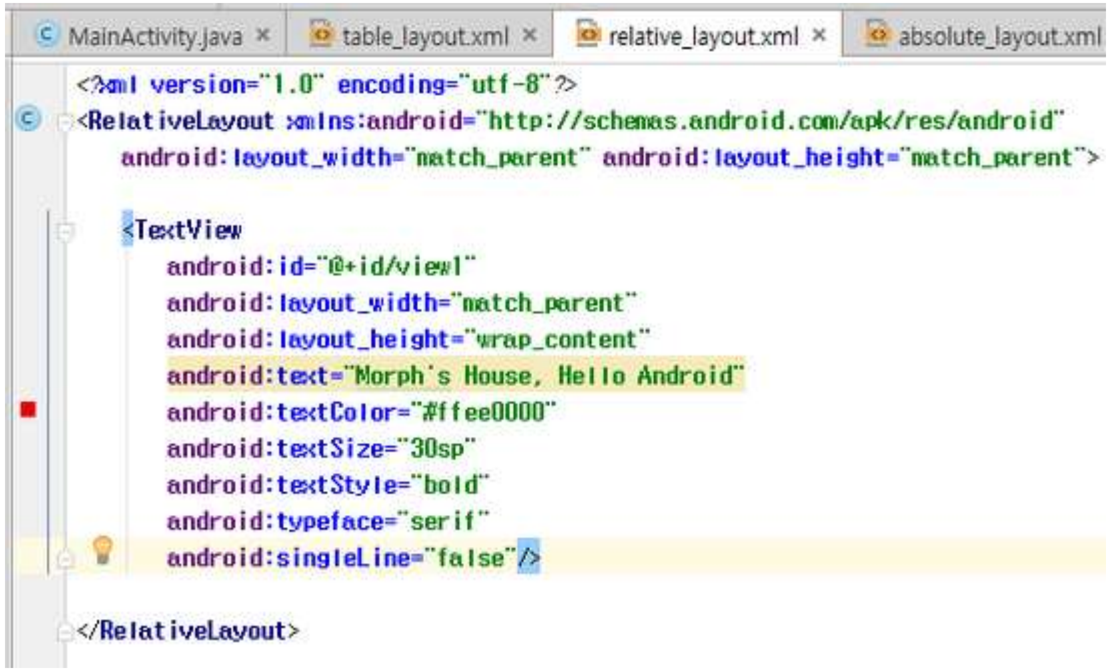
Binding

발표자: 장범준

Binding이란…

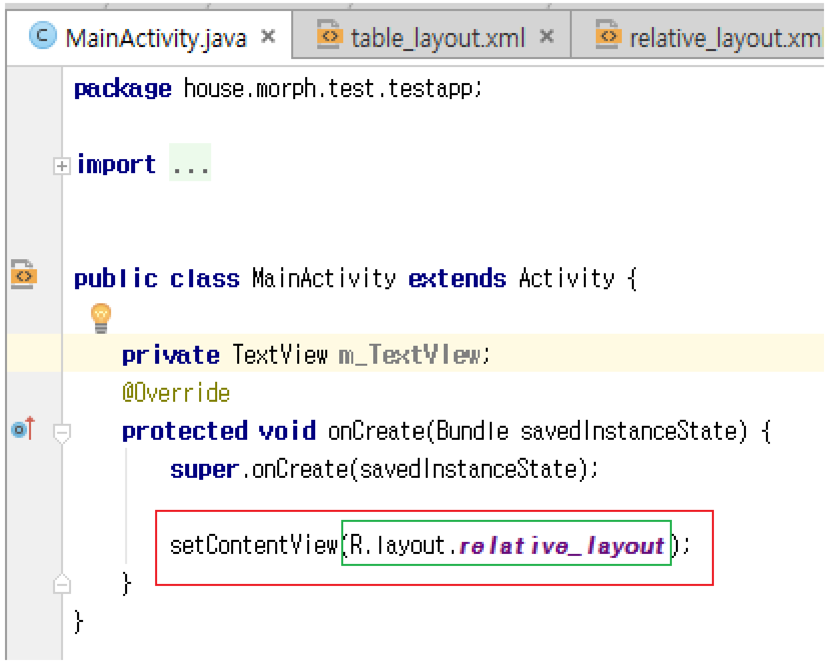
프로그램의 어떤 기본 단위가 가질 수 있는 구성요소의
구체적인 값, 성격을 확정하는 것을 말한다.

findViewById & setContentView



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent">

    <TextView
        android:id="@+id/view1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Morph's House, Hello Android"
        android:textColor="#ffee0000"
        android:textSize="30sp"
        android:textStyle="bold"
        android:typeface="serif"
        android:singleLine="false"/>

</RelativeLayout>
```

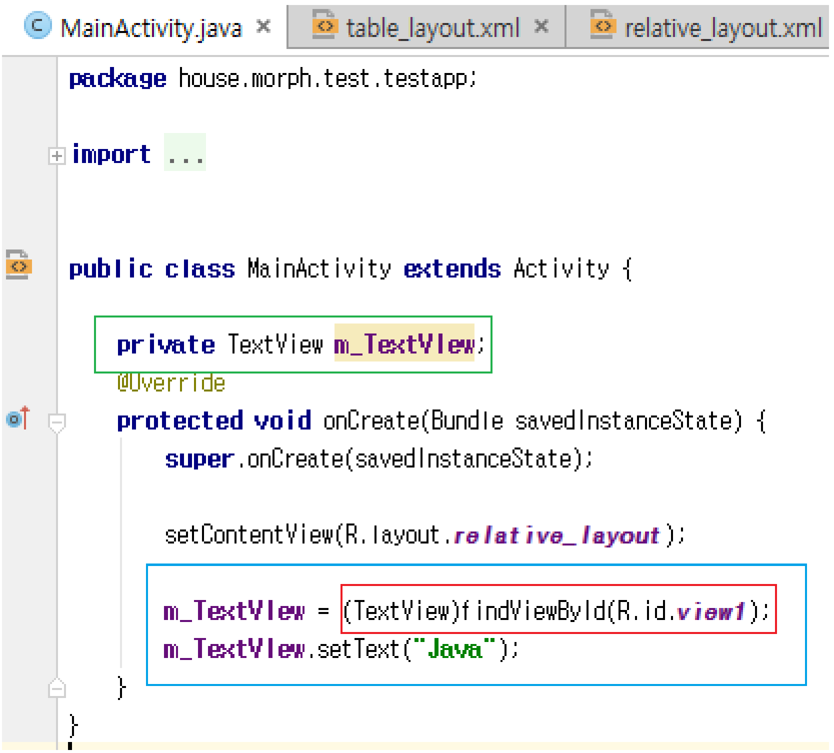
액티비티에 적용할
레이아웃 리소스



```
package house.morph.test.testapp;

import ...


public class MainActivity extends Activity {

    private TextView m_TextView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.relative_layout);
    }
}
```

리소스를 적용받은 액티비티

findViewById & setContentView



```java
package house.morph.test.testapp;

import ...


public class MainActivity extends Activity {

    private TextView m_TextView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.relative_layout);

        m_TextView = (TextView)findViewById(R.id.view1);
        m_TextView.setText("Java");
    }
}
```

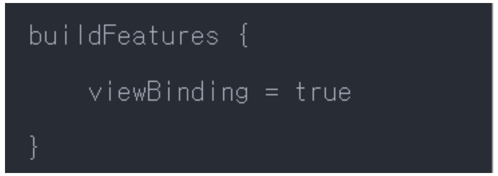
레이아웃 리소스를 연결했다 해서 이벤트를 뷰에 변화를 줄 순 없다.

이때, 사용하는게 findviewbyId입니다.

viewBinding

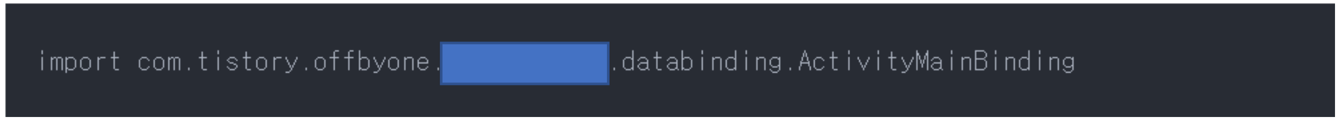
XML의 view 요소에 접근하는 객체를 반환받아view에 접근하는 방식이다.
단순히 findViewById를 대체하기 위한 방법으로만 사용된다.

viewBinding



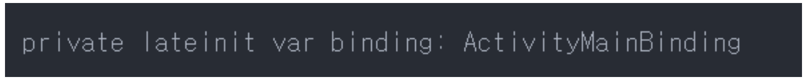
```
buildFeatures {
    viewBinding = true
}
```

Gradle에 의존성 추가



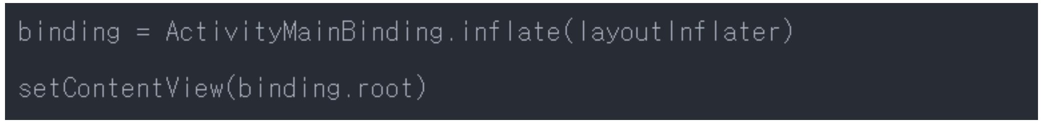
```
import com.tistory.offbyone.[redacted].databinding.ActivityMainBinding
```

Import 해주기



```
private lateinit var binding: ActivityMainBinding
```

ActivityMainBinding형의 멤버 변수 선언!

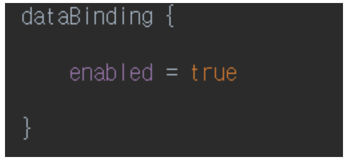


```
binding = ActivityMainBinding.inflate(layoutInflater)
setContentView(binding.root)
```

객체화 시킨후
리소스 연결해버림

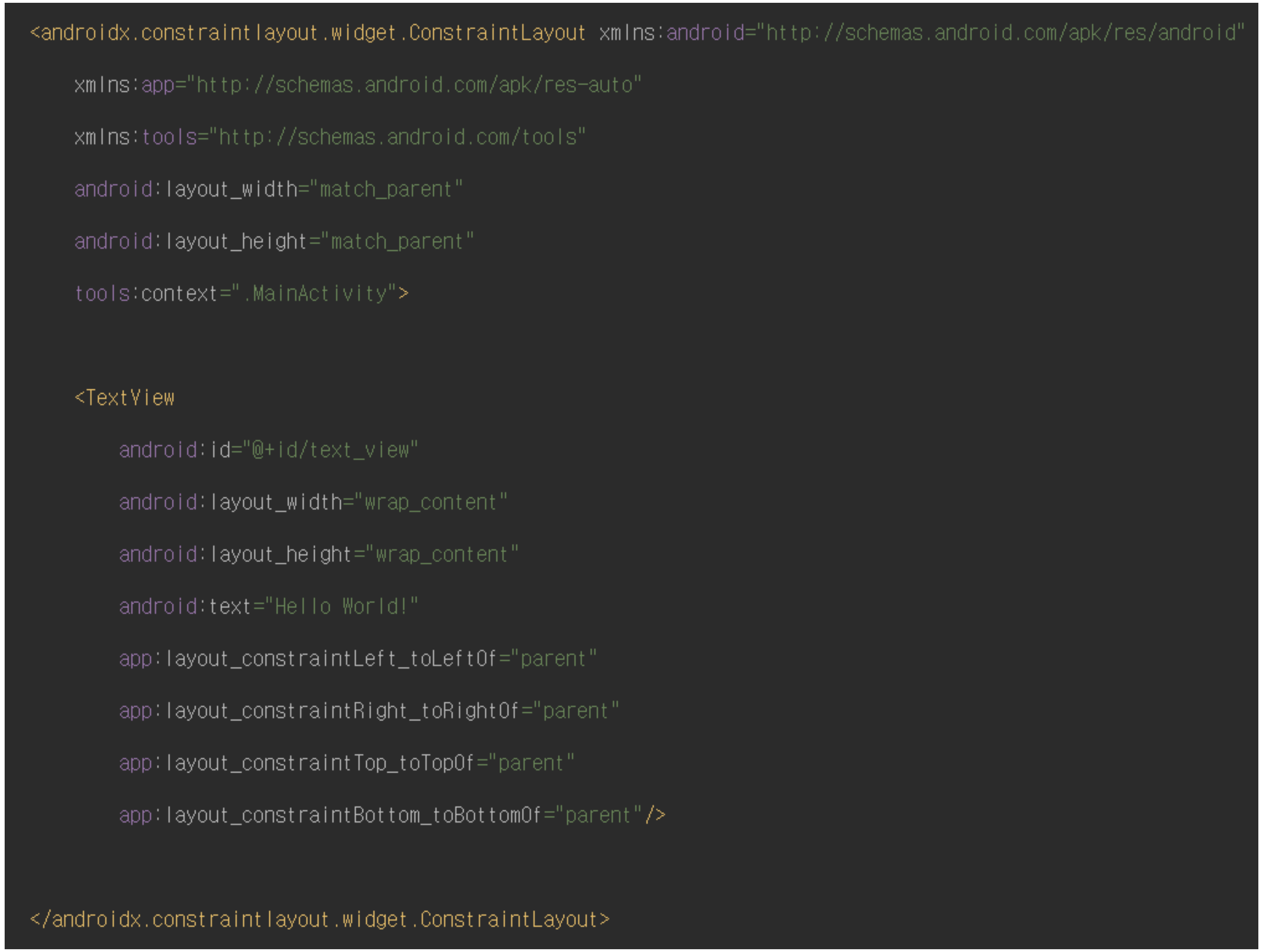
```
binding.helloText.text = "Hello, World!"
binding.textPanel.msgText.text = "Message"
```

dataBinding

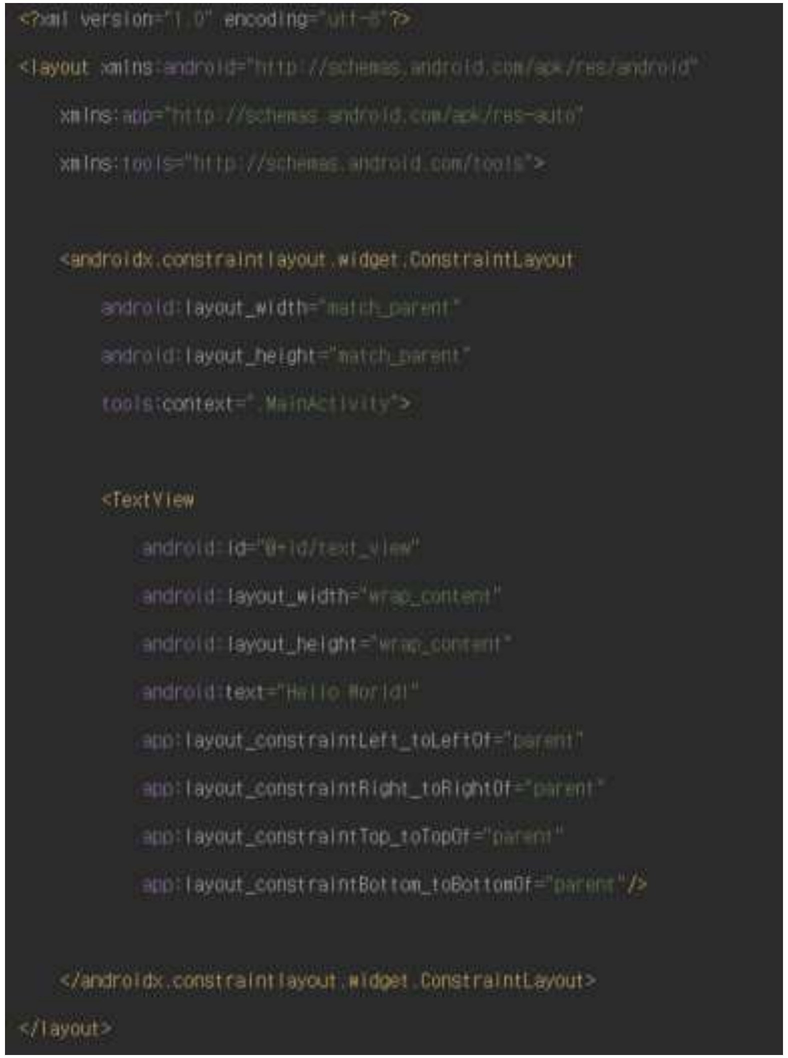


```
dataBinding {
    enabled = true
}
```

의존성 추가

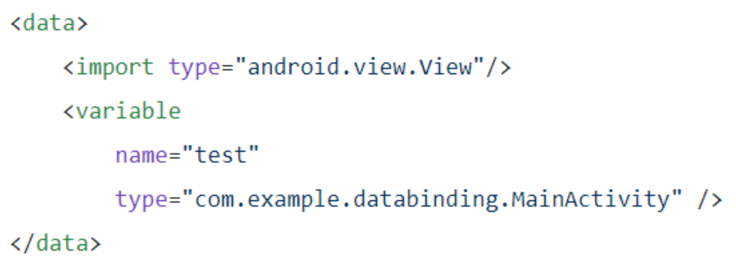


```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"/>

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

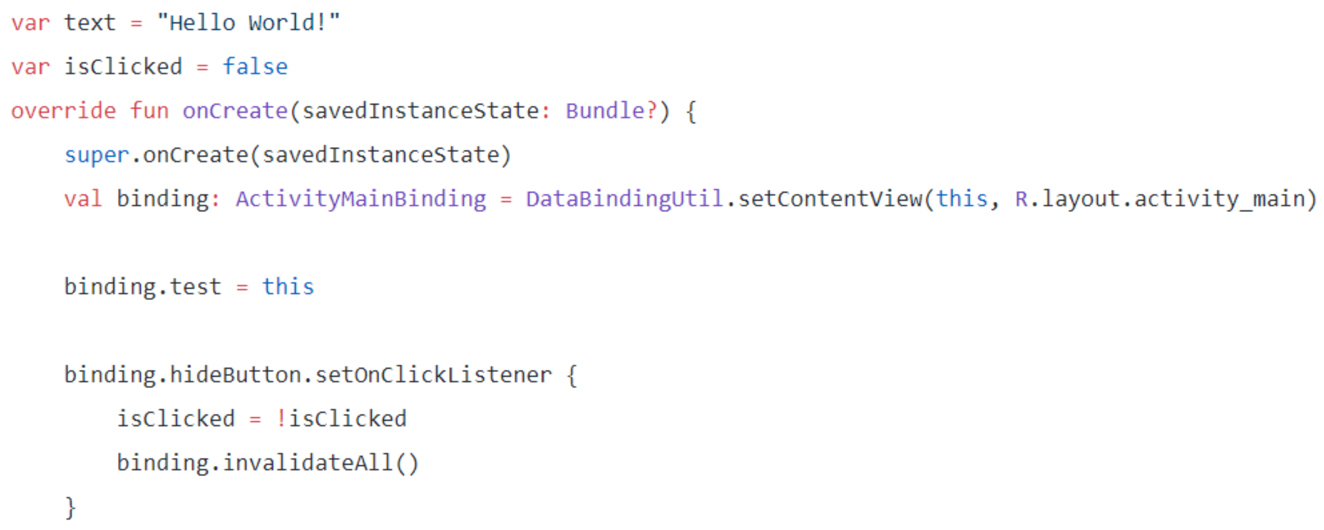


```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

        <TextView
            android:id="@+id/text_view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello World!"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"/>

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

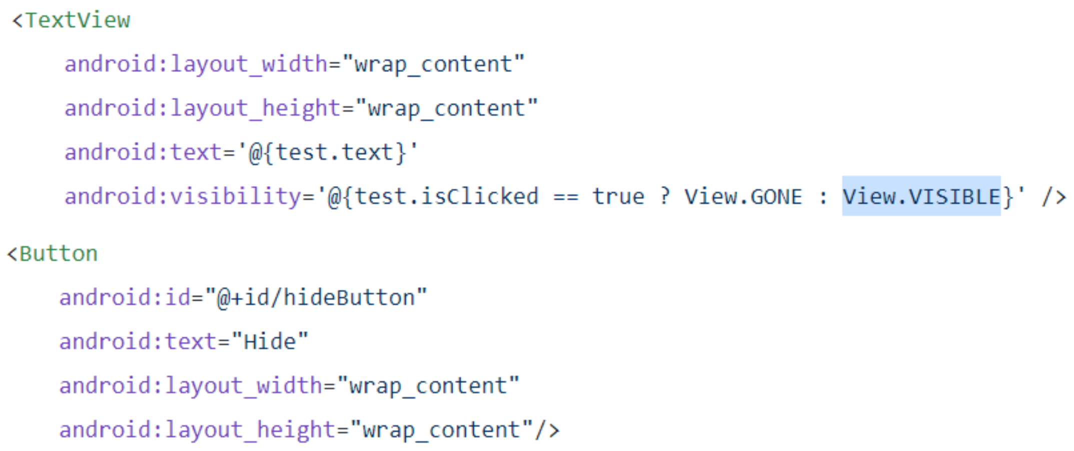
dataBinding



```xml
<data>
    <import type="android.view.View"/>
    <variable
        name="test"
        type="com.example.databinding.MainActivity" />
</data>
```



```kotlin
var text = "Hello World!"
var isClicked = false
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    val binding: ActivityMainBinding = DataBindingUtil.setContentView(this, R.layout.activity_main)

    binding.test = this

    binding.hideButton.setOnClickListener {
        isClicked = !isClicked
        binding.invalidateAll()
    }
```



```xml
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text='@{test.text}'
    android:visibility='@{test.isClicked == true ? View.GONE : View.VISIBLE}' />
<Button
    android:id="@+id/hideButton"
    android:text="Hide"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>
```

databinding
binding