

Android application 만들기

4. Widget-기본

TextView

Button

Listener 구현하기

Android widget

부모 클래스는 `android.view.View` 이다. `id` 등의 각종 기본 기능이 `View` 클래스에 구현되어 있다.

Android의 모든 `Widget`과 `Layout`들은 `View`의 자식 타입이다.

`int`형의 아이디를 가질 수 있으며 `width`, `height`에 대한 설정을 가진다.

클릭, 터치 등의 이벤트를 받을 수 있다.

`xml` 파일에서 정의하는 대부분의 속성은 `java` 코드로 제어 가능하며 주로 `get/set` 함수를 사용한다.

`java` 코드에서 위젯을 제어하기 위해서는 `id` 값이 반드시 필요하다.

`onCreate()`의 `setContentView()` 함수 호출 이후에 각 위젯을 사용할 수 있다.

Widget을 코드에서 사용하기

xml상의 위젯 태그 이름은 Java class 이름이며

setContentView 함수가 모두 실행되고 나면 각 위젯들의 객체가 메모리 상에 잡히게 된다.

이를 하나씩 불러 사용하는데 이 때 사용하는 함수가 findViewById 이다.

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

setContentView의 인자로 전달되는 xml 파일에 추가한 위젯들을 사용할 수 있게되며

setContentView 함수 호출 이전에 위젯을 설정하려고 하거나

인자로 전달 된 xml 파일에 없는 위젯을 설정하려 하면 런타임 에러가 발생한다.

TextView

정해진 문자열을 출력하는 위젯. 문자열을 지정하는 `android:text` 속성을 가진다.

레이아웃에서 하드코딩해서 지정할 수도 있으며 `string` 리소스의 아이디로 지정할 수 있다.

최종 버전은 다국어 지원 및 품질을 위해 `string` 리소스를 사용하는 코드로 변환하여 사용하는 것이 좋다.

하드 코딩 한 경우

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

strings.xml을 참조한 경우

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/app_name"
```

strings.xml

```
<resources>
    <string name="app_name">Hello World</string>
</resources>
```

TextView

대부분의 버튼들이 TextView를 상속 받으므로 다음 속성들은 버튼들도 사용한다.

주요 속성 (빨간 글씨체의 값이 기본 값)

`android:textSize="18sp"` // 생략할 경우 appTheme에 정의 된 값을 사용

`android:textColor="#123456"` 또는 `"@color/color_accent"` // 생략할 경우 appTheme 값을 따름

`android:textAlignment="center/viewStart/viewEnd"`

`android:visibility="visible/gone/invisible"`

`android:clickable="false/true"`

TextView: 실습

MainActivity.java 수정

실행해서 출력되는
문구를 확인한다.

```
package com.codesample.helloworld;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private TextView textView;

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

        textView = findViewById(R.id.textView);
        textView.setText("Simple Calculator!!");
    }
}
```

findViewById는 반드시
setContentView 다음에 사용

xml 에서 추가한 id 값

사용자 이벤트 처리

위젯들은 사용자의 이벤트(클릭, 드래그 등)를 처리 해야 함.

사용자가 버튼을 누른 시점을 아는 방법

1. Polling

짧은 간격을 두고 상태를 주기적으로 확인하는 방법.

2. Callback

리스너(Listener) 함수를 등록시킨 후 특정 이벤트가 발생하면 등록된 함수를 호출해 달라고 하는 방법.

안드로이드 위젯은 리스너를 사용한다.

리스너를 사용하기 위한 최소한의 규칙을 지정 -> **Interface**

안드로이드의 리스너

안드로이드의 위젯은 리스너를 이용해 사용자의 입력을 처리하며 위젯의 특성에 맞도록 Interface들이 정의 됨.

리스너를 구현하는 방법

1. Activity가 interface를 implement 하는 방법
2. Interface type의 익명 객체 리스너를 추가하는 방법
3. Interface type의 객체를 사용하는 방법
4. Interface를 implement한 별도의 클래스를 구현해 사용하는 방법 (잘 사용하지 않음)
5. 클릭 이벤트에 한해: Activity에 지정된 양식의 함수를 만들어 사용하는 방법

+버튼은 Activity가 implement한 함수로, - 버튼은 익명 리스너, *버튼은 객체, / 버튼은 5번 방법을 이용하여 다양한 방법으로 리스너를 등록해 본다.

안드로이드의 리스너

버튼을 비롯한 안드로이드의 모든 위젯들은 클릭 이벤트 리스너를 등록할 수 있으며 타입은 interface로 `android.view.View.OnClickListener`이다.

`android.view.View.java` 코드 부분

```
public interface OnClickListener {  
    /**  
     * Called when a view has been clicked.  
     *  
     * @param v The view that was clicked.  
     */  
    void onClick(View v);  
}
```

implement 해야 할 함수는 `void onClick(View v)` 이며 파라미터로 전달되는 `View`는 클릭 이벤트가 일어난 `view`이다.

OnClick: 실습

Activity가 implement 하기

MainActivity 선언부에 implements View.OnClickListener를 추가한다.

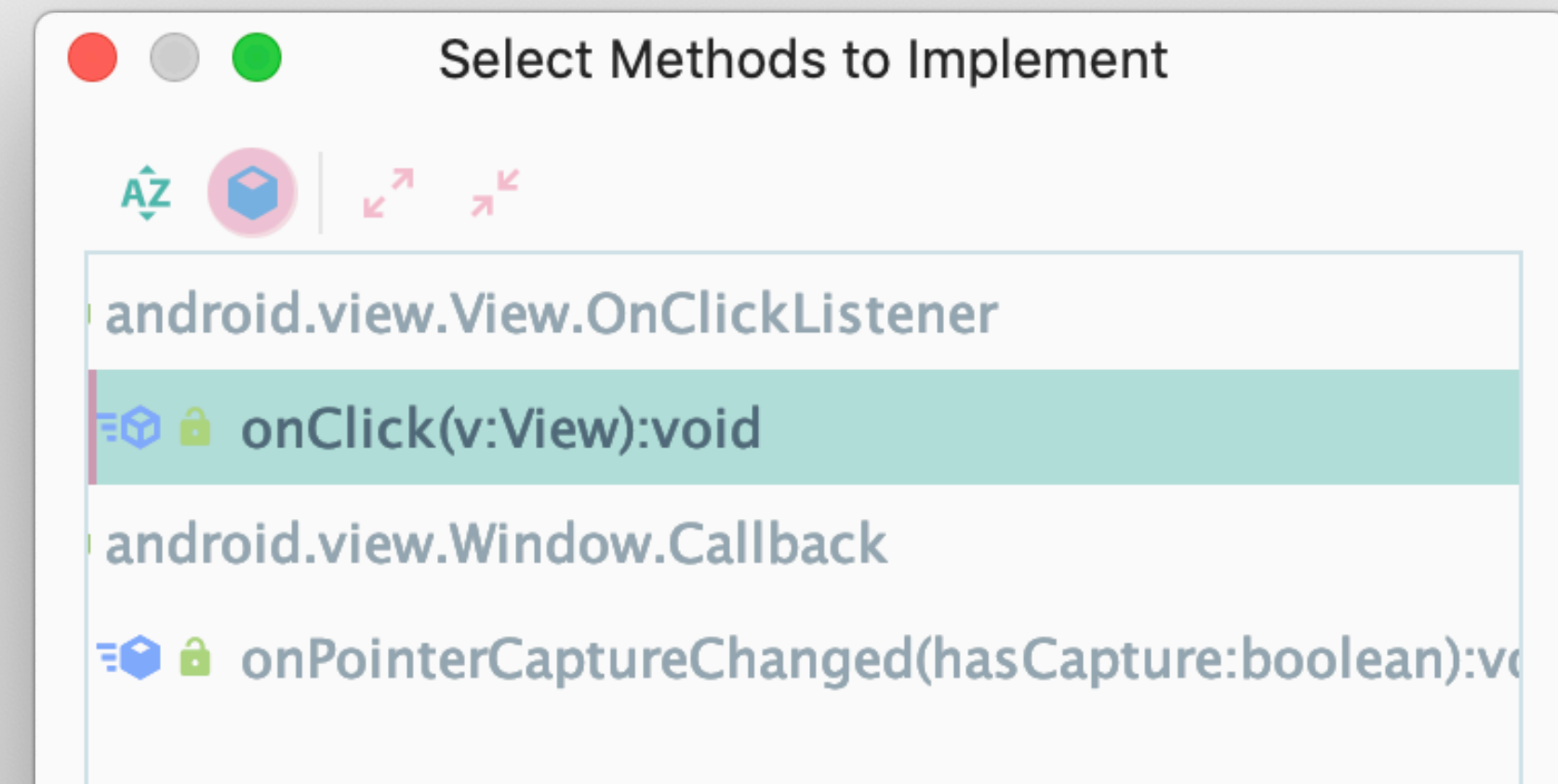
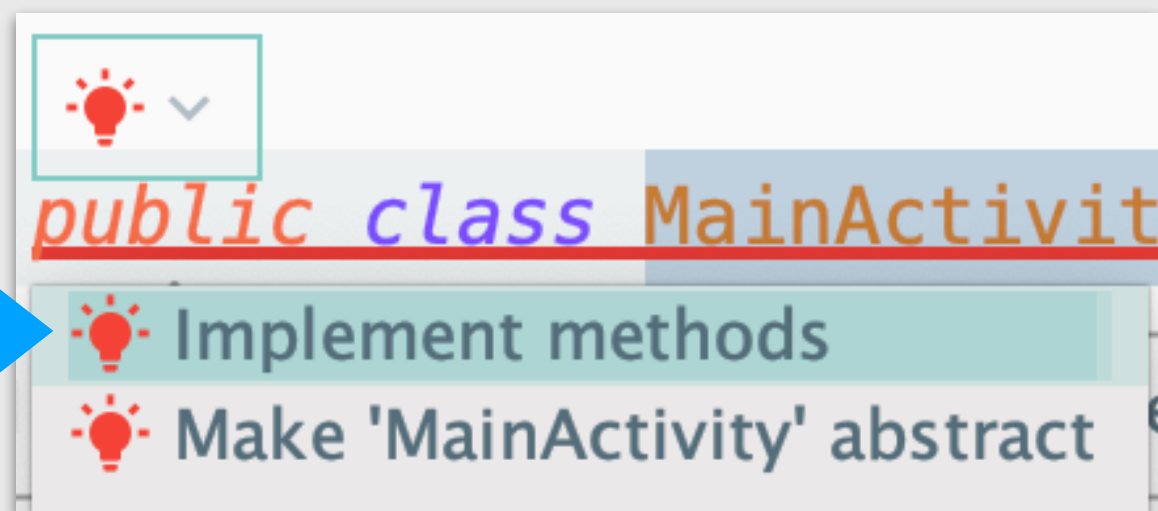
1



```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
```

클래스 이름에 커서를 두면 좌측 상단에 빨간 전구가 뜨는데 클릭한 다음 뜨는 메뉴에서 **Implement methods**를 선택한다. 이후 뜨는 팝업에서 **onClick**을 선택하면 된다.

2



OnClick: 실습

Activity가 implement 하기

메뉴를 사용하지 않고 직접 다음 함수를 추가해도 된다.

```
@Override  
public void onClick(View v) {  
  
}
```

위젯들을 불러 쓸 수 있도록 findViewById와 버튼에 리스너를 등록하는 코드를 작성한다.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{  
    private TextView textView;  
    private EditText editTextNum1, editTextNum2;  
    private Button buttonAdd, buttonSub, buttonMul, buttonDiv;
```

OnClick: 실습

Activity가 implement 하기

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

    textView = findViewById(R.id.textView);
    textView.setText("Simple Calculator!!");

    editTextNum1=findViewById(R.id.editTextNum1);
    editTextNum2=findViewById(R.id.editTextNum2);

    buttonAdd=findViewById(R.id.buttonAdd);
    buttonAdd.setOnClickListener(this);
}
```

OnClick: 실습

Activity가 implement 하기

```
@Override
public void onClick(View v) {
    float num1=0, num2=0;
    try {
        num1 = Float.parseFloat(editTextNum1.getText().toString());
        num2 = Float.parseFloat(editTextNum2.getText().toString());
    } catch (NumberFormatException e) {
        textView.setText("Invalid Number");
        return;
    }
    textView.setText(String.valueOf(num1+num2));
}
```

두 개 이상의 위젯이 각각 `setOnClickListener(this)`를 이용해 리스너를 공동으로 이용할 경우
파라미터로 넘어오는 `v.getId()`를 이용해 이벤트가 발생한 위젯을 구별할 수 있다.

OnClick: 실습

익명 객체 리스너 추가하기

onCreate 함수에 다음 코드 추가: buttonSub 관련 코드 추가

전통적인 방법

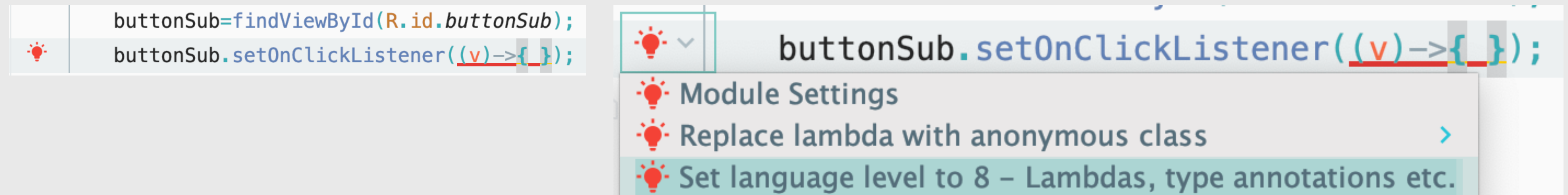
```
buttonSub=findViewById(R.id.buttonSub);
buttonSub.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        float num1=Float.parseFloat(editTextNum1.getText().toString());
        float num2=Float.parseFloat(editTextNum2.getText().toString());
        textView.setText(String.valueOf(num1-num2));
    }
});
```

OnClick: 실습

익명 객체 리스너 추가하기

람다식으로 구현하기

다음과 같이 수정 후 빨강 전구를 클릭 > Set language level 8을 선택



프로젝트가 새로 빌드되는 것을 기다리면 람다식을 사용할 수 있게 된다.

OnClick: 실습

익명 객체 리스너 추가하기

람다식으로 구현하기: 예외 처리 코드 생략 버전

```
buttonSub=findViewById(R.id.buttonSub);
buttonSub.setOnClickListener((v)->{
    float num1 = Float.parseFloat(editTextNum1.getText().toString());
    float num2 = Float.parseFloat(editTextNum2.getText().toString());
    textView.setText(String.valueOf(num1-num2));
});
```


OnClick: 실습

Interface type의 객체 만들어 사용하기

다음과 같이 클래스 변수를 추가

```
private Button buttonAdd, buttonSub, buttonMul, buttonDiv;
private View.OnClickListener listener = new View.OnClickListener(){
    @Override
    public void onClick(View v) {
        float num1=Float.parseFloat(editTextNum1.getText().toString());
        float num2=Float.parseFloat(editTextNum2.getText().toString());
        textView.setText(String.valueOf(num1*num2));
    }
};
```

```
private Button buttonAdd, buttonSub, buttonMul, buttonDiv;
private View.OnClickListener listener = (v)->{
    float num1=Float.parseFloat(editTextNum1.getText().toString());
    float num2=Float.parseFloat(editTextNum2.getText().toString());
    textView.setText(String.valueOf(num1*num2));
};
```

OnClick: 실습

Interface type의 객체 만들어 사용하기

onCreate 함수에 다음 코드 추가

```
buttonMul=findViewById(R.id.buttonMul);  
buttonMul.setOnClickListener(listener);
```

OnClick: 실습

특정 함수 만들어 클릭 이벤트 처리하기

위젯들이 처리하는 터치, 드래그 등의 다양한 이벤트는 앞서 설명한 리스너 인터페이스를 이용해 처리해야 함.
각 위젯마다 이벤트가 조금씩 다르므로 다양한 리스너를 사용함.

클릭 이벤트는 모든 위젯들의 공통 이벤트로, 클릭에 한해 다음과 같이 이벤트를 처리할 수 있음.

특정 형식을 지키는 함수를 Activity에 추가. (함수 이름과 파라미터 변수 이름은 자유)

```
public void func(View v){ }
```

레이아웃 xml에서 이를 등록해 사용.

OnClick: 실습

특정 함수 만들어 클릭 이벤트 처리하기

MainActivity에 다음 함수 추가

```
public void onDivision(View v){  
    float num1=Float.parseFloat(editTextNum1.getText().toString());  
    float num2=Float.parseFloat(editTextNum2.getText().toString());  
    if(num2==0) textView.setText("Divided by zero");  
    else textView.setText(String.valueOf(num1/num2));  
}
```

activity_main.xml 파일에서 @+id/buttonDiv 항목에 다음 코드 추가

```
<Button  
    android:id="@+id/buttonDiv"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="/"   
    android:onClick="onDivision"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/buttonMul" />
```

OnClick: 실습

비슷한 코드가 중복되는 문제점이 있음. MainActivity의 onClick 함수로 기능을 통합하는 리팩토링 진행

각 버튼은 `setOnClickListener(this)` 로 수정하면 됨

```
@Override
public void onClick(View v) {
    float num1=0, num2=0;
    int id = v.getId();
    try {
        num1 = Float.parseFloat(editTextNum1.getText().toString());
        num2 = Float.parseFloat(editTextNum2.getText().toString());
    } catch (NumberFormatException e) {
        textView.setText("Invalid Number");
        return;
    }
    if(id == R.id.buttonAdd){ num1 += num2; }
    else if(id == R.id.buttonSub){ num1 -= num2; }
    else if(id == R.id.buttonMul){ num1 *= num2; }
    else if(id == R.id.buttonDiv){
        if(num2 ==0) { textView.setText("Divided by zero"); return; }
        else num1 /= num2;
    }
    textView.setText(String.valueOf(num1));
}
```