource_name>



스타일 속성의 경우 ?[패키지명:]<resource_type>/<resource_name>

ex. android:textColor = "?android:textColorSecondary"

2. 프로젝트 구성 살펴보기

* 사용자 인터페이스 작성 방법

코드를 사용하는 방법(기존의 자바)

XML을 사용하는 방법(안드로이드 선호 방법)

안드로이드에서는 XML을 이용하여 UI 화면의 구성을 선언적으로 나타내는 방법을 선호

애플리케이션의 외관과 애플리케이션의 로직을 서로 분리

빠르게 UI를 구축

2. 프로젝트 구성 살펴보기

Manifest.xml

애플리케이션에 관한 정보들이 설정됨

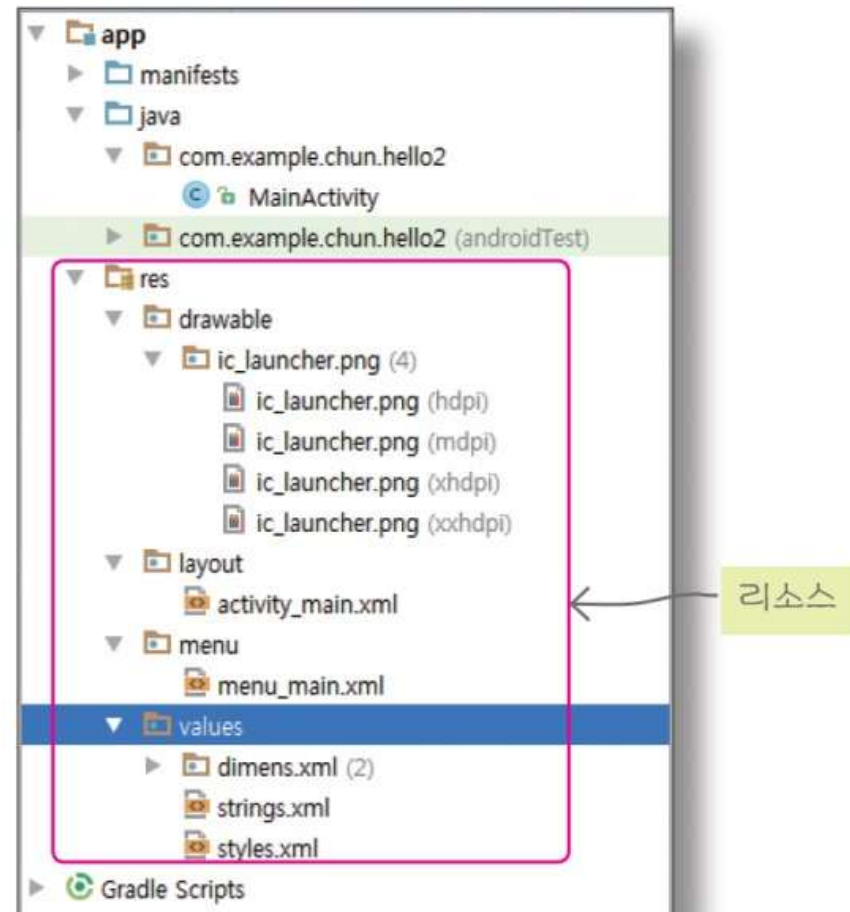
앱 라벨, 아이콘, 액티비티 레벨, 처음 실행될 액티비티명, 권한 등을 명시

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="kr.ac.mjc.kjs.ch02_baseapp" >
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="Ch02_BaseApp"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme" >
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>
22
```

2. 프로젝트 구성 살펴보기

리소스

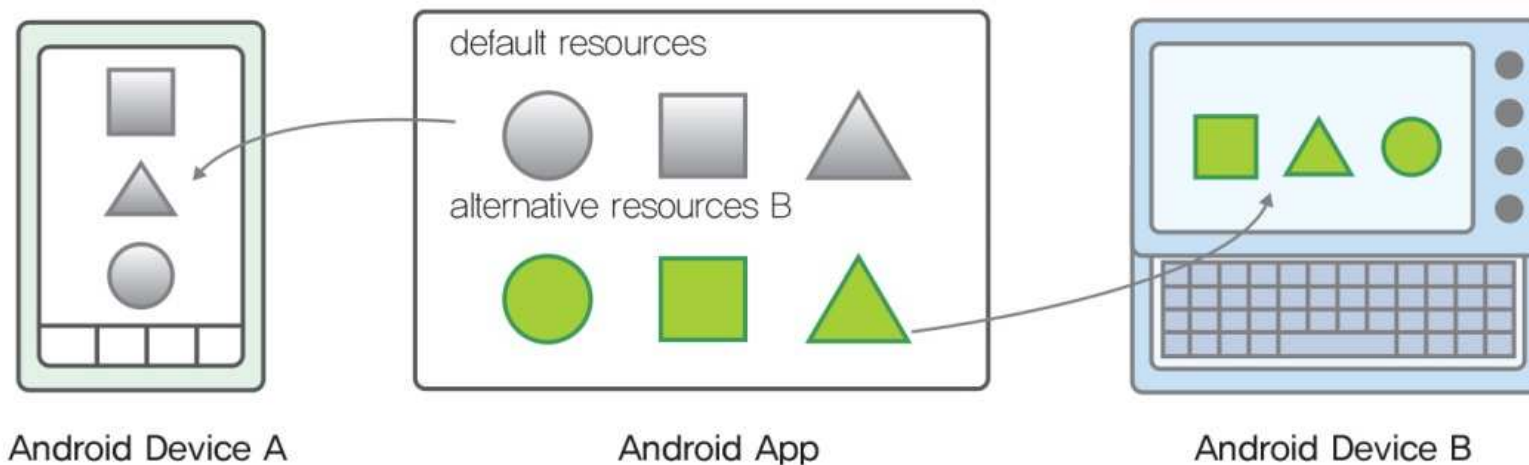
안드로이드에서 레이아웃,
이미지, 문자열들을 리소스로
취급



2. 프로젝트 구성 살펴보기

* 코드와 리소스를 분리하는 이유

안드로이드가 탑재된 장치들이 다양해지면서 언어나 화면 크기에 따라서, 리소스를 다르게 하는 것이 필요

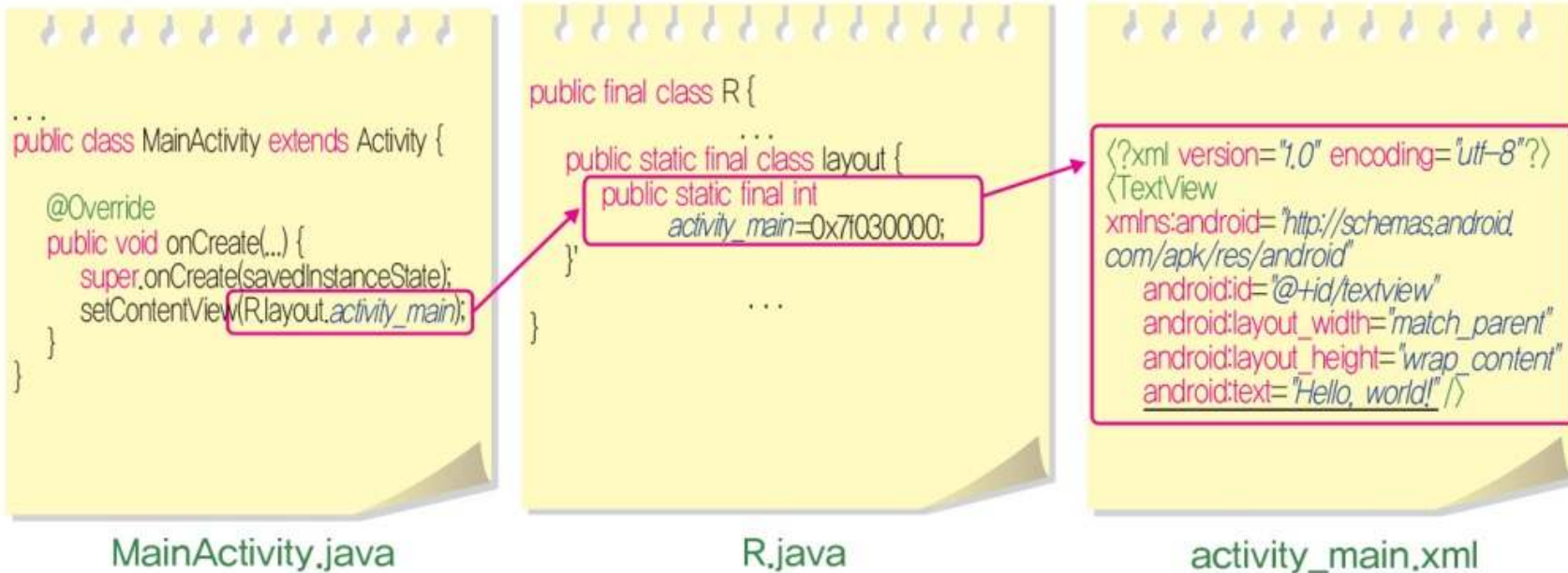


실제 장치에 따라서 서로 다른 리소스가 사용됨

2. 프로젝트 구성 살펴보기

R.Java

빌드 시 xml이 필드로 참조될 수 있도록 바인딩됨



2. 프로젝트 구성 살펴보기

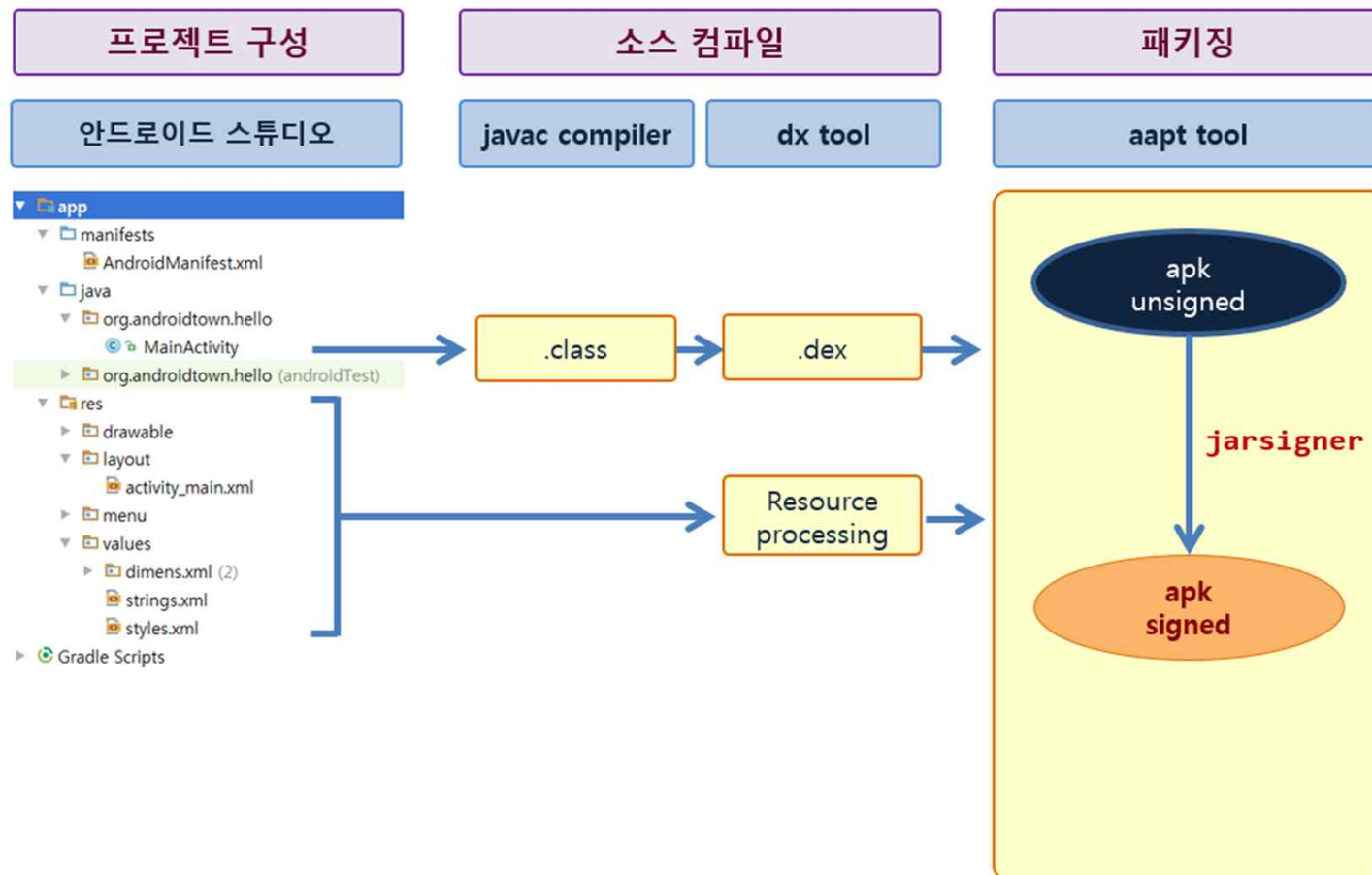
앱이 실행되는 시점에서 setContentView 메소드로 XML에서 설계된 것이 메모리로 로딩되어 객체화(Inflation)



R.layout.레이아웃 파일 이름

2. 프로젝트 구성 살펴보기

프로젝트 빌드 과정



03



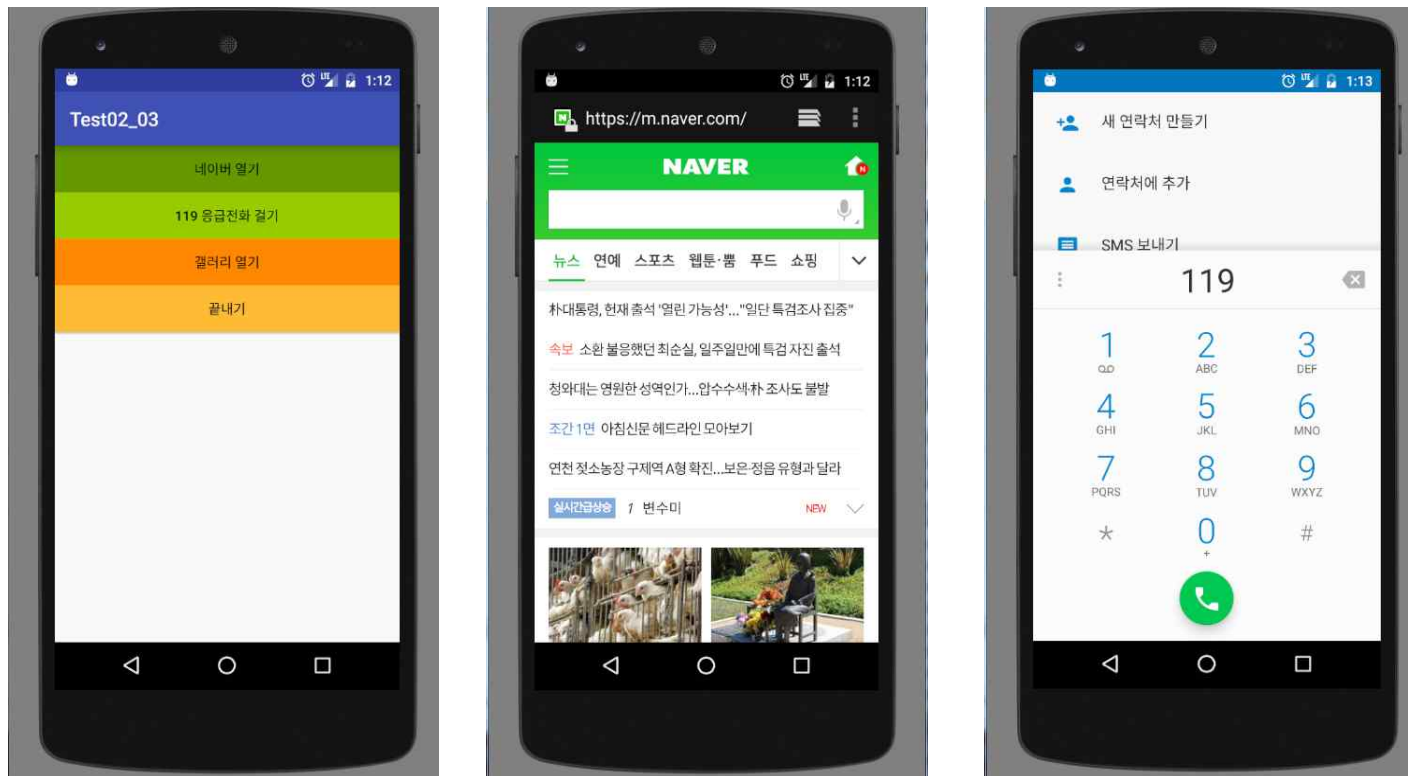
간단한 앱 만들어보기



3. 간단한 앱 만들어보기

실습: Test02_03

버튼으로 웹사이트, 전화걸기 등의 기능 실행



3. 간단한 앱 만들어보기

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:orientation="vertical"
8      tools:context="com.example.a02_activityintent.MainActivity">
9
10     <Button
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:id="@+id/btnNaver"
14         android:text="Naver"
15         android:background="#006400"/>
16
17     <Button
18         android:layout_width="match_parent"
19         android:layout_height="wrap_content"
20         android:id="@+id/btn119"
21         android:text="119 응급전화 걸기"
22         android:background="#5EC75E"/>
23
24     <Button
```

3. 간단한 앱 만들어보기

```
22 <Button
23     android:id="@+id/btnGal"
24     android:layout_width="match_parent"
25     android:layout_height="wrap_content"
26     android:background="#ff8800"
27     android:text="갤러리 열기" />
28
29 <Button
30     android:id="@+id/btnEnd"
31     android:layout_width="match_parent"
32     android:layout_height="wrap_content"
33     android:background="#ffbb33"
34     android:text="끝내기" />
35
36 </LinearLayout>
```

3. 간단한 앱 만들어보기

MainActivity.java

```
10 public class MainActivity extends AppCompatActivity {  
11  
12     Button btnNaver, btn119, btnGai, btnEnd;  
13  
14     @Override  
15     protected void onCreate(Bundle savedInstanceState) {  
16         super.onCreate(savedInstanceState);  
17         setContentView(R.layout.activity_main);  
18  
19         btnNaver = (Button) findViewById(R.id.btnNaver);  
20         btn119 = (Button) findViewById(R.id.btn119);  
21         btnGai = (Button) findViewById(R.id.btnGai);  
22         btnEnd = (Button) findViewById(R.id.btnEnd);  
23     }
```

3. 간단한 앱 만들어보기

```
24  
25  
26  
27  
28  
29  
30  
31  
32  
...  
  
btnNaver.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
  
        Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://m.naver.com"));  
        startActivity(mIntent);  
  
    }  
});
```

3. 간단한 앱 만들어보기

```
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58
```

```
btn119.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("tel:119"));  
        startActivity(mIntent);  
    }  
});  
  
btnGall.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Intent mIntent = new Intent(Intent.ACTION_VIEW, Uri.parse("content://media/internal/images/media"));  
        startActivity(mIntent);  
    }  
});  
  
btnEnd.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        finish();  
    }  
});  
}
```


Q&A

