

안드로이드 프로그래밍 및 실습 - 개인과제1

- 유용석 교수님, 최유진 조교님 -

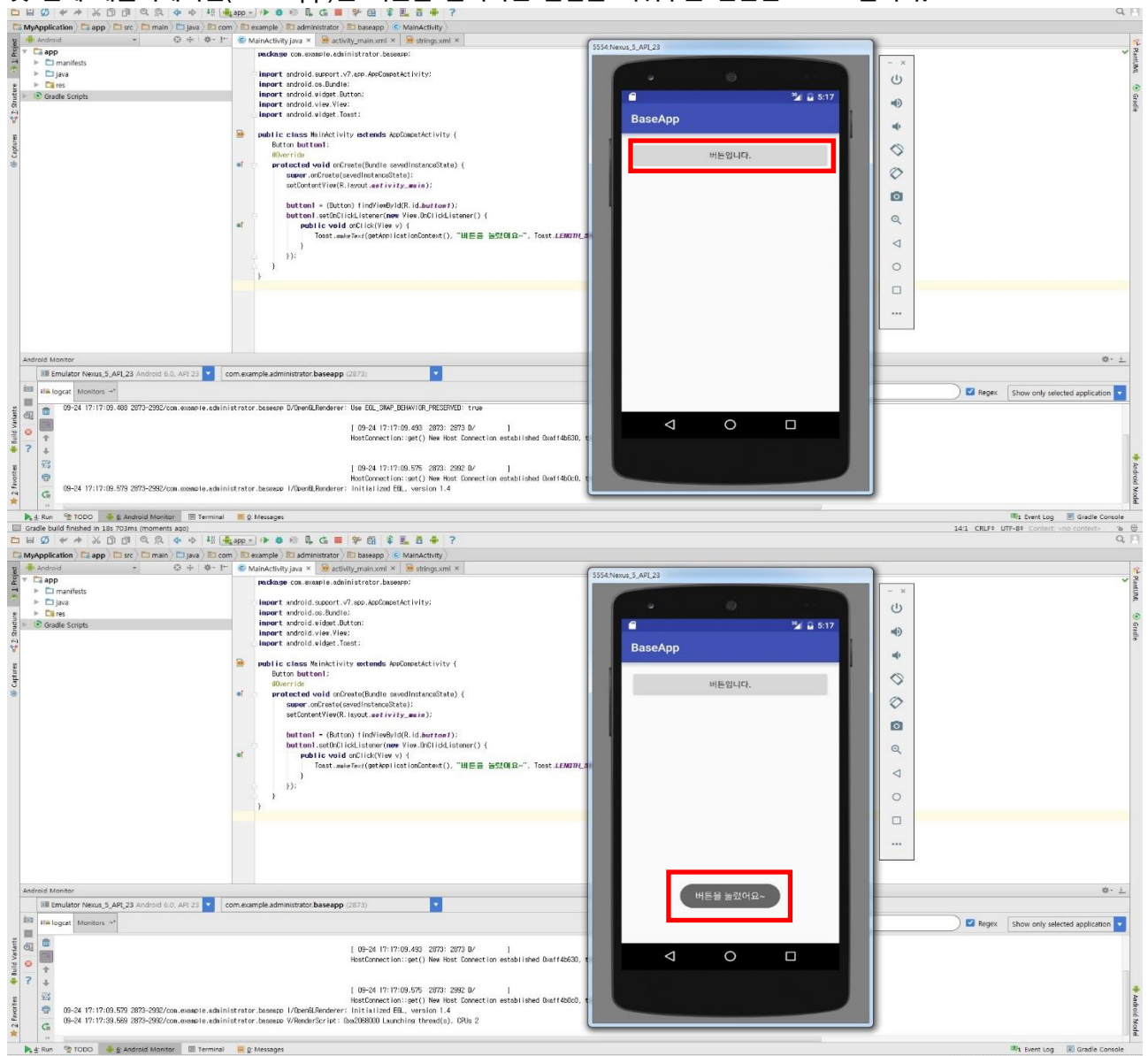
B289070, 이상민

1. 목표

기본적인 모바일 안드로이드 애플리케이션을 직접 제작해봄으로써 모바일 안드로이드 프로그래밍에 익숙해지기 위함이다. 특히, 이번 과제를 수행하여 버튼 widget을 클릭하면 알림을 띄워주는 간단한 예제를 실습해본다. 추가적으로 인터넷 연결, 전화, 갤러리 열기 등 심도 있는 학습을 진행하며, 자동완성 기능을 사용해본다.

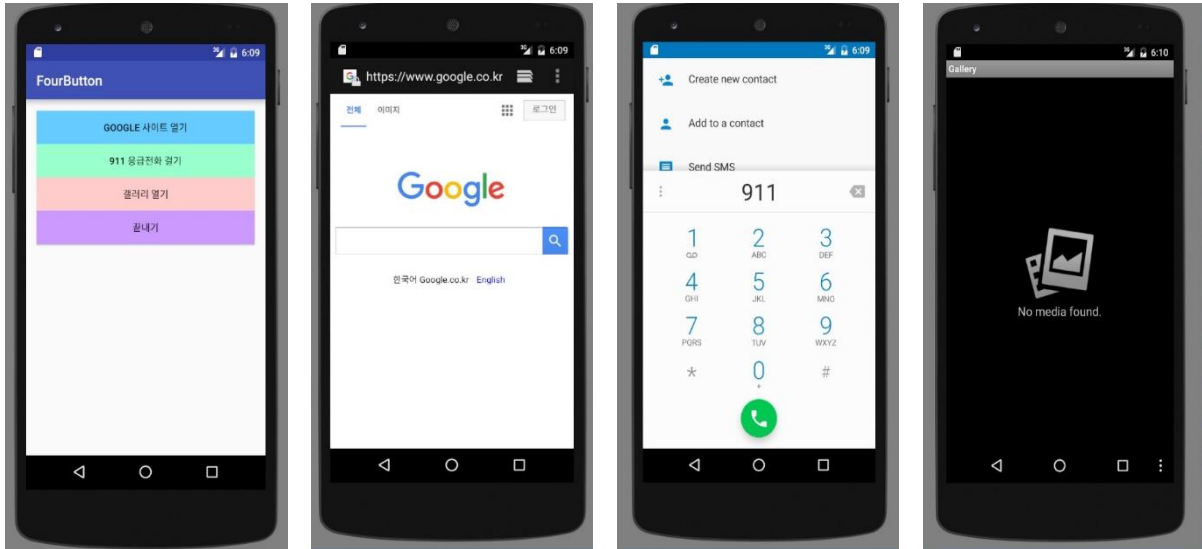
2. 기능 및 사용법

첫 번째 애플리케이션(BaseApp)은 버튼을 클릭하면 알림을 띄워주는 간단한 프로그램이다.



상단 '버튼입니다.' 버튼을 클릭하면, 아래와 같이 '버튼을 눌렀어요~' 알림이 뜨는 것을 알 수 있다.

두 번째 애플리케이션(FourButton)은 4개의 버튼이 있으며, 각각의 버튼이 홈페이지 열기, 911 응급전화 걸기, 사진 갤러리 열기, 종료를 담당한다.



첫 번째 'GOOGLE 사이트 열기' 버튼을 누르면 왼쪽에서 두 번째 그림과 같이 구글 홈페이지가 뜨는 것을 확인할 수 있다. 그 밑에 '911 응급전화 걸기' 버튼을 누르면 왼쪽에서 세 번째 그림과 같이 바로 전화를 걸 수 있도록 창이 뜬다. 세 번째 '갤러리 열기' 버튼을 누르면 맨 오른쪽 그림과 같이 사진 갤러리가 열린다. 지금은 보유한 사진이 없으므로, No media found가 나타난다. 마지막 버튼은 이 애플리케이션을 종료하는 버튼이다.

3. 구현 방법

이번 실습에 관한 소스 코드, 레이아웃 XML의 링크를 첨부한다.

BaseApp 소스코드 : <https://github.com/carpfishz/Android-Study/tree/master/HelloAndroid>

FourButton 소스코드 : <https://github.com/carpfishz/Android-Study/tree/master/FourButton>

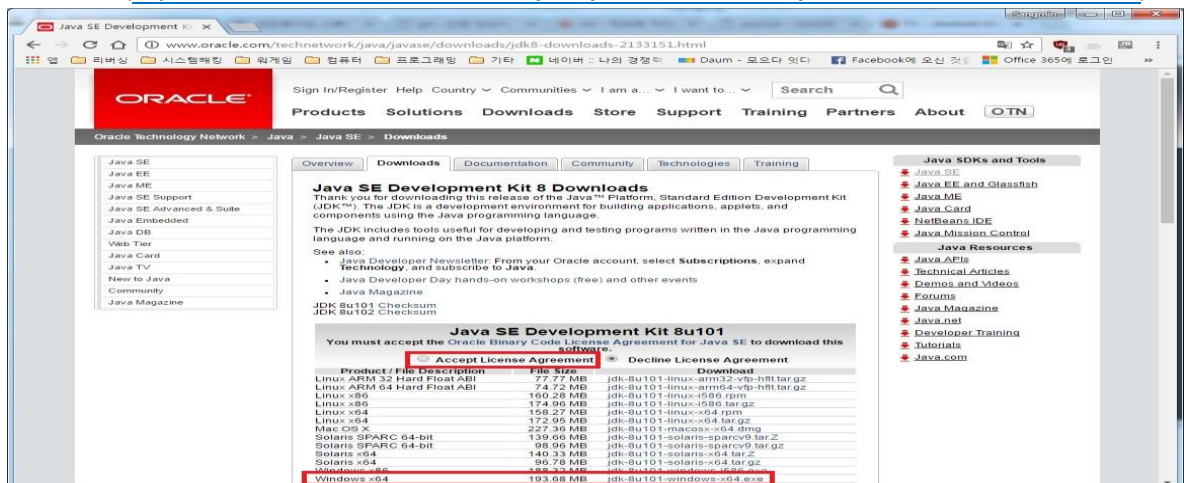
가. 테스트 환경 및 도구

- 1) Windows 7 64bit
- 2) Java 1.8.0_101
- 3) Android Studio 2.1
- 4) AVD : Nexus5, API23

나. 환경 구축

안드로이드 스튜디오를 설치하기 전에 자바가 설치되어 있어야 한다.

- 1) 자바 설치(<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)



라이선스 동의 버튼을 클릭한 뒤, 자신의 버전에 맞는 JDK 파일을 다운로드 한다.

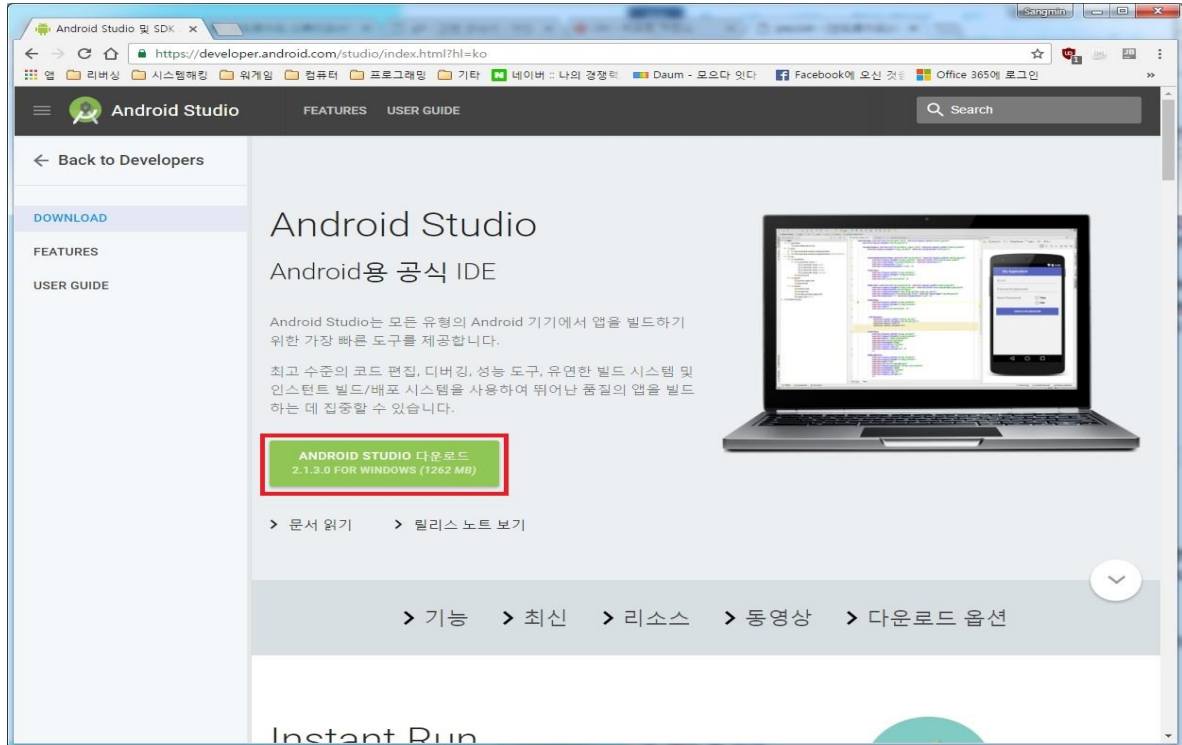
* 언어별 버전 확인 방법

- Windows : 내 컴퓨터 → 마우스 오른쪽 버튼 클릭 → 속성 → Windows 버전
- Linux : 리눅스 버전 확인(터미널 창 `cat /etc/issue`), 비트 확인(터미널 창 `getconf LONG_BIT` or `arch`)
- Mac : 위 사과모양 → 첫 번째 탭 → 이 Mac 정보 클릭

다운로드 받은 파일을 실행시킨 뒤, next 버튼을 누르면 설치가 진행된다.

자세한 설치 및 환경변수 설정 방법이 나와있는 링크를 첨부한다. <http://t2t2tt.tistory.com/9>

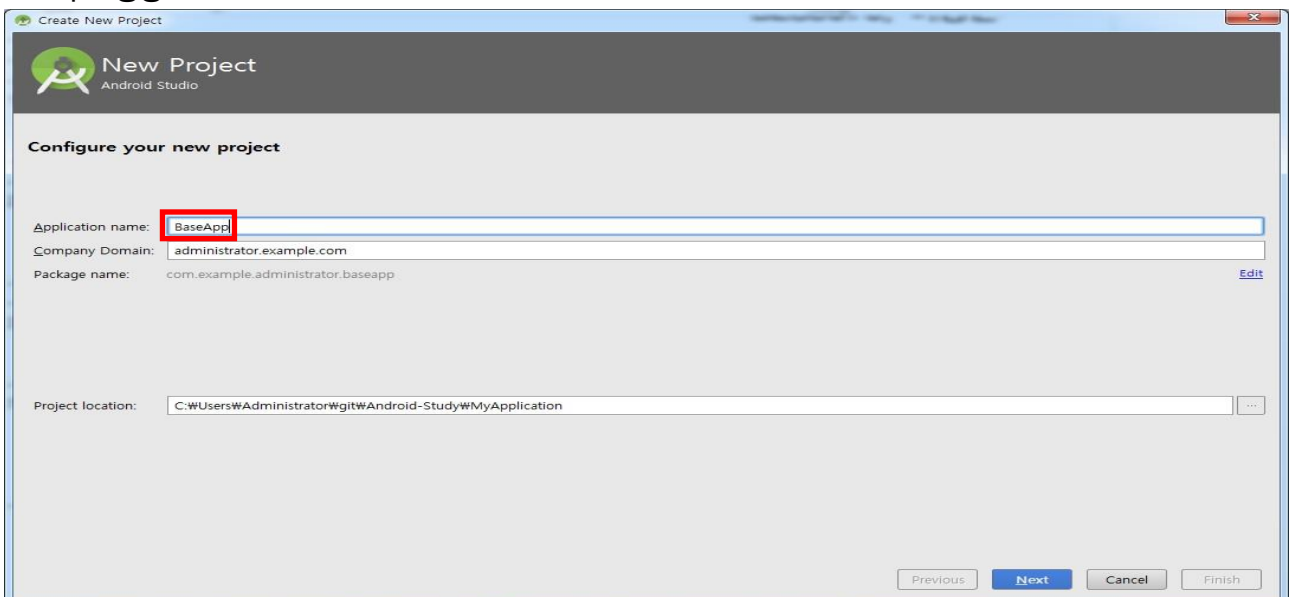
2) 안드로이드 스튜디오 설치(<https://developer.android.com/studio/index.html?hl=ko>)



안드로이드 스튜디오 공식사이트에서 다운로드 받는다. 그 후, 다운로드 받은 파일을 실행시킨 뒤 설치 시 특별히 설정한 부분이 없어 next 버튼을 눌러 설치를 진행한다.

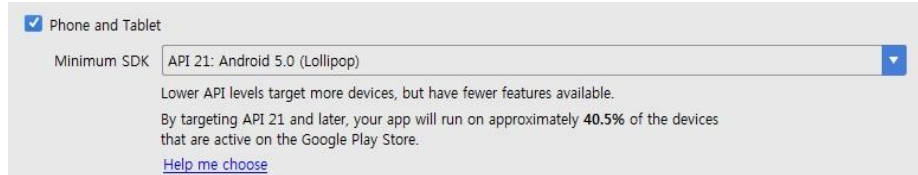
자세한 설치 방법이 나와있는 사이트 링크를 첨부한다. <http://withcoding.com/52>

다. 프로젝트 생성



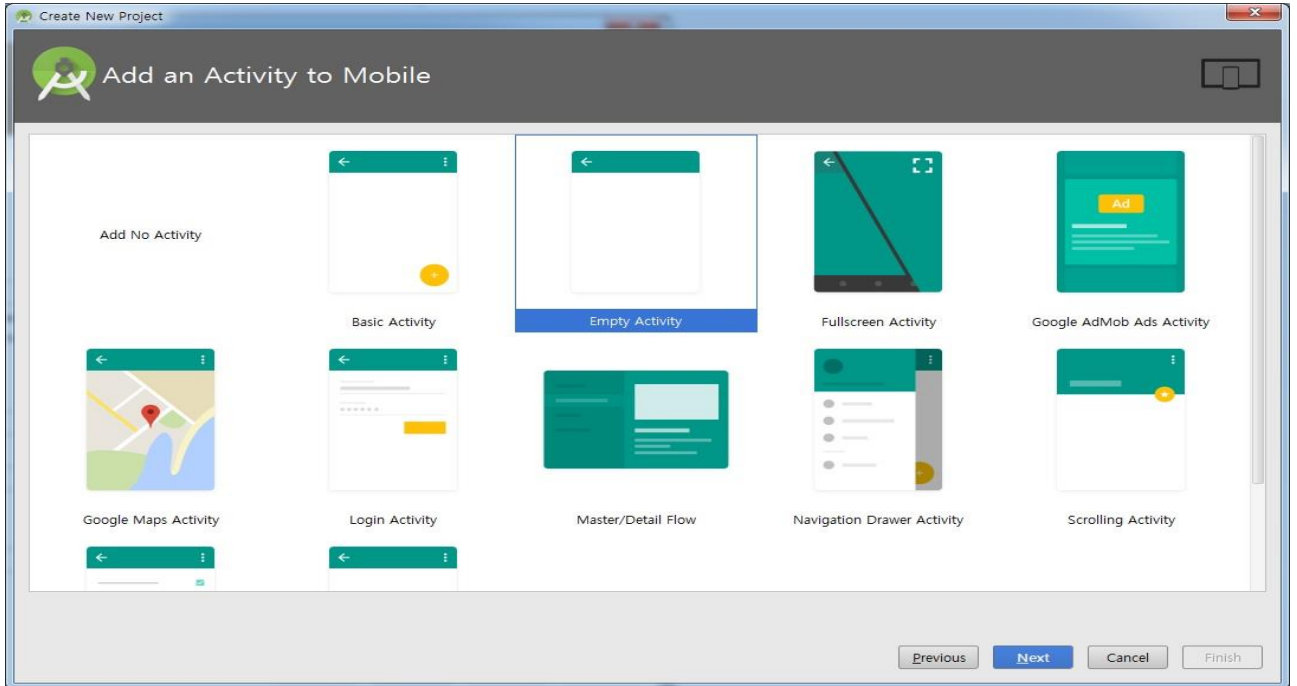
애플리케이션 이름을 BaseApp이라고 지정해준다. 프로젝트 위치는 원하는 위치에 설정하면 된다. 필자는

Git을 사용해 소스코드를 관리할 것이므로 git 폴더 아래에 프로젝트를 생성할 것이다.

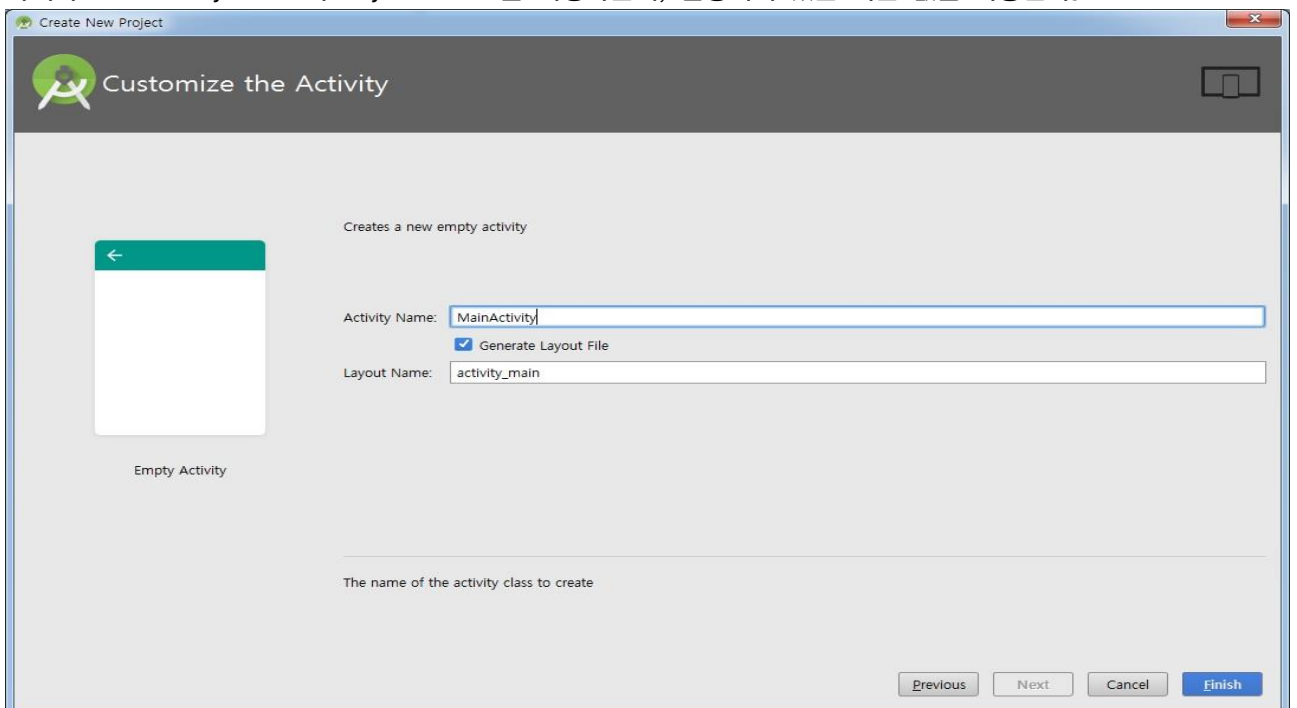


모바일 애플리케이션을 제작할 것이므로 Phone and Tablet을 선택하고 최소 지원 API를 선택한다. 여기서 API 21 : Android 5.0 (Lollipop)을 선택했다.

액티비티 선택 화면에선 우선 기본적으로 선택되어 있는 Empty Activity를 선택한다. 우리가 직접 레이아웃을 구성할 것이다.

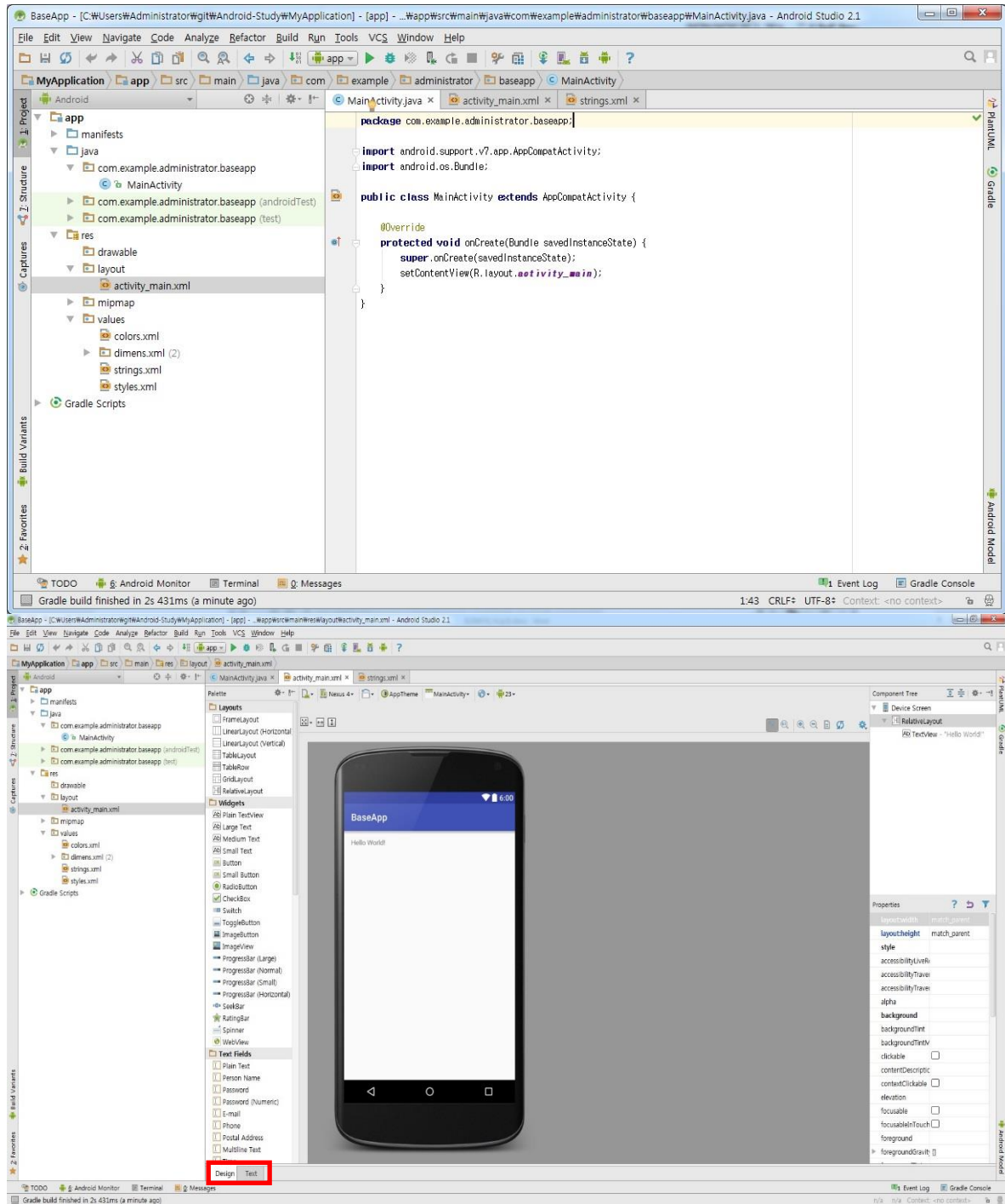


마지막으로 Activity Name과 Layout Name을 지정하는데, 설정되어 있는 기본 값을 사용한다.



라. 애플리케이션 개발

프로젝트를 실행하면 아래의 그림과 같이 MainActivity.java, activity_main.xml을 볼 수 있다.



우리는 activity_main.xml의 하단에 Text 모드에서 버튼을 생성할 것이다.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.administrator.baseapp.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
</RelativeLayout>
```


RelativeLayout을 LinearLayout으로 변경하고, 필요없는 TextView 부분을 삭제한다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.administrator.baseapp.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />

</LinearLayout>
```

Text 모드에서 버튼을 추가하기 위해서 <But를 입력했다. 그런데 빨간색 창과 같이 다양한 내용들이 표시되고 있다. 이중 맨 위 Button을 선택해 Enter를 누른다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.administrator.baseapp.MainActivity">

    <But
```

Button

- android.support.v7.widget.ButtonBarLayout
- ImageButton
- MediaRouteButton
- RadioButton
- ToggleButton
- ZoomButton
- android.support.v7.widget.ActionMenuPresenter.OverflowMenuButton
- android.support.v7.widget.AppCompatButton
- android.support.v7.widget.AppCompatImageButton
- android.support.v7.widget.AppCompatRadioButton

Press Ctrl+Space to view tags from other namespaces

그러면, 아래의 그림과 같이 width, height가 자동으로 입력되고 각각의 값을 선택하도록 도와준다. 우리는 match_parent와 wrap_content를 선택한다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.administrator.baseapp.MainActivity">

    <Button
```

Button

android:layout_width="match_parent"

android:layout_height="wrap_content"

</LinearLayout>

- match_parent
- wrap_content
- @android:
- @dimen/activity_horizontal_margin
- @dimen/activity_vertical_margin
- fill_parent

Documentation for match_parent

The view should be as big as its parent (minus padding). Introduced in API Level 8.

그 외 버튼을 구성하는 나머지 옵션을 적어준다. 하지만, 나머지 옵션을 작성하고 있는데, text 부분에서 오류가 발생한 것을 볼 수 있다. 이는 strings.xml에 해당 내용이 없어서 그런 것이다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.administrator.baseapp.MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button1"
        android:text="@string/strBtn1"

    />

</LinearLayout>
```

strings.xml에 해당 내용을 추가해주면,

```
<resources>
    <string name="app_name">BaseApp</string>
    <string name="strBtn1">버튼입니다.</string>
</resources>
```

아래 그림과 같이 오류가 수정된 것을 볼 수 있다.

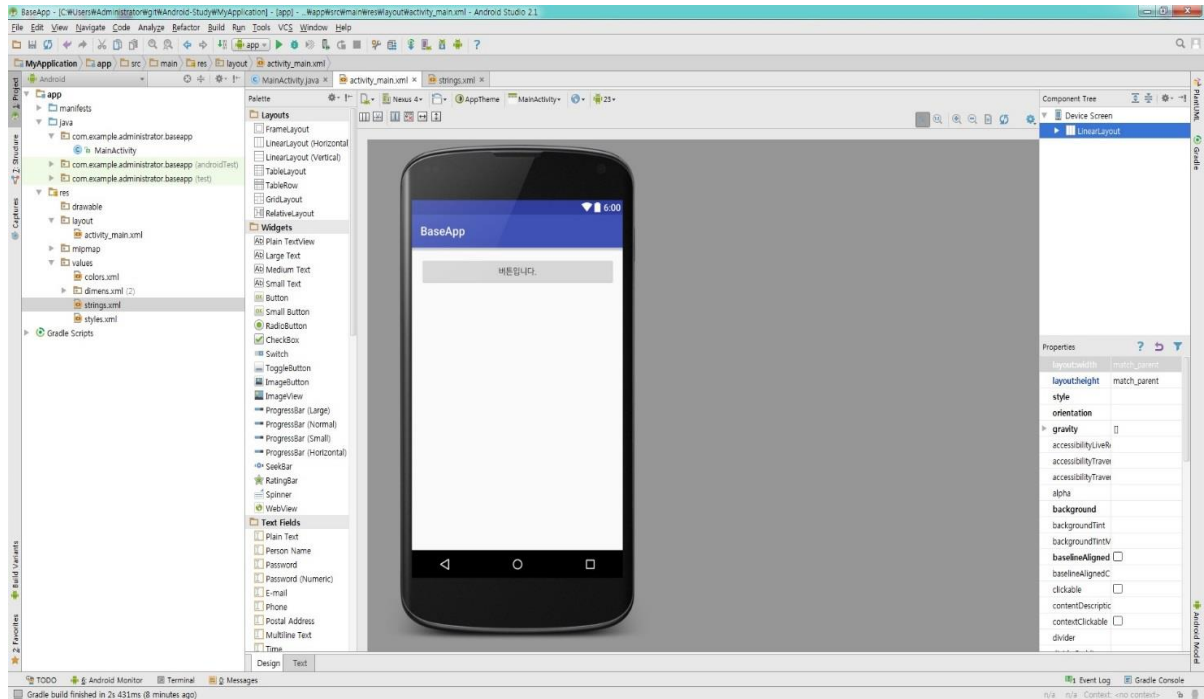
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.administrator.baseapp.MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button1"
        android:text="@string/strBtn1"

    />

</LinearLayout>
```

이를 Design 모드에서 확인해보면, 버튼이 추가된 것을 확인할 수 있다. 지금까지는 버튼을 추가해준 것이다. 그러면, 버튼을 클릭했을 때, 알림 이벤트가 발생하도록 코드를 추가해줘야 한다.



MainActivity.java에 Button button1이라고 작성하면 빨간색 창과 같이 알림창이 뜨는데, 여기서 Alt + Enter 키를 누르면, 관련 라이브러리가 자동으로 import 된다.

```
package com.example.administrator.baseapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
? android.widget.Button? Alt+Enter
public class MainActivity extends AppCompatActivity {
    Button button1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```
package com.example.administrator.baseapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button button1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

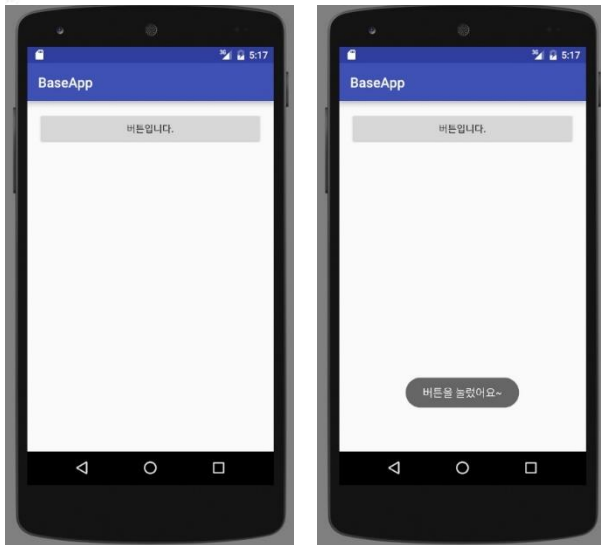

버튼 동작과 관련된 코드를 마저 작성한 뒤, 실행하면 아래와 같이 동작하는 것을 확인할 수 있다.

```
package com.example.administrator.baseapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Button;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button button1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button1 = (Button) findViewById(R.id.button1);
        button1.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                Toast.makeText(getApplicationContext(), "버튼을 눌렀어요~", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```



소스 코드를 간략하게 분석해보면, Button button1은 버튼 객체를 생성해주는 구문이다.

button1 = (Button) findViewById(R.id.button1)는 button1의 고유 값(id)를 가져와 button1에 대입한다. button1.setOnClickListener(new View.OnClickListener() { button1을 클릭을 기다리는 부분을 구현한 부분이다. View.OnClickListener() 부분은 매개변수로 전달하며 이는 view가 클릭될 때 onClick을 수행한다. public void onClick(View v) {

Toast.makeText(getApplicationContext(), "버튼을 눌렀어요~", Toast.LENGTH_SHORT).show()

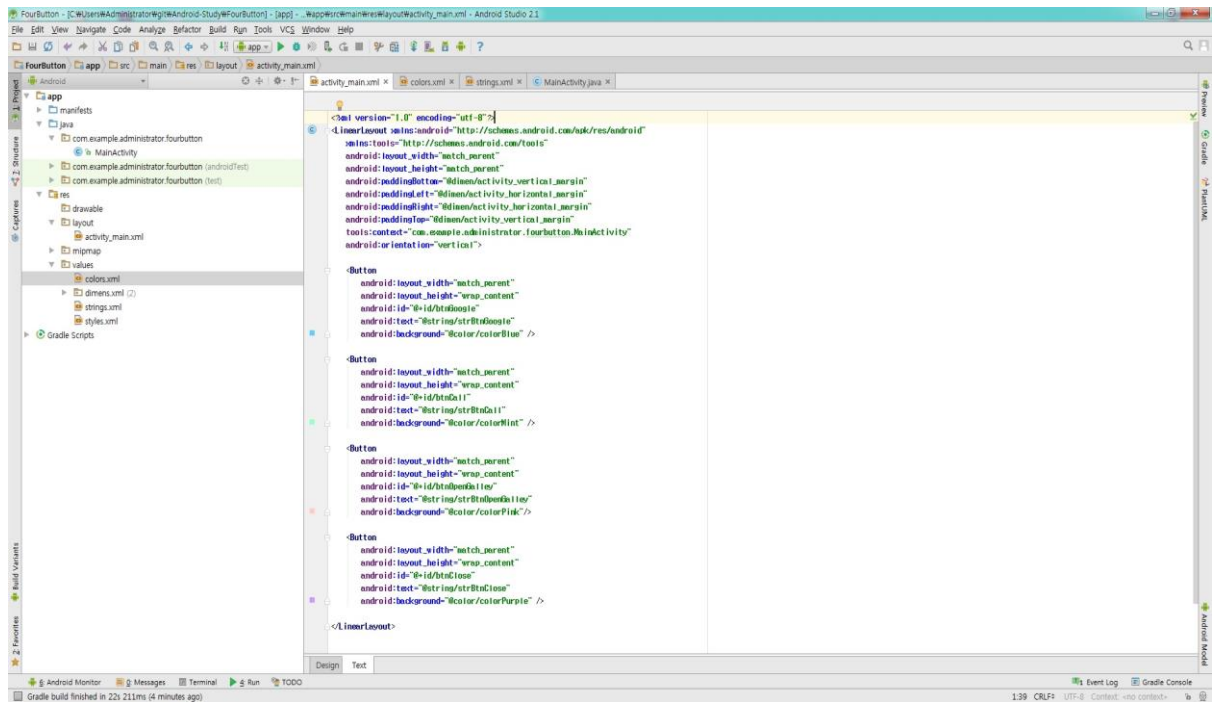
이 구문이 실질적인 기능을 수행하는 부분이며, Toast 클래스는 애플리케이션에서 짧은 메시지를 띄워주는 것을 말한다. 그 중 makeText 메소드는 3개의 매개변수를 갖는다.

static	makeText(Context context, CharSequence text, int duration)
Toast	Make a standard toast that just contains a text view.

첫 번째 매개변수는 Context 객체를 갖는데, getApplicationContext()를 사용하고, 두 번째 매개변수로 화면에 출력한 내용을 넣는다. 마지막으로 알림을 몇 초 동안 표시할 것인지 정하는 매개변수를 넣어준다.

<code>abstract Context</code>	<code>getApplicationContext()</code> Return the context of the single, global Application object of the current process.
-------------------------------	---

이를 응용하여 Four Button 애플리케이션을 제작해보자. 우선, 4개의 버튼을 추가해준다. 순서대로 홈페이지 열기, 전화 걸기, 갤러리 열기, 종료 기능을 수행하는 버튼이다. 버튼 안에 들어가는 텍스트는 strings.xml에 정의하고, 버튼 색깔과 관련 내용은 color.xml 안에 정의한다. 이는 웹 개발 중 HTML, CSS로 구분하여 관리하는 것과 비슷한 개념이다.



<strings.xml>

```
<resources>
    <string name="app_name">FourButton</string>
    <string name="strBtnGoogle">Google 사이트 열기</string>
    <string name="strBtnCall">911 응급전화 걸기</string>
    <string name="strBtnOpenGallery">갤러리 열기</string>
    <string name="strBtnClose">끝내기</string>
</resources>
```

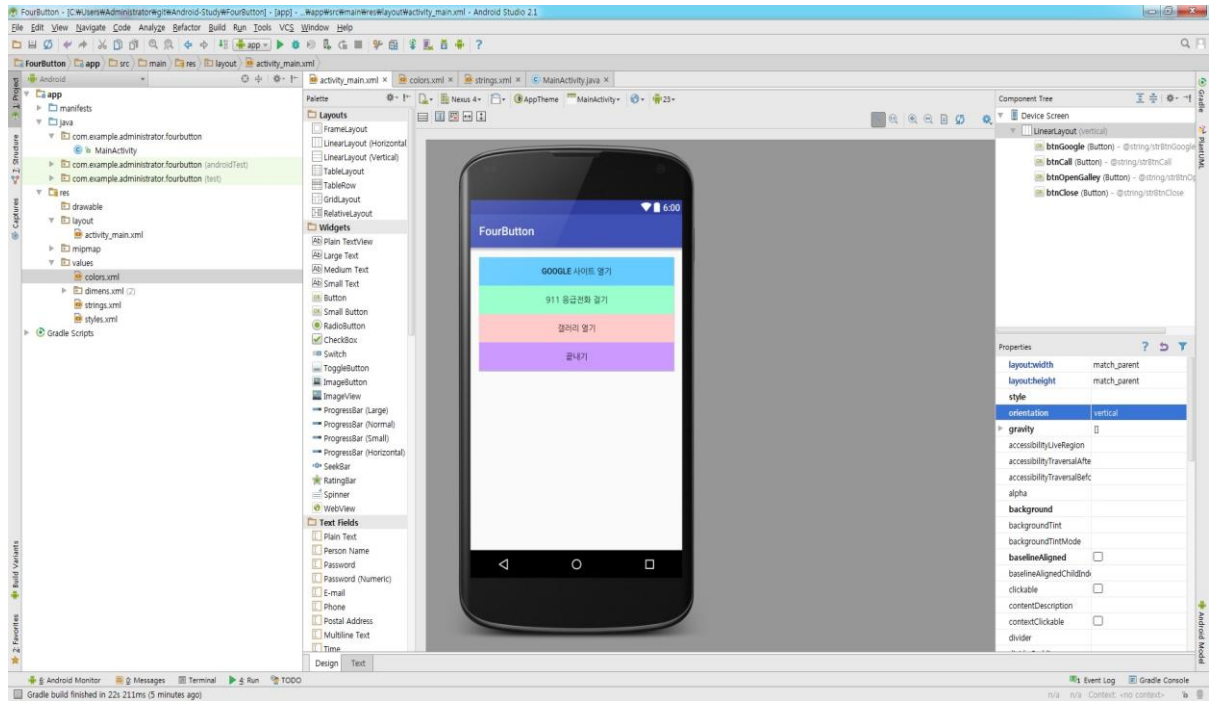
<color.xml>

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>

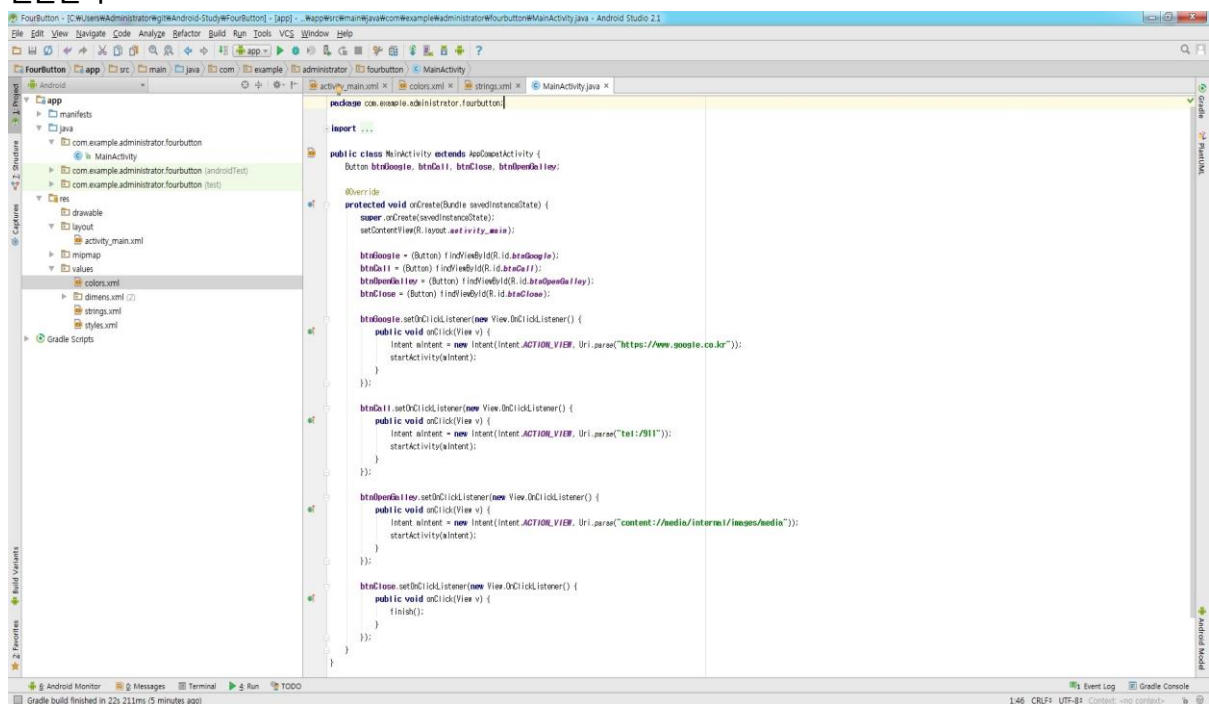
    <color name="colorBlue">#66ccff</color>
    <color name="colorPurple">#cc99ff</color>
    <color name="colorMint">#99ffcc</color>
    <color name="colorPink">#ffcccc</color>
</resources>
```

Android Studio의 장점으로는 바로 왼쪽에 색을 표시해준다는 점이다. 클릭하여 RGB 값으로 수정이 가능하다.

이렇게 버튼을 구성하면 아래와 같이 레이아웃이 구성된다. 만약 이렇게 레이아웃이 안 보일 경우 오른쪽 Properties → orientation → vertical 을 선택하면 된다.



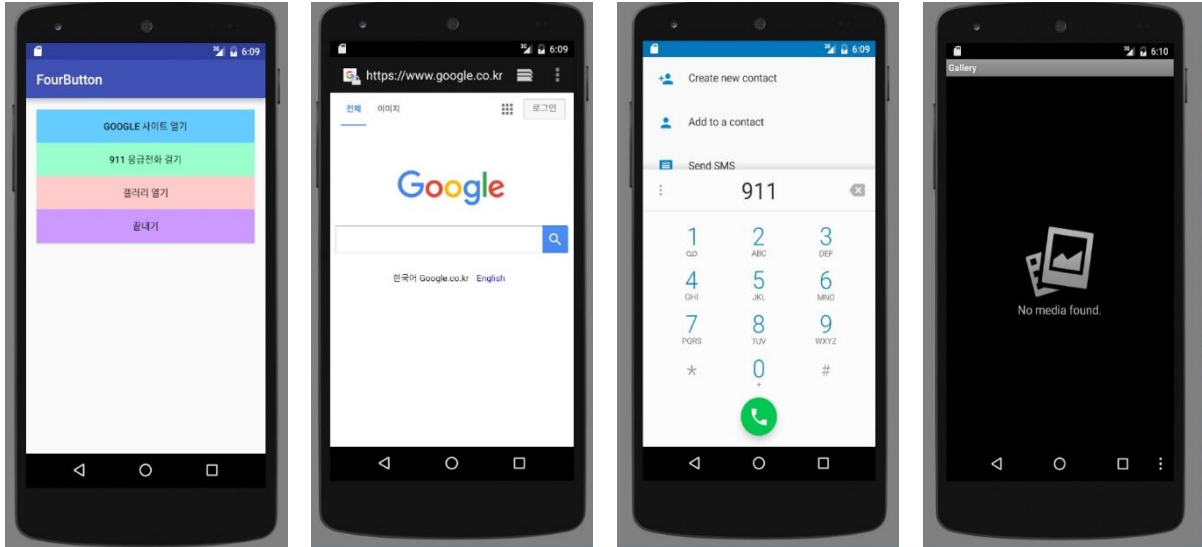
레이아웃 구성이 다 끝나면, 기능을 작성해야 한다. 앞에서 학습한 예제와 동일한 방법으로 구현을 한다. 다른 점은, Intent라는 개념이 추가되었는데, 간단하게 설명하면 다른 화면으로 전환할 때 Intent를 통해 전환한다



4. 결과 및 분석

데모 영상 : <https://youtu.be/bU2f5SyqXTI>

맨 왼쪽 첫 번째 그림은 우리가 제작한 애플리케이션을 실행한 모습이다. activity_main.xml에서 본 모습과 동일하다. 첫 번째 버튼 'Google 사이트 열기'을 눌러보면 왼쪽에서 두 번째 그림과 같이 구글 홈페이지가 켜진 것을 확인할 수 있다. 두 번째 버튼을 누르면 왼쪽 세 번째 그림과 같이 전화를 걸 수 있는 화면이 표시된다. 세 번째 '갤러리 열기' 버튼을 클릭하면 맨 오른쪽 그림과 같이 갤러리가 실행되며, 지금은 저장된 파일이 없기 때문에 No media found 라고 표시된다. 마지막으로 '끝내기' 버튼을 클릭하면 애플리케이션이 종료한다.



이번 과제는 버튼을 클릭하여 특정 이벤트가 발생하는 애플리케이션을 제작하였다. 그 과정에서 Toast, Intent 클래스를 사용하였다. 기존에는 레이아웃 구성만을 실습하였는데, 이번에는 구성한 레이아웃에 특정 이벤트를 추가하여 진정한 모바일 안드로이드 애플리케이션을 제작하였다.

우리는 앞으로 버튼을 조금 더 직관적이고 예쁘게 꾸며야 하고, 더 나아가 이벤트를 발생하는 방법에는 4가지 정도가 있는데, 이에 대하여 학습하면 도움이 될 것 같다.

5. 참고자료

예제 : Android Studio를 활용한 안드로이드 프로그래밍(우재남, 박길식 지음) p78, 98

안드로이드 API 공식 문서에서 View.OnClickListener 부분

<https://developer.android.com/reference/android/view/View.OnClickListener.html>