



## 인텐트로 화면 전환하기

# 학습 목표

- 안드로이드의 인텐트를 이해한다.
- 명시적 인텐트 처리에 대해 익힌다.
- 암시적 인텐트 처리에 대해 익힌다.
- 인텐트 필터를 이해한다.

# 목차

01 인텐트 살펴보기

02 명시적 인텐트로 화면 전환하기

03 암시적 인텐트로 화면 전환하기

04 인텐트 필터 다루기

[도서 쇼핑몰] 도서 상세 정보 표시하기

01

인텐트 살펴보기

# 1. 인텐트 살펴보기

## ■ 인텐트

- 안드로이드 구성 요소인 액티비티, 서비스, 브로드캐스트 리시버 등이 작업을 요청하거나 통신할 때 사용하는 객체

표 6-1 인텐트의 용도

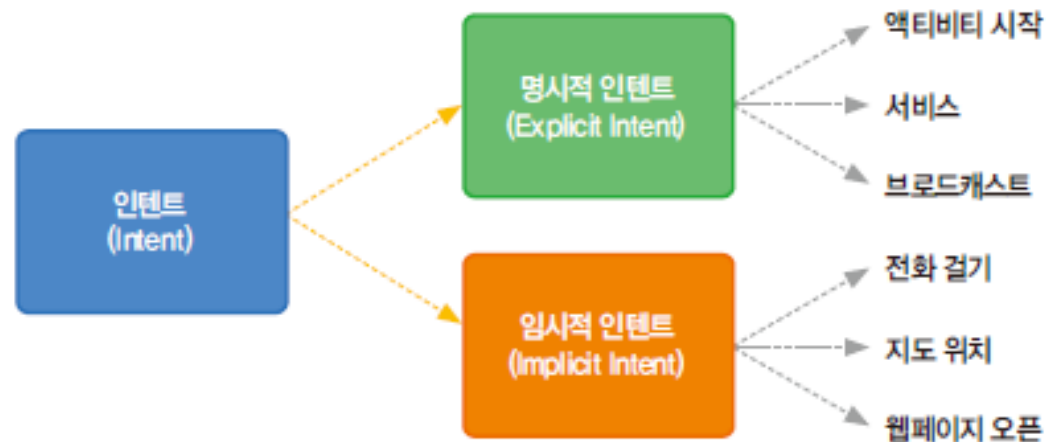
용도	설명
액티비티 시작	인텐트 객체를 <code>startActivity()</code> 메서드에 전달하면 새로운 액티비티 작업을 시작(수행)합니다.
서비스 시작	인텐트 객체를 <code>startService()</code> 메서드에 전달하면 새로운 서비스를 시작합니다. 기존 서비스에 필요한 요청을 보낼 수도 있습니다.
브로드캐스트 전달	인텐트 객체를 <code>sendBroadcast()</code> 메서드에 전달하면 브로드캐스트 리시버에 메시지를 보냅니다.

# 1. 인텐트 살펴보기

## ■ 인텐트의 유형

- 명시적 인텐트(Explicit Intent)
  - 정보를 전달할 때 클래스 객체나 구성 요소 이름을 지정하므로 호출할 대상을 확실히 알 수 있음
- 암시적 인텐트(Implicit Intent)
  - 호출할 대상의 속성은 지정하지만, 호출할 대상이 달라질 수 있음

그림 6-1 인텐트의 유형



02

# 명시적 인텐트로 화면 전환하기

## 2. 명시적 인텐트로 화면 전환하기

### ■ 명시적 인텐트

- 액티비티가 호출할 대상이나 구성 요소(클래스명)를 구체적으로 명시
- 시작하려는 액티비티나 서비스의 클래스명을 알고 있을 때 명시적 인텐트를 사용하여 구성 요소를 직접 시작
- 사용자 작업에 대한 응답으로 토스트 메시지, 상태 바, 대화상자와 같은 알림을 활성화하거나 미리 지정한 시간에 알람을 울리는 등의 작업 수행
- 명시적 인텐트로 액티비티를 지정하면 안드로이드 시스템이 지정된 앱의 구성 요소를 즉시 시작

그림 6-2 명시적 인텐트의 작동 방식





## 2. 명시적 인텐트로 화면 전환하기

### ■ 액티비티 이동

```
public Intent (Context packageContext, Class<?> cls)
```

표 6-2 명시적 인텐트 생성자의 매개 변수

매개 변수	설명
Context packageContext	이 클래스를 구현하는 애플리케이션 패키지의 컨텍스트(Context)입니다.
Class<?> cls	인텐트에 사용할 구성 요소의 클래스입니다.

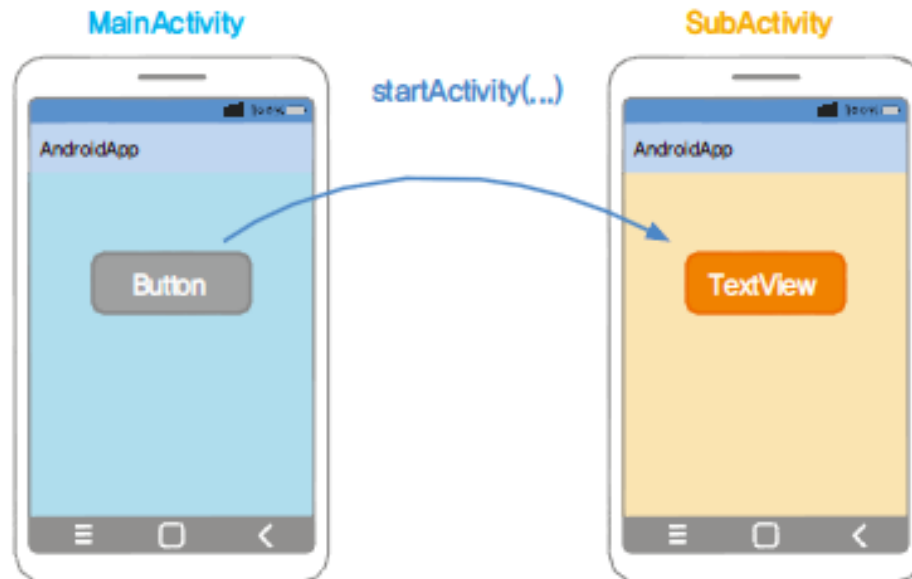
```
void startActivity(Intent intent)
```

## 2. 명시적 인텐트로 화면 전환하기

### ■ 액티비티 이동

```
Intent intent = new Intent(getApplicationContext(), SubActivity.class);  
startActivity(intent);
```

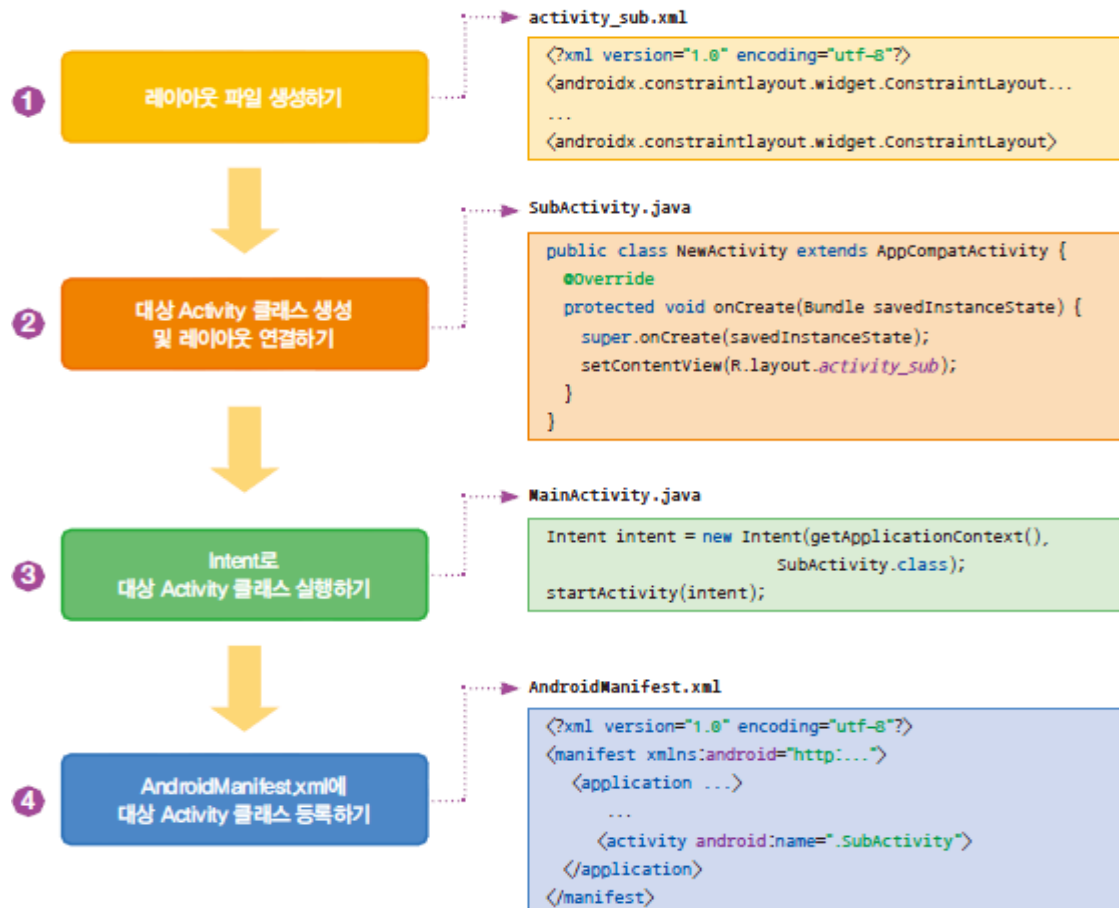
그림 6-3 명시적 인텐트로 액티비티 이동하기 1



## 2. 명시적 인텐트로 화면 전환하기

### ■ 액티비티 이동

그림 6-4 명시적 인텐트로 액티비티 이동하기 2



## 2. 명시적 인텐트로 화면 전환하기

### ■ 액티비티 이동

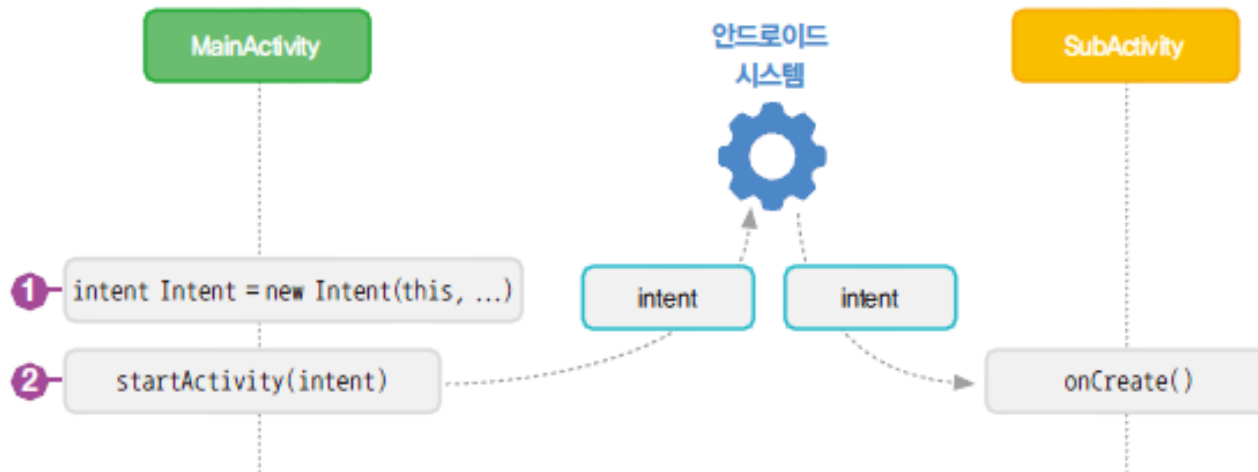


그림 6-5 명시적 인텐트로 액티비티 이동하기 3



첫 번째 액티비티



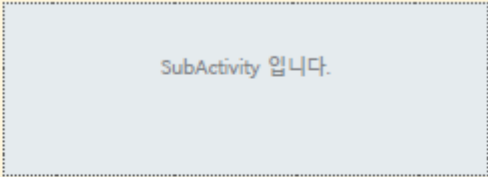
두 번째 액티비티

## 2. 명시적 인텐트로 화면 전환하기

### ■ 액티비티 이동

- 대상 액티비티의 레이아웃 파일 생성하기

```
<androidx.constraintlayout.widget.ConstraintLayout ...>  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="SubActivity 입니다."  
        app:layout_constraintLeft_toLeftOf="parent"  
        app:layout_constraintRight_toRightOf="parent"  
        app:layout_constraintTop_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```



SubActivity 입니다.

## 2. 명시적 인텐트로 화면 전환하기

### ■ 액티비티 이동

- 액티비티를 생성하고 레이아웃 연결하기

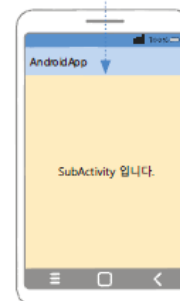
```
import androidx.appcompat.app.AppCompatActivity;

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

그림 6-12 레이아웃 파일 연결

SubActivity.java

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



res/layout/activity\_sub.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout...>

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=" SubActivity 입니다."
        .../>

</androidx.constraintlayout.widget.ConstraintLayout>
```

## 2. 명시적 인텐트로 화면 전환하기

### ■ 액티비티 이동

#### ■ 대상 액티비티 실행하기

```
public class MainActivity extends AppCompatActivity {

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

        Button button = (Button)findViewById(R.id.button); // 버튼 아이디 가져와 저장
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), SubActivity.class);
                // 두 번째 액티비티로 이동할 인텐트 설정
                startActivity(intent); // 액티비티 호출(실행)
            }
        });
    }
}
```

```
<androidx.constraintlayout.widget.ConstraintLayout ...>
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="버튼"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



## 2. 명시적 인텐트로 화면 전환하기

### ■ 액티비티 이동

- AndroidManifest.xml에 대상 액티비티 등록하기

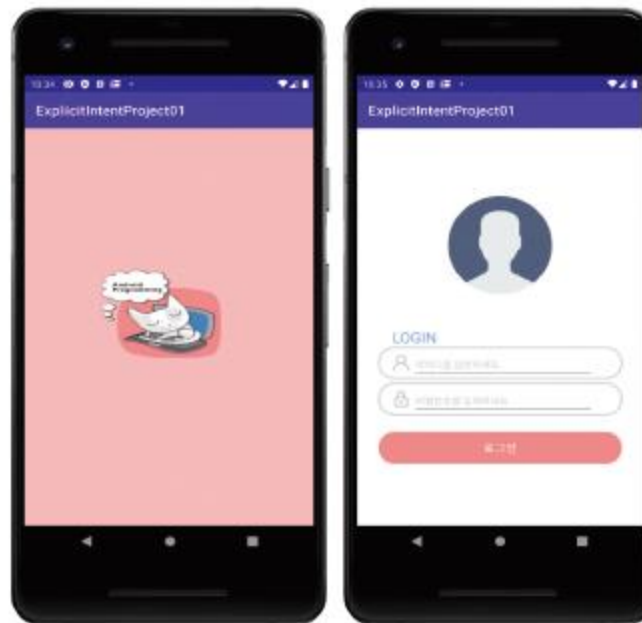
```
<manifest...>
  <application ...>
    <activity android:name=".MainActivity">
      ...
    </activity>
    <activity android:name=".SubActivity" />
  </application>
</manifest>
```



## 2. 명시적 인텐트로 화면 전환하기

### ■ [실습 예제 6-1] 스플래시 화면 만들기

- 구현 내용 : 스플래시 화면을 만듭니다. 첫 번째 화면(스플래시 화면)에서 2초 후에 두 번째 화면(로그인 화면)으로 이동합니다.



## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 전달하는 단방향 통신

- 한 액티비티에서 다른 액티비티로 정보를 전달하기 위해 명시적 인텐트 사용
  - `putExtra()` : 데이터를 보내는 데 사용
  - `getXxxExtra()` : 데이터를 받는 데 사용

그림 6-13 명시적 인텐트로 단방향 통신하기

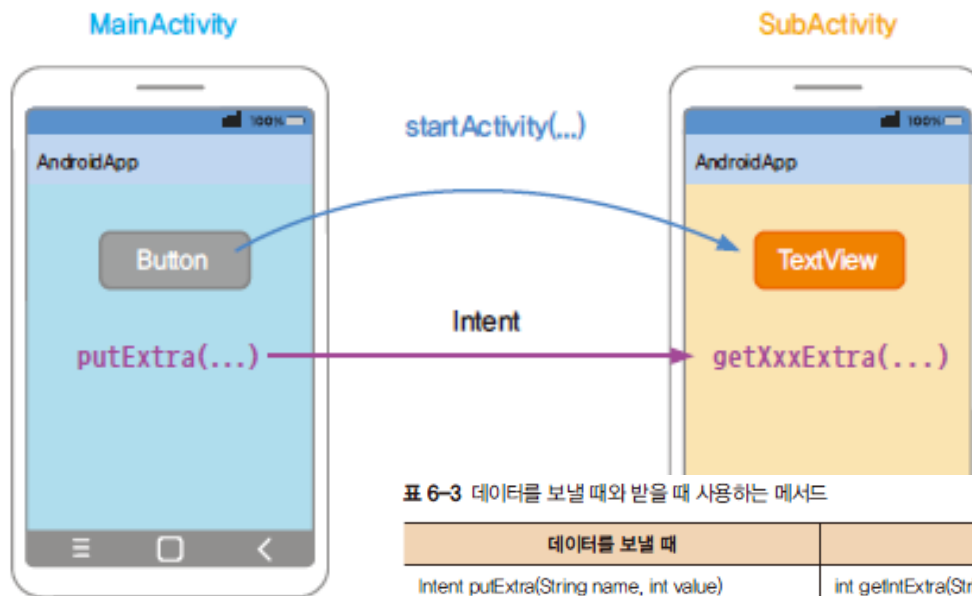


표 6-3 데이터를 보낼 때와 받을 때 사용하는 메서드

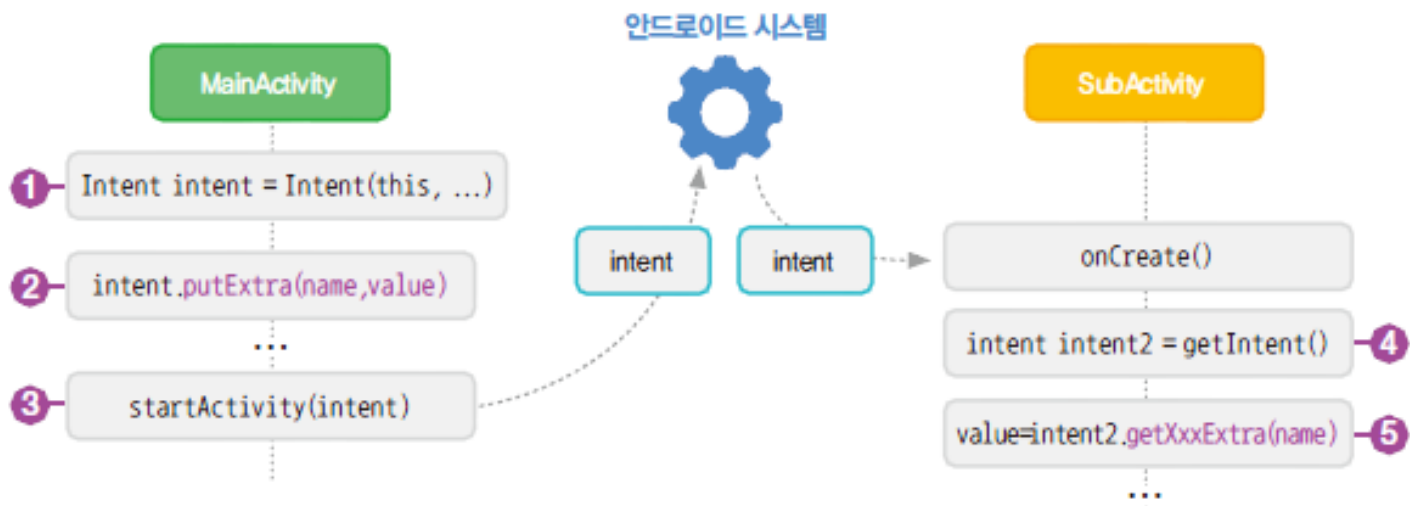
데이터를 보낼 때	데이터를 받을 때
<code>Intent.putExtra(String name, int value)</code>	<code>int getIntentExtra(String name, int defaultValue)</code>
<code>Intent.putExtra(String name, String value)</code>	<code>String getStringExtra(String name, String defaultValue)</code>
<code>Intent.putExtra(String name, boolean value)</code>	<code>boolean getBooleanExtra(String name, boolean defaultValue)</code>

## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 전달하는 단방향 통신

- 1 인텐트 객체를 생성합니다.
  - 2 putExtra() 메서드로 데이터를 전달합니다.
  - 3 startActivity() 메서드로 대상 액티비티를 실행합니다.
  - 4 전달되는 인텐트 객체를 getIntent() 메서드로 받습니다.
  - 5 인텐트 객체로 전달된 데이터 값을 getXxxExtra() 형식의 메서드로 받습니다.
- 첫 번째 액티비티
- 두 번째 액티비티

그림 6-14 명시적 인텐트로 데이터를 전달하는 과정



## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 전달하는 단방향 통신

그림 6-15 명시적 인텐트로 데이터 전달하기

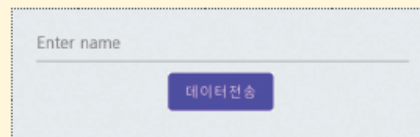


## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 전달하는 단방향 통신

#### ■ 데이터 보내기

```
<LinearLayout ...>
    <EditText
        android:id="@+id/editText"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter name" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="데이터전송" />
</LinearLayout>
```



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

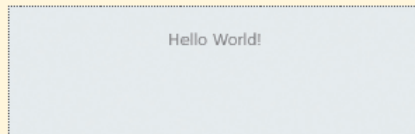
        button = findViewById(R.id.button);
        editText = findViewById(R.id.editText);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), SubActivity.class); ❶
                intent.putExtra("name", editText.getText().toString()); ❷
                startActivity(intent); ❸
            }
        });
    }
}
```

## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 전달하는 단방향 통신

#### ■ 데이터 받기

```
<LinearLayout..>
  <TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
</LinearLayout>
```



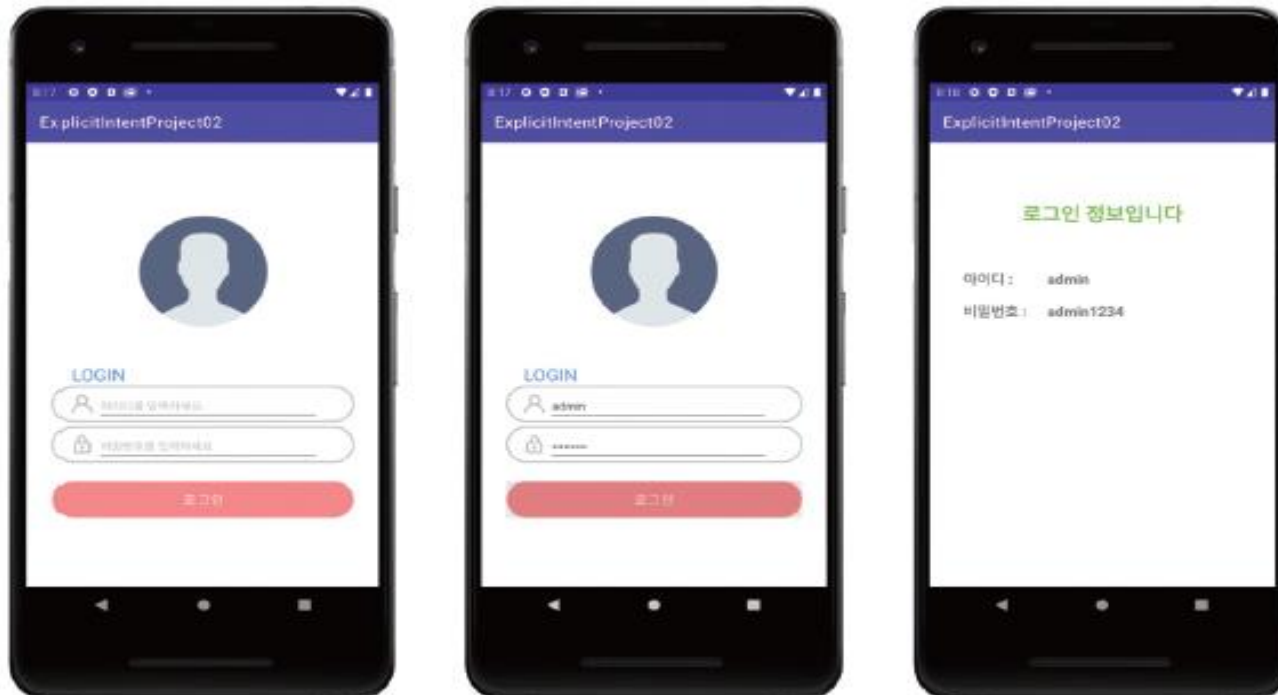
```
public class SubActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sub);

        TextView textView = findViewById(R.id.textView);
        Intent intent = getIntent(); ④
        String strName = intent.getStringExtra("name"); ⑤
        textView.setText(strName);
    }
}
```

## 2. 명시적 인텐트로 화면 전환하기

### ■ [실습 예제 6-2] 로그인 화면에서 아이디와 비밀번호 전달받아 출력하기

- 구현 내용 : 첫 번째 액티비티인 로그인 화면에서 아이디와 비밀번호를 입력하고 < 로그인> 버튼을 클릭하면 이를 두 번째 화면으로 전달해 아이디와 비밀번호를 출력합니다



## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 주고받는 양방향 통신

❶ 첫 번째 액티비티에서 인텐트 객체를 생성합니다.

❷ `registerForActivityResult()` 메서드를 호출하여 `ActivityResultLauncher` 객체를 초기화합니다.

❸ `ActivityResultLauncher` 객체 변수의 `launch()` 메서드를 통해 두 번째 액티비티를 실행합니다.

첫 번째 액티비티

❹ 두 번째 액티비티에서 인텐트 객체를 생성합니다.

❺ `putExtra()` 메서드로 데이터를 전달합니다.

❻ `setResult()` 메서드로 생성된 인텐트 객체를 전달합니다.

❼ `finish()` 메서드를 호출해 액티비티를 종료합니다.

두 번째 액티비티

❽ `registerForActivityResult()` 메서드 내 `ActivityResultCallback`의 `onActivityResult()` 메서드로 결과 정보를 확인합니다.

❾ `getXxxExtra()` 메서드를 사용해 인텐트로 전달된 데이터를 받습니다.

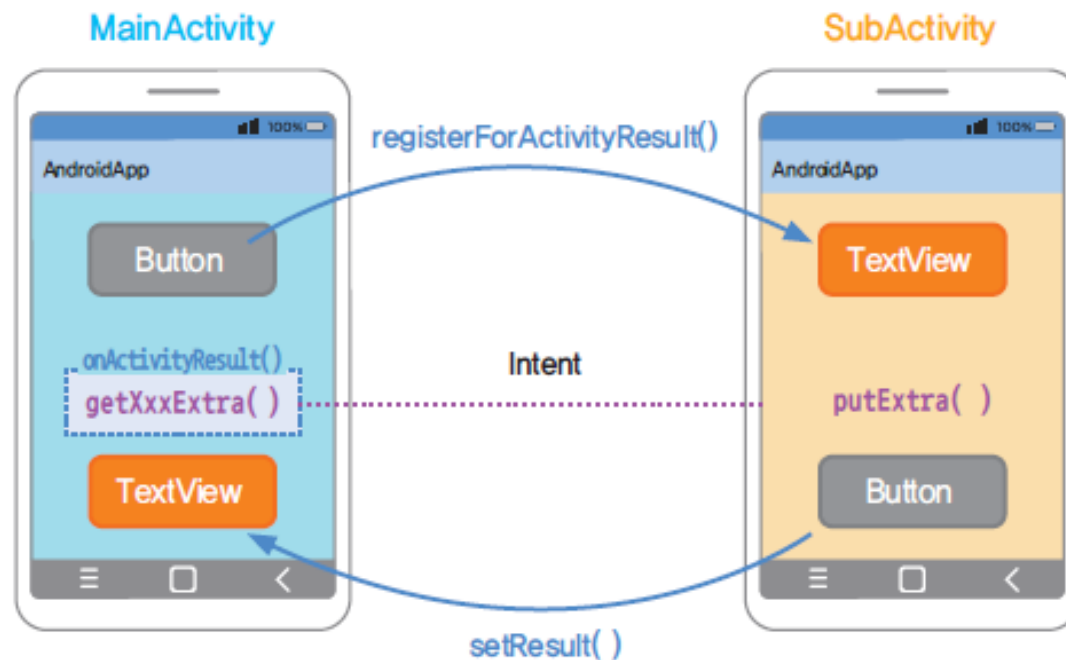
첫 번째 액티비티



## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 주고받는 양방향 통신

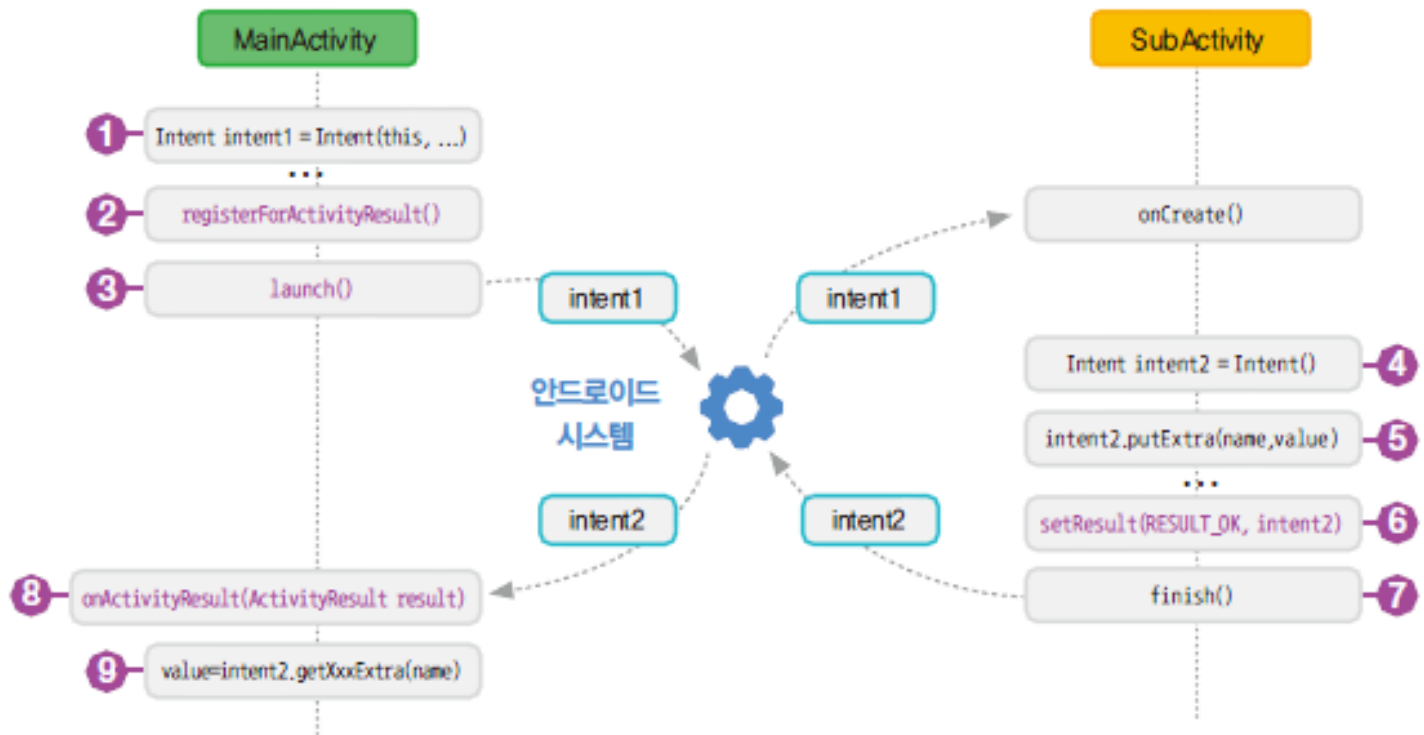
그림 6-16 명시적 인텐트로 양방향 통신하기



## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 주고받는 양방향 통신

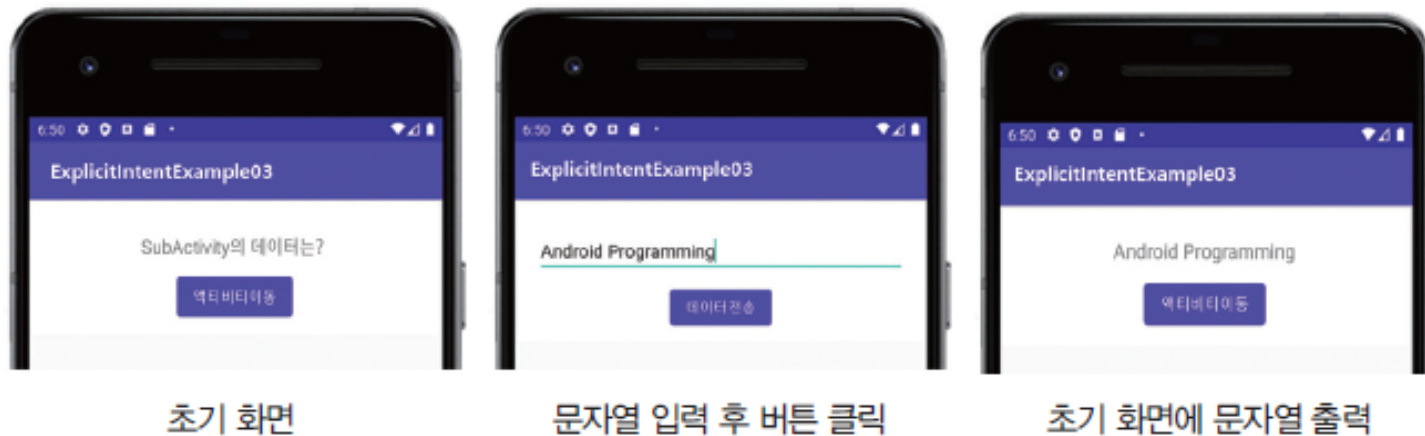
그림 6-17 명시적 인텐트를 사용해 양방향으로 데이터를 전달하는 과정



## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 주고받는 양방향 통신

그림 6-18 명시적 인텐트로 데이터 주고받기



## 2. 명시적 인텐트로 화면 전환하기

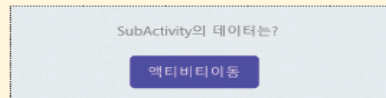
### ■ 데이터를 주고받는 양방향 통신

- 첫 번째 액티비티에서 두 번째 액티비티로 이동하기

```
<LinearLayout ...>
  <TextView
    android:id="@+id/textView"

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="SubActivity의 데이터는?"
    android:textSize="18sp" />

  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="액티비티 이동" />
</LinearLayout>
```



```
public class MainActivity extends AppCompatActivity {
    ActivityResultLauncher<Intent> launcher;

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

        Button button = findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), SubActivity.class);
                // 인텐트 객체 생성 ❶
                launcher.launch(intent); // 두 번째 액티비티 실행 ❸
            }
        });

        launcher = registerForActivityResult(new ActivityResultContracts.
            StartActivityForResult(), null); // ActivityResultLauncher 객체 초기화 ❷
    }
}
```

## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 주고받는 양방향 통신

- 두 번째 액티비티에서 첫 번째 액티비티로 데이터 보내기

```
<LinearLayout ...>
  <EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:hint="전송 데이터를 입력하세요"
    android:inputType="textPersonName" />
  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="데이터 전송" />
</LinearLayout>
```

전송 데이터를 입력하세요

데이터 전송

```
public class SubActivity extends AppCompatActivity {

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

        Button button = findViewById(R.id.button); // 버튼 아이디 저장
        EditText editText = findViewById(R.id.editText);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                EditText editText = findViewById(R.id.editText); // 에디트 텍스트 아이디 저장
                Intent intent = new Intent(); // 인텐트 객체 생성 ④
                intent.putExtra("editText", editText.getText().toString()); // 데이터 전달 ⑤

                setResult(RESULT_OK, intent); // 생성된 인텐트 객체 전달 ⑥
                finish(); // 액티비티 실행 종료 ⑦
            }
        });
    }
}
```

## 2. 명시적 인텐트로 화면 전환하기

### ■ 데이터를 주고받는 양방향 통신

- 두 번째 액티비티에서 받은 데이터 출력하기

```
public class MainActivity extends AppCompatActivity {
   ActivityResultLauncher<Intent> launcher; ❸
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button button = findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), SubActivity.class);
                // 인텐트 객체 생성 ❶
                launcher.launch(intent); // 두 번째 액티비티 실행 ❸
            }
        });
    }
}
```

```
launcher = registerForActivityResult(
    new ActivityResultContracts.StartActivityForResult(),
    new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            if (result.getResultCode() == RESULT_OK) {
                Intent data = result.getData();
                TextView textView = findViewById(R.id.textView);
                textView.setText(data.getStringExtra("editText")); ❹
            }
        }
    });
```

## 2. 명시적 인텐트로 화면 전환하기

### ■ [실습 예제 6-3] 같은 모양 찾기 게임에서 결과 확인하기

- 구현 내용 : 첫 번째 화면에서 원하는 도형을 선택하고 <시작하기> 버튼을 클릭 하면 두 번째 화면으로 이동합니다. 두 번째 화면에서는 첫 번째 화면에서 선택한 도형과 같은 그림을 3개까지 선택합니다. <결과보기> 버튼을 클릭하면 다시 첫 번째 화면으로 이동해 선택한 그림을 출력합니다. 명시적 인텐트로 예제를 구현해 보시다.



# 03

## 암시적 인텐트로 화면 전환하기



### 3. 암시적 인텐트로 화면 전환하기

#### ■ 암시적 인텐트(Implicit Intent)

- 구성 요소의 이름을 직접 지정하는 대신, 수행할 작업을 선언하여 해당 작업을 수행할 수 있는 구성 요소가 작업을 처리하도록 하는 방식

그림 6-19 암시적 인텐트의 작동 방식



### 3. 암시적 인텐트로 화면 전환하기

#### ■ 암시적 인텐트(Implicit Intent)

```
Intent(String action)
Intent(String action, Uri uri)
```

표 6-4 암시적 인텐트 생성자의 매개 변수

매개 변수	설명
action	인텐트의 액션입니다.
uri	인텐트의 데이터입니다.

### 3. 암시적 인텐트로 화면 전환하기

#### ■ 암시적 인텐트(Implicit Intent)

표 6-5 암시적 인텐트에서 사용할 수 있는 주요 액션

클래스	액션	설명
Activity	ACTION_VIEW	사용자에게 데이터를 표시합니다.
	ACTION_DIAL	데이터에 지정된 번호로 전화를 겁니다. 사용자에게는 통화를 시작할 수 있도록 번호가 지정된 UI가 표시됩니다.
	ACTION_CALL	데이터에 지정된 사람에게 전화를 겁니다.
	ACTION_PICK	데이터에서 항목을 선택하고 선택한 항목을 반환합니다.
	ACTION_SEND	일부 데이터를 다른 사람에게 전달합니다. 데이터가 전달되는 대상이 지정되지 않으며, 데이터를 어디로 보내야 하는지 묻는 것은 이 액션의 수신자에게 달려 있습니다.
MediaStore	ACTION_IMAGE_CAPTURE	카메라 앱에서 이미지를 캡처해 반환하도록 전송할 수 있는 표준 인텐트 액션입니다.
	ACTION_REVIEW	지정된 미디어 파일을 검토하기 위해 전송할 수 있는 표준 인텐트 액션입니다.
	ACTION_VIDEO_CAPTURE	카메라 앱에서 비디오를 캡처해 반환하도록 전송할 수 있는 표준 인텐트 액션입니다.
AlarmClock	ACTION_SHOW_ALARMS	경보를 표시하며, 이 액션은 알람 페이지를 엽니다.
	ACTION_SHOW_TIMERS	타이머를 표시하며, 이 액션은 타이머 페이지를 엽니다.

## ■ 암시적 인텐트(Implicit Intent)

### ■ 전화 걸기 예

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(Intent.ACTION_DIAL);
        intent.setData(Uri.parse("tel:01012341234"));
        startActivity(intent);
    }
});
```



### 3. 암시적 인텐트로 화면 전환하기

#### ■ 암시적 인텐트(Implicit Intent)

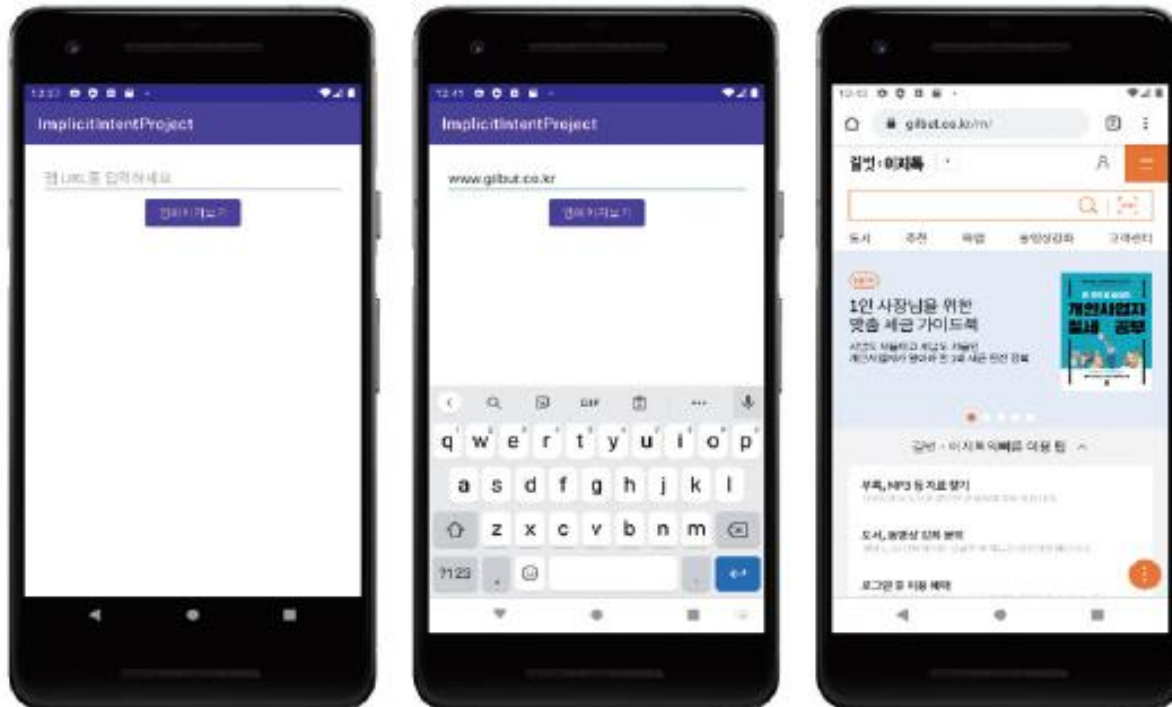
표 6-6 암시적 인텐트로 수행할 수 있는 작업

작업	예시 코드
전화걸기	<pre>Intent intent = new Intent(Intent.ACTION_DIAL); intent.setData(Uri.parse("tel:01012341234")); startActivity(intent); &lt;uses-permission android:name="android.permission.CALL_PHONE" /&gt;</pre>
통화목록 보기	<pre>Intent intent = new Intent(Intent.ACTION_VIEW); intent.setType(CallLog.Calls.CONTENT_TYPE); startActivity(intent);</pre>
연락처 검색	<pre>Intent intent = new Intent(Intent.ACTION_PICK); intent.setType(ContactsContract.Contacts.CONTENT_TYPE); startActivity(intent);</pre>
브라우저 열기	<pre>Intent intent = new Intent(Intent.ACTION_VIEW); intent.setData(Uri.parse("https://tutorial.eyehunts.com/")); startActivity(intent);</pre>
갤러리 보기	<pre>Intent intent = new Intent(Intent.ACTION_VIEW); intent.setData(Uri.parse("content://media/external/images/media/")); startActivity(intent);</pre>
카메라 실행	<pre>Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE); startActivity(intent);</pre>
알람 올리기	<pre>Intent intent = new Intent(AlarmClock.ACTION_SHOW_ALARMS); startActivity(intent);</pre>

### 3. 암시적 인텐트로 화면 전환하기

#### ■ [실습 예제 6-4] 웹 페이지로 이동하기

- 구현 내용 : 첫 번째 화면에서 원하는 URL을 입력한 후 <웹페이지보기> 버튼을 클릭하면 브라우저 앱이 실행되어 입력한 URL의 웹 페이지를 출력합니다.



04

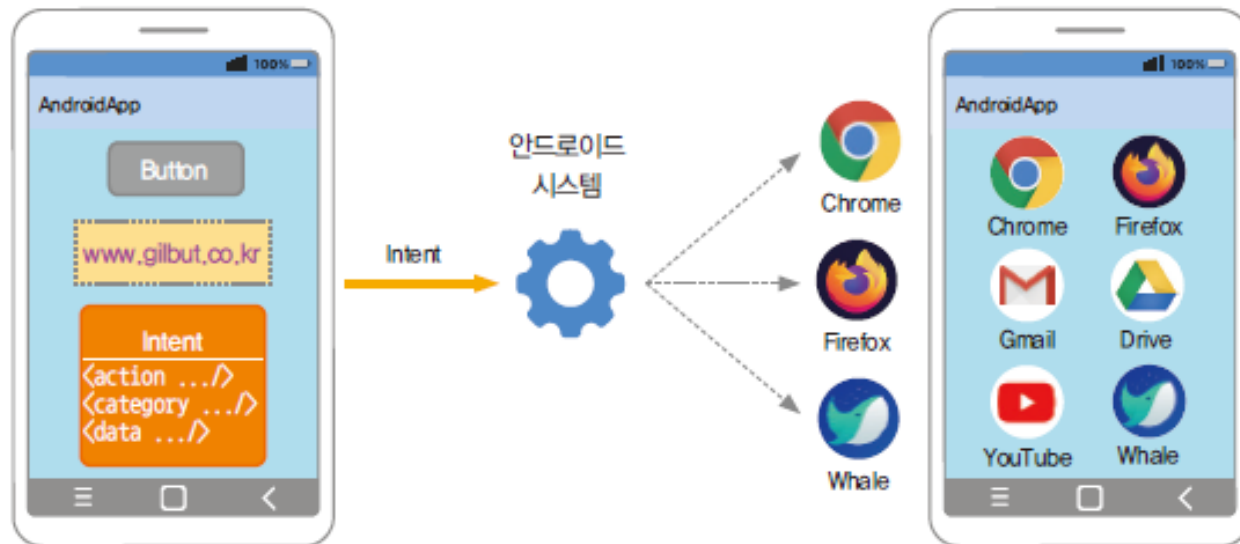
# 인텐트 필터 다루기

## 4. 인텐트 필터 다루기

### ■ 인텐트 필터(Intent Filter)

- 구성 요소가 수신하는 인텐트의 유형을 지정하는 데 사용
- 주로 암시적 인텐트에서 사용자가 기기에 설치된 앱 중에서 어느 앱을 사용할지 선택하는 데 사용
- 앱에서 암시적 인텐트를 수신하려면 앱의 구성 요소에 대해 하나 이상의 인텐트 필터를 선언해야 하며, 이는 안드로이드 정의 파일(ActivityManifest.xml)에 정의

그림 6-20 인텐트 필터의 작동 방식 예





## 4. 인텐트 필터 다루기

### ■ 인텐트 필터(Intent Filter)

표 6-7 인텐트 필터의 요소

요소	설명
<action>	실행할 인텐트 동작을 정의합니다. ACTION_CALL, ACTION_MAIN, ACTION_BATTERY_LOW
<category>	실행할 인텐트 카테고리의 이름을 정의합니다. CATEGORY_BROWSABLE, CATEGORY_LAUNCHER
<data>	실행할 데이터 유형을 정의합니다. 하나 이상의 속성을 사용해 데이터 URI와 MIME 유형을 정의합니다.

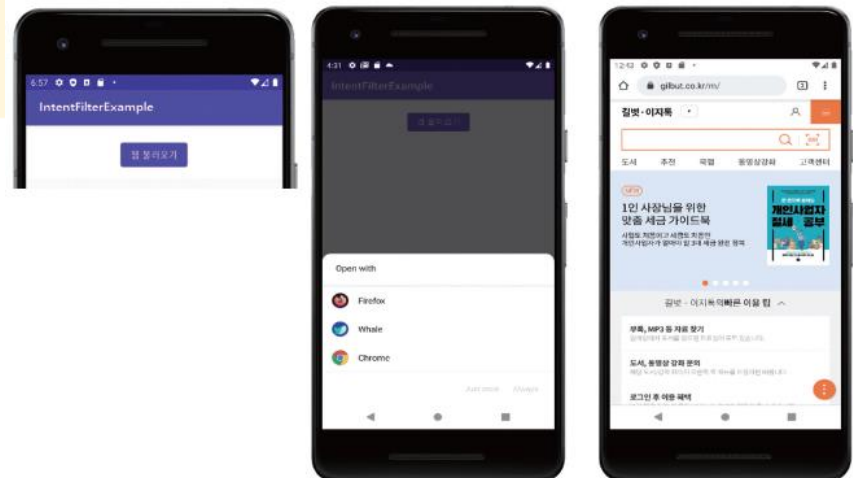
```
<activity android:name=".MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
    <data android:mimeType="text/plain" />
  </intent-filter>
</activity>
```

## ■ 인텐트 필터(Intent Filter)

### ■ 예

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String url = editText.getText().toString();
        Intent intent = new Intent(Intent.ACTION_VIEW);
        intent.setData(Uri.parse("http://www.gibut.co.kr"));

        startActivity(intent);
    }
});
```



## 4. 인텐트 필터 다루기

### ■ [실습 예제 6-5] 인텐트 필터를 이용해 메일 보내기

- 구현 내용 : 첫 번째 화면에서 <메일보내기> 버튼을 클릭하면 메일을 작성할 수 있는 앱이 실행됩니다.

