

안드로이드 앱 구조

목차

안드로이드 앱

- 개요
- 구조
- setContentView()
- 화면에 텍스트 표시 (with layout XML)
 - ◆ TextView
- 앱 리소스 연결
- findViewById()
- 화면에 텍스트 표시 (without layout XML)
- LinearLayout

안드로이드 앱 개요

Reference: <https://developer.android.com/guide/>



안드로이드 앱

- 앱 컴포넌트(들), 매니페스트 파일, 리소스
- 확장자 .apk인 아카이브 파일로 컴파일됨
- 서로 다른 앱 간 분리
 - ◆ 각 앱은 별도 리눅스 프로세스에서 실행되며 자체 가상머신을 가짐



앱 컴포넌트 유형

- Activity, Broadcast receiver, Content provider, Service



인텐트

- 하나의 앱에서 동일 앱 내 컴포넌트 혹은 다른 앱 내 컴포넌트를 인텐트를 통해 시작시킬 수 있음
- 명시적 인텐트
 - ◆ 시작시킬 컴포넌트의 클래스 이름을 명확히 설정하여 해당 컴포넌트를 시작
- 암시적 인텐트
 - ◆ 시작시킬 작업 유형을 바탕으로 안드로이드 시스템이 장치 내 해당 작업 유형을 만족하는 컴포넌트를 시작



매니페스트 파일

- 앱 구성요소 (activity, broadcast receiver, content provider, service) 선언
- 앱 요구사항 선언
 - ◆ 사용자 권한(예: 인터넷 접근, 연락처 접근), 최소 요구 안드로이드 버전, HW 요구 기능(예: 카메라 필요 등)

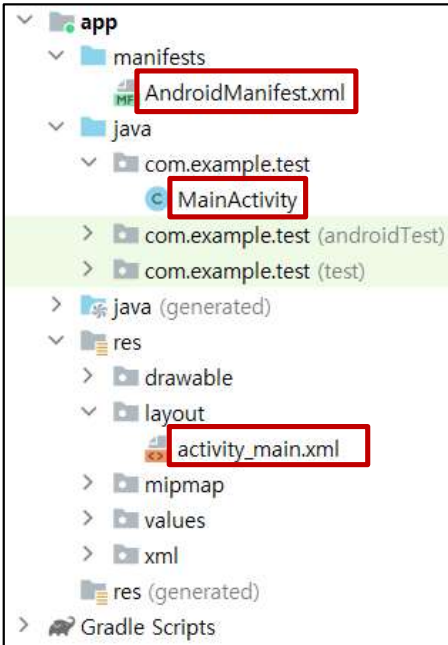


리소스

- 이미지, 오디오, 메뉴, 색상 등 리소스를 소스코드와 분리

안드로이드 앱 구조

프로젝트



AndroidManifest.xml (매니페스트 파일)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.test">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.Test"
        tools:targetApi="31">
        <activity android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

activity_main.xml (레이아웃 리소스 파일, optional)

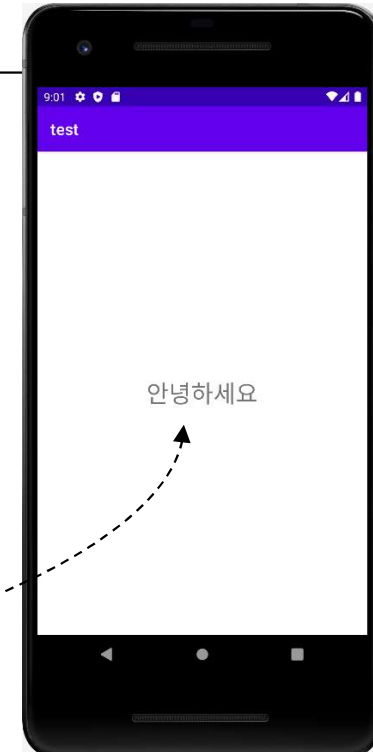
```
<?xml version="1.0" encoding="utf-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="안녕하세요"
    android:textSize="30sp" />
```

MainActivity.java

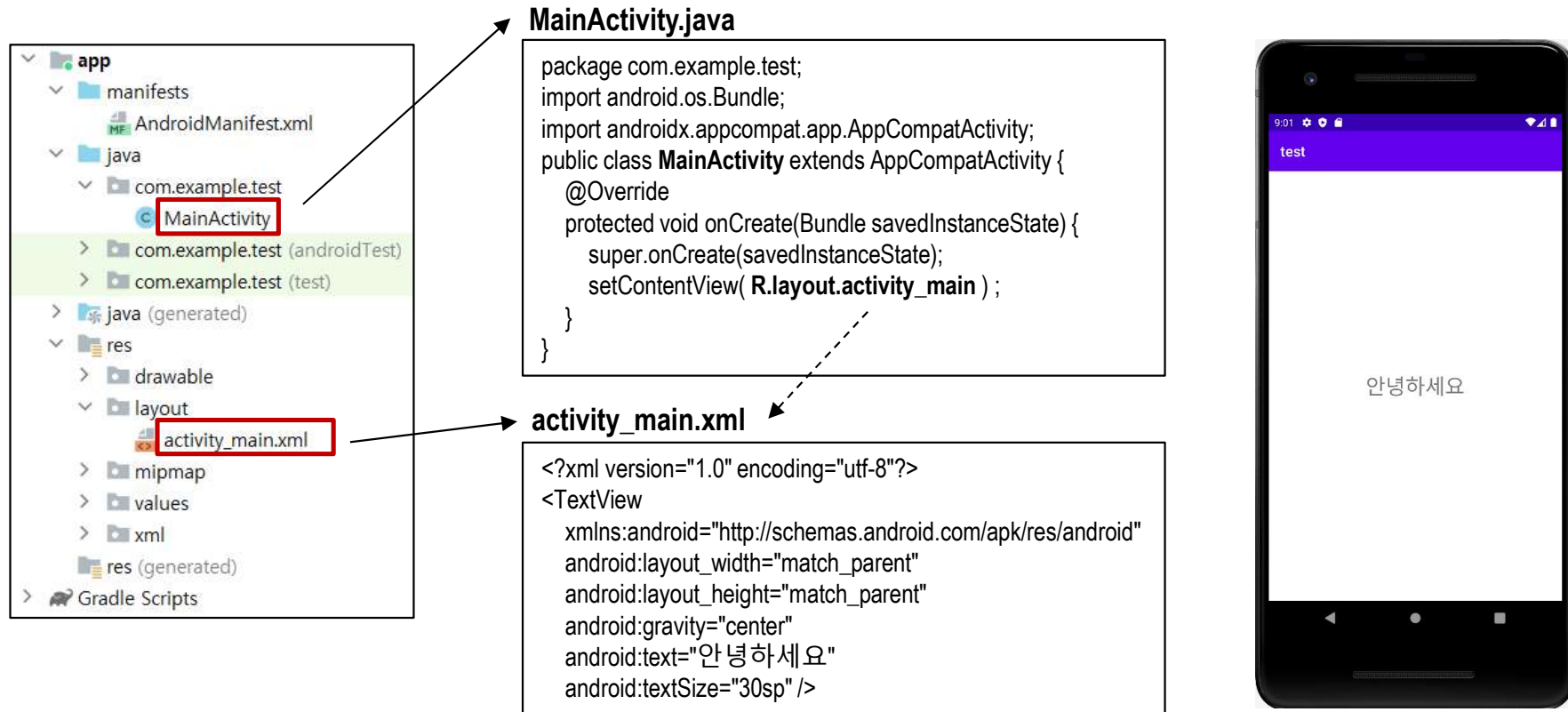
```
package com.example.test;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

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

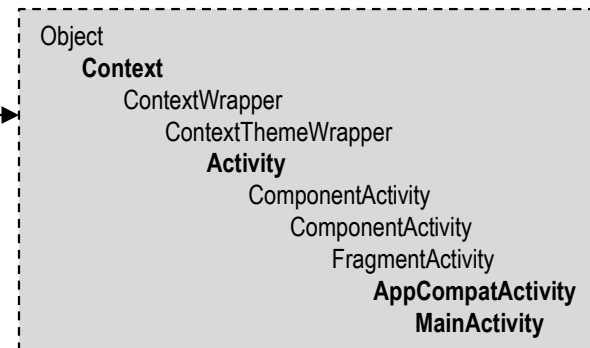


화면에 텍스트 표시(with layout XML)



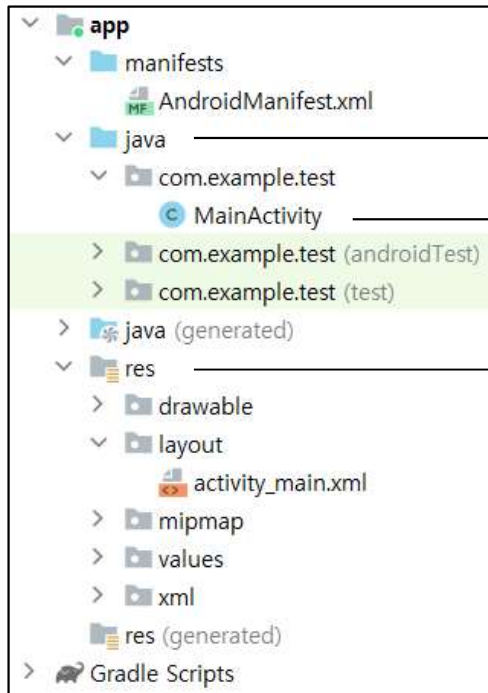
화면에 텍스트 표시 (레이아웃 XML 파일 사용)

- 동작개요 → 앱을 실행하면 텍스트 뷰에 안녕하세요를 표시
- MainActivity의 상위 클래스 계층 확인(안드로이드 스튜디오에서 CTRL+H)
- onCreate 메소드
- setContentView 메소드 → public void setContentView(int layoutResID) → 액티비티 내용을 주어진 레이아웃 리소스로 설정
- R.layout.activity_main에 대응하는 파일
- R.layout.activity_main의 자료형
- 텍스트 뷰 → public class **TextView** extends **View** implements ... { ... }



앱 리소스 연결: R.java

C:\Users\사용자아이디\AndroidStudioProjects\test\app\src\main\java\com\example\test\MainActivity.java



C:\Users\사용자아이디\AndroidStudioProjects\test\app\src\main\res

앱 리소스 연결

- setContentView 메소드 → public void setContentView(int layoutResID)
- R.layout.activity_main에 대응하는 파일
- R.layout.activity_main의 자료형

MainActivity.java

```
package com.example.test;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView( R.layout.activity_main );
    }
}
```

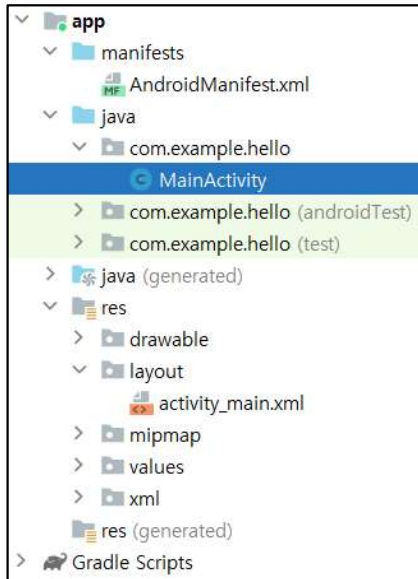
R.java

```
public final class R {
    public static final class layout {
        public static final int activity_main = 1300000;
        // 다른 내용 생략
    }
    // 다른 내용 생략
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:text="안녕하세요"
    android:textSize="30sp" />
```

findViewById()



MainActivity.java

```
package com.example.test;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView textview = findViewById(R.id.textview);
        textview.setText("대한민국");
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
    android:id="@+id/textview"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="안녕하세요"
    android:textSize="30sp" />
```



화면에 텍스트 표시

- 앱 리소스 연결 → public <T extends View> T **findViewById**(int id) → id로 식별되는 뷰를 반환
- TextView → public final void **setText**(CharSequence text) → 파라미터로 전달된 텍스트를 텍스트 뷰에 설정

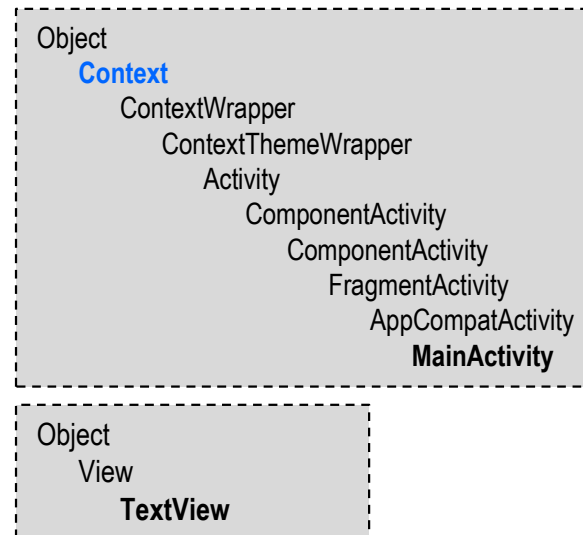
화면에 텍스트 표시(without layout XML)

MainActivity.java

```
// package 문, import 문 생략
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        TextView textView=new TextView(this);
        textView.setText("안녕하세요");
        textView.setTextSize(30);
        textView.setGravity(Gravity.CENTER);

        setContentView(textView);
    }
}
```



화면에 텍스트 표시 (레이아웃 XML 파일 미사용)

- 동작개요 → 앱을 실행하면 텍스트 뷰에 안녕하세요를 표시
- TextView 생성자 → public TextView(**Context** context)
- TextView → public final void setText(CharSequence text) → 파라미터로 전달된 텍스트를 텍스트 뷰에 설정
- String → public final class **String** implements java.io.Serializable, Comparable<String>, **CharSequence** { ... }
- TextView → public void setTextSize(float size) → size는 scaled pixel 단위로 해석됨
- TextView → public void setGravity(int gravity)
- setContentView 메소드 → public void setContentView(View view) → 액티비티 내용을 주어진 뷰로 설정

화면에 텍스트 표시(with layout XML): LinearLayout, TextView

MainActivity.java

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

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:gravity="center">  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:gravity="center"  
        android:text="안녕하세요"  
        android:textSize="30sp"/>  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:gravity="center"  
        android:text="반갑습니다"  
        android:textSize="30sp"/>  
</LinearLayout>
```

화면에 텍스트 표시 (레이아웃 XML 파일 사용)

- 동작개요 → LinearLayout 내에 수직 배치된 두 개 텍스트 뷰에 각각 안녕하세요와 반갑습니다를 표시
- public class TextView extends View implements ... { ... }
- public class LinearLayout extends ViewGroup { ... }
- public abstract class ViewGroup extends View implements ... { ... }
- android:orientation="vertical" → 수직 배치

Object
View
ViewGroup
LinearLayout

Object
View
TextView



화면에 텍스트 표시(without layout XML): LinearLayout, TextView

MainActivity.java

```
// package 문, import 문 생략
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        LinearLayout linearLayout=new LinearLayout(this);

        linearLayout.setOrientation(LinearLayout.VERTICAL);
        linearLayout.setGravity(Gravity.CENTER);

        TextView textView1=new TextView(this);
        textView1.setText("안녕하세요");
        textView1.setTextSize(30);
        textView1.setGravity(Gravity.CENTER);

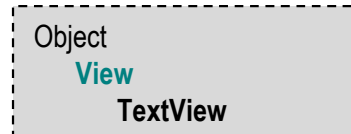
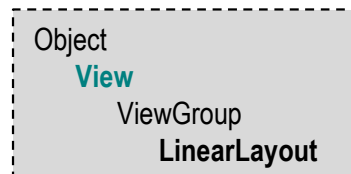
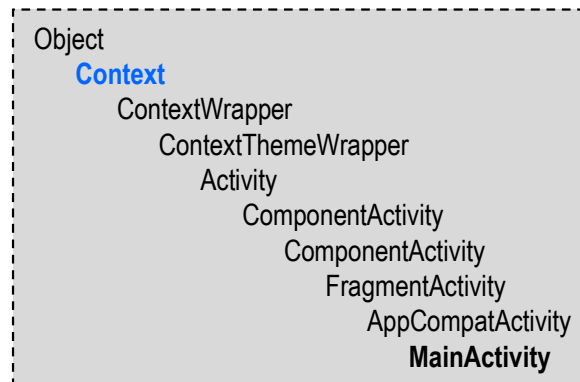
        TextView textView2=new TextView(this);
        textView2.setText("반갑습니다");
        textView2.setTextSize(30);
        textView2.setGravity(Gravity.CENTER);

        linearLayout.addView(textView1);
        linearLayout.addView(textView2);

        setContentView(linearLayout);
    }
}
```

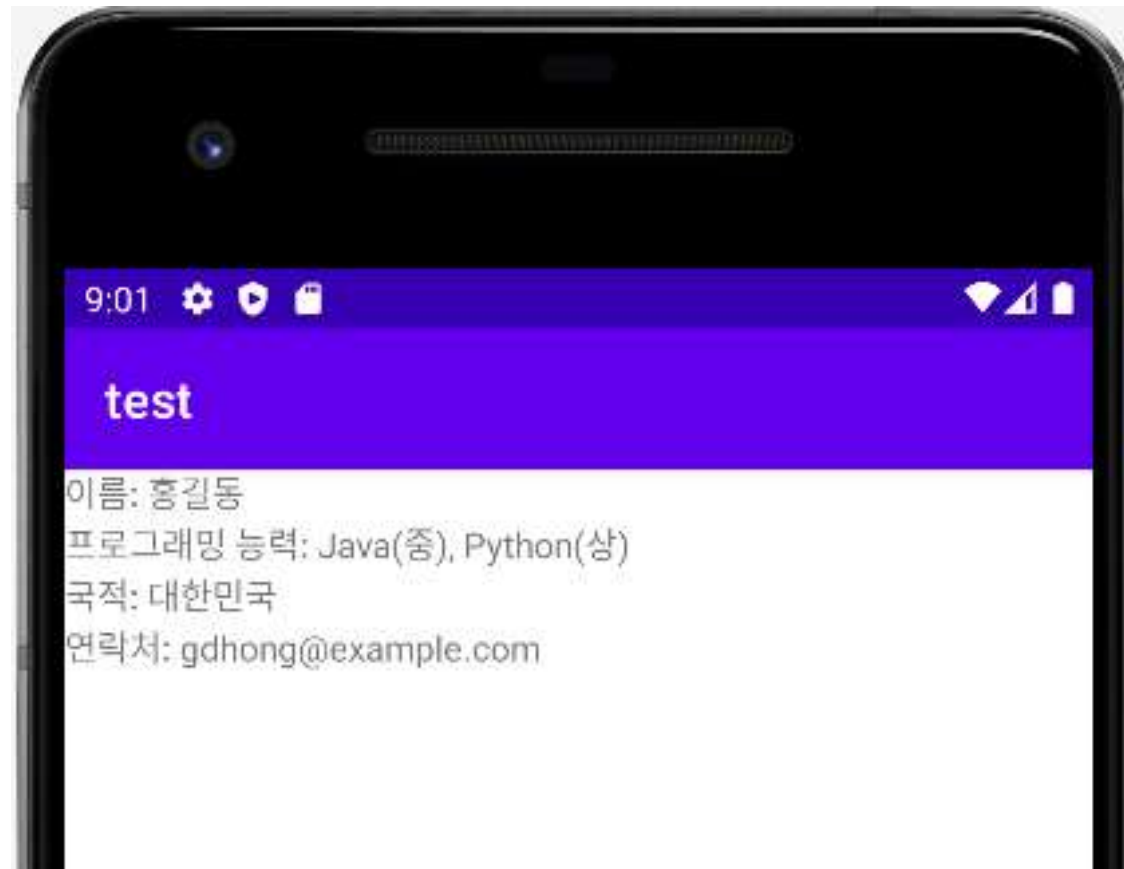
화면에 텍스트 표시 (레이아웃 XML 파일 미사용)

- 동작개요 → LinearLayout 내에 수직 배치된 두 개 텍스트 뷰에 각각 안녕하세요와 반갑습니다를 표시
- LinearLayout 생성자 → public LinearLayout(**Context** context)
- TextView 생성자 → public TextView(**Context** context)
- LinearLayout → public void addView(**View** child)
- setContentView 메소드 → public void setContentView(**View** view)




실습

- ① 아래 실행 화면과 같이 동작하는 앱을 레이아웃 XML을 사용하여 작성하시오
- ② 아래 실행 화면과 같이 동작하는 앱을 레이아웃 XML을 사용하지 않고 작성하시오



References

 <https://developer.android.com/>

 천인국. (2020). 그림으로 쉽게 설명하는 안드로이드 프로그래밍
. 생능출판사.