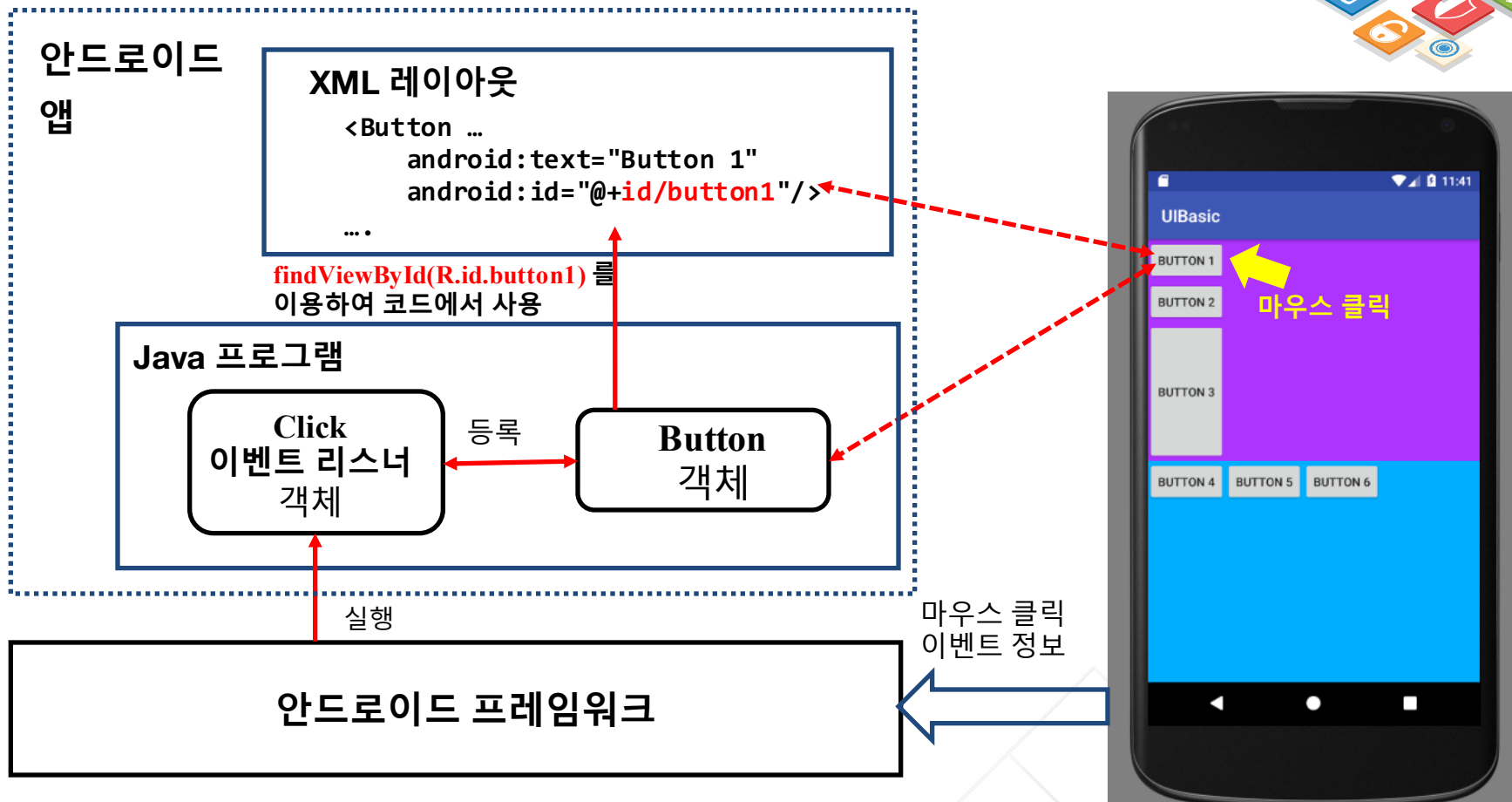




05

# 이벤트 처리





## Event Listener

### ◆ View 클래스 내의 interface로 하나의 Callback 메소드를 포함

- `View.OnClickListener`
  - `onClick()`: 사용자가 해당 View를 클릭하거나 터치할 때 호출
- `View.OnLongClickListener`
  - `onLongClick()`: 사용자가 해당 View를 길게 클릭할 때 호출
- `View.OnFocusChangeListener`
  - `onFocusChange()` : 사용자의 초점이 해당 View 위로 이동하거나 해당 View로부터 벗어날 때 호출
- `View.OnKeyListener`
  - `onKey()`: 사용자가 해당 View에 초점이 있는 상태에서 키를 누른 경우에 호출
- `View.onTouchListener`
  - `onTouch()`: 해당 View의 범위 내에서 일어나는 사용자의 행위(누르기, 손 떼기와 화면에서 이루어지는 모든 움직임 동작이 포함)가 발생할 때 호출됨



## Event Listener 객체 등록

- ◆ 이벤트 리스너 interface의 Callback 메소드를 구현한 객체는 해당 View에 사전에 등록되어야 함.
  - 이벤트 리스너 등록 방법
    - View.setOnClickListener(View.OnClickListener)
    - View.setOnLongClickListener(View.OnLongClickListener)
    - View.setOnFocusChangeListener(View.OnFocusChangeListener)
    - View.setOnKeyListener(View.OnKeyListener)
    - View.setOnTouchListener(View.OnTouchListener)
- ◆ 해당 View와 사용자가 상호작용할 때, **안드로이드 프레임워크**에서 Callback 메소드 호출



## Event Listener Example

```
public class MainActivity extends AppCompatActivity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.linear_layout);

        Button btn = (Button)findViewById(R.id.button1);
        btn.setOnClickListener(new MyOnClickListener());
    }
    private class MyOnClickListener implements

        View.OnClickListener {
            public void onClick(View view) {
                Toast.makeText(getApplicationContext(),
                    R.string.button_clicked_msg,
                    Toast.LENGTH_SHORT).show();
            }
        }
    /* void onClickB1(View v) { // 토스트 메시지 출력 } */
}
```

