



[스파르타코딩클럽] 안드로이드 앱개발 기초반 - 2주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

[수업 목표]

- 레이아웃을 xml 코드를 이용해 다뤄본다.
 - xml 코드 구성을 파악한다.
 - Attributes 사용법을 익힌다.
 - 복잡한 레이아웃을 구성한다.
- 새로운 액티비티를 추가해본다.
- 성격 유형 조회 화면에 인터랙션을 추가한다.

[목차]

- 01. 2주차 오늘 배울 것
- 02. xml 기초
- 03. XML로 레이아웃 다뤄보기
- 04. Quiz_로그인 액티비티를 발전 시켜보세요!
- 05. 자주 쓰이는 LayoutView 연습하기
- 06. Quiz_성격 유형 조회 화면 뼈대잡기!
- 07. Quiz_성격 유형 조회 화면 완성하기!
- 08. 새로운 화면 추가
- 09. 레이아웃에 생명 불어넣기
- 10. 2주차 끝 & 숙제 설명
- HW. 2주차 숙제 답안



모든 토글을 열고 닫는 단축키

Windows : **Ctrl** + **alt** + **t**

Mac : **⌘** + **⌥** + **t**

01. 2주차 오늘 배울 것

- ▼ 1) 레이아웃 심화, 액티비티

- ☞ 1주차 때 배운 레이아웃을 코드로 작성하는 법을 배워볼게요. 코드로 다루면 더 복잡한 레이아웃을 간단하게 다룰 수 있게됩니다.
한 화면에서만 레이아웃을 구성해서 아쉬웠다고요? 새로운 액티비티를 프로젝트에 추가해볼게요!
새로운 액티비티가 생겼으니 화면에 생명을 불어넣어볼게요.
코틀린 언어를 살짝 맛보게 됩니다.

02. xml 기초

▼ 2) xml 기초

- xml 은 어떻게 사용하나요?

- ☞ 레이아웃 에디터에서 'Code' 혹은 'Split'을 클릭하면 나오는 코드를 보면 아래와 같이 구성되어 있을거예요.

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

xml은 크게 element와 attribute로 구성돼요. 위 코드에서 TextView element이며, attribute로 android:layout_width, android:layout_height, android:text 3가지를 갖고 있네요.

- 1주차때 레이아웃 에디터를 이용해 추가했던 뷰들 기억나시나요? 모두 xml을 이용해 표현가능하답니다.
지난주에 사용했던 뷰들을 코드로 알려드릴게요.
외울필요 없이 검색해서 쓰시면 됩니다!

- ☞

```
<TextView .../>  
<EditText .../>  
<Button .../>  
<ImageView .../>  
<Switch .../>  
<SeekBar .../>
```

- Attributes는 단어가대로 Element가 어떤 특성을 갖는지 표시하는 값이에요.
- 모든 Attributes를 다 알 수는 없겠죠~~ 오늘 쓰는 것만 알아도 굿!
나머지는 검색해서 쓰시면 된답니다~!

👉 크기
android:layout_width
android:layout_height

간격
android:layout_margin
android:padding

글자
android:text
android:textColor
android:textSize

배경관련
android:background

정렬
android:gravity

👉 이 외에도 아주 많으며, 개발자들도 모두 외우고 있지 않습니다. 필요할 때마다 검색해서 사용하면 돼요!

03. XML로 레이아웃 다뤄보기

👉 'Practice' 이름의 프로젝트를 생성해주세요. 그리고 activity_main.xml을 열어주세요.
벌써 잊어버리셨나요? 괜찮아요! 다시 연습하면 됩니다.
File > New > New Project 를 눌러주세요. Empty Activity를 선택하고 프로젝트 이름을 바꾸고 폴더를 설정한 뒤 언어는 Kotlin 인지 확인하고 Finish!

▼ 3) xml로 Attributes 다뤄보기

👉 레이아웃 에디터만 보이고 코드가 안보이신다고요?
오른쪽 위에 있는 'Code', 'Split', 'Design' 버튼 중에서 'Split'을 눌러주세요.

👉 [연습할 것들]
text, width, layout_margin, background, textColor, padding, textSize, gravity, id

👉 **잠깐! 자꾸 오타를 입력하시나요?**
안드로이드 스튜디오에서 자동완성을 제공해줘요. 코드를 입력하고 있으면 아래에 추천리스트를 보여준답니다. 만약 자동완성 기능이 나타나지 않는다면 단축키 **ctrl+space** 를 사용해 보세요.

☞ id는 view를 가르키는 이름을 지정하는 것이라고 생각하면 돼요. 작명이 필요한 시간입니다.
일반적으로 camelCase와 snake_case를 많이 사용해요.
camelCase는 첫 단어는 소문자며 띄어쓰기 단위로 대문자로 구성되어 낙타처럼 생겼다해서 camelCase라고 불러요.
snake_case는 감이오시나요?
snake_case는 모두 소문자로 구성하고 띄어쓰기는 _로 연결하여 뱀처럼 생겼다해서 snake_case라고 불러요.

☞ 예전에는 리소스와 연관된 것들은 snake_case를 사용했다보니 많은 프로젝트들이 snake_case로 구성되어 있습니다.
리소스 중 하나인 이미지와 레이아웃의 파일의 이름형태 기억하시나요?
맞아요. ic_launcher, activity_main 이었죠. snake_case 로 구성되어있었습니다.

그럼 id는 snake_case 사용하면 되나요?
글쎄요. 이건 여러분이 정해야해요. 작명은 정해진 답이 없어요. 개인마다 회사마다 정하기 나름이라서 편하신 걸 사용하면 돼요.

☞ 그래도 결정못하시겠나요?
그러면 수업에서는 id는 camelCase, 그외의 리소스 파일들은 snake_case를 사용하도록 할게요.

왜 snake_case로 통일하지 않고 id만 camelCase를 사용하나요?
최근 안드로이드 스튜디오에서도 자동으로 id를 만들어줄 때 camelCase 형태로 만들어 주고 있어요. 구글이 답은 아니지만 구글에서 가이드하는 트렌드가 layout.xml 안에서의 id 값은 camelCase를 사용하기 시작했어요.

▼ [코드스니펫] - xml로 attributes 다뤄보기(완성)

```
<TextView
    android:id="@+id/easyXml"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="XML 참 쉽죠?"
    android:padding="32dp"
    android:layout_margin="60dp"
    android:background="#0000"
    android:textColor="#ffffff"
    android:textSize="32sp"
    android:gravity="center"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

▼ 4) xml로 레이아웃 구성하기

☞ **잠깐! 정렬의 중요성**
코드의 정렬이 제대로 되어있지 않으면, 코드의 생김새를 파악할 수 없어 오류를 해결하기가 무척 어려워집니다. 안드로이드 스튜디오에서 **ctrl+alt+L** (맥은 cmd+alt+L) 로 자동정렬 기능을 사용해보세요.



[연습할 것들]

Button, ImageView

▼ [코드스니펫] - 르탄이 이미지 파일

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/12d60a1e-962a-4556-afc2-01aa471456ed/rtanny.png>



app:layout_constraintTop_toTopOf 이런 것들이 뭐냐구요?

레이아웃 에디터에서 흰색 원으로 화면에 연결하는 작업을 했었죠? 그 역할입니다.

자세한 내용은 조금 있다가 LayoutView를 다룰때 설명해드릴게요. 우선 모두 parent로 설정을 해주세요.

▼ [코드스니펫] - xml로 레이아웃 구성하기 다뤄보기(완성)

```

<?xml version="1.0" encoding="utf-8"?>
<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/easyXml"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="60dp"
        android:background="#000"
        android:gravity="center"
        android:padding="32dp"
        android:text="XML 참 쉽죠?"
        android:textColor="#ffffff"
        android:textSize="32sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="더 쉽죠?"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/rtanny"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

04. Quiz_로그인 액티비티를 발전 시켜보세요!

▼ 5) 📌 로그인 액티비티를 발전 시켜보세요!

▼ [코드스니펫] - 로그인 프로젝트 (심화) 시작 템플릿

```
<?xml version="1.0" encoding="utf-8"?>
<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/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="로그인 페이지"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editTextTextEmailAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="hint"
        android:inputType="textEmailAddress"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView" />

    <EditText
        android:id="@+id/editTextTextPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="hint"
        android:inputType="textPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextEmailAddress" />

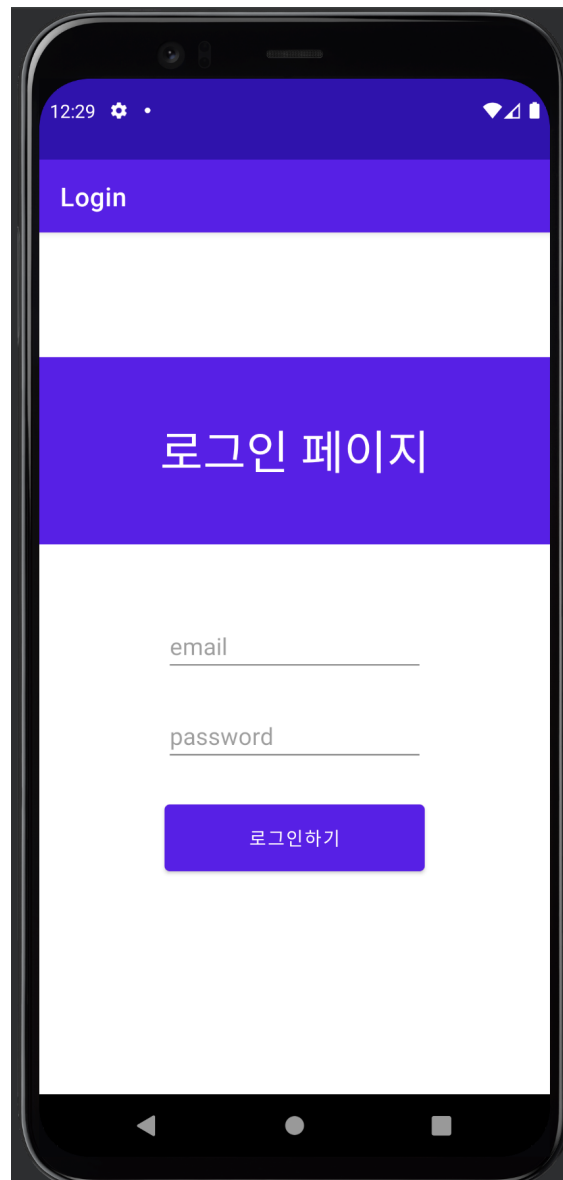
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPassword" />

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

👉 'Practice'의 activity_main.xml를 로그인 프로젝트 (심화) 시작 템플릿 코드로 바꿔서 시작해주세요.

▼ Q. 퀴즈설명

▼ 모습 보기



👉 레이아웃 에디터보다는 xml을 이용해서 수정해주세요!

'로그인 페이지' 글자색상은 #000, 배경색상은 #6200EE 예요.
width, height, margin, padding 값은 정확히 같지 않아도 괜찮습니다. 비슷하게만 보여지면 돼요.



힌트:

- (1) width, height를 체크해주세요.
- (2) margin, padding을 체크해주세요.
- (3) 색상 설정을 체크해주세요.
- (4) 글자 크기를 체크해주세요.
- (5) EditText에서 어떤 것을 수정하면 미리보기 값이 변경 된다고 했죠?

▼ A. 함께하기(완성본)



어때요, 할만했나요? 다만 조금씩 다른 방법으로 해결하셨더라도,
다음 강의 진행을 위해 아래 코드를 복사→붙여넣기 해주세요!

▼ [코드스니펫] - 로그인 프로젝트 (심화) - 완성본

```
<?xml version="1.0" encoding="utf-8"?>
<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/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="96dp"
        android:background="#6200EE"
        android:gravity="center"
        android:padding="48dp"
        android:text="로그인 페이지"
        android:textColor="fff"
        android:textSize="36sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editTextTextEmailAddress"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="56dp"
        android:hint="email"
        android:inputType="textEmailAddress"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView" />

    <EditText
        android:id="@+id/editTextTextPassword"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:hint="password"
        android:inputType="textPassword"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextEmailAddress" />

    <Button
        android:id="@+id/button"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:padding="16dp"
        android:text="로그인하기"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPassword" />

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

05. 자주 쓰이는 LayoutView 연습하기

▼ 6) 자주 쓰이는 ConstraintLayout 연습하기



[연습할 것들]

app:layout_constraintEnd_toEndOf
 app:layout_constraintStart_toStartOf
 app:layout_constraintBottom_toBottomOf
 app:layout_constraintBottom_toTopOf

app:layout_constraintTop_toBottomOf
 app:layout_constraintBottom_toBottomOf

▼ [코드스니펫] - 자주 쓰이는 ConstraintLayout - 완성본

```
<?xml version="1.0" encoding="utf-8"?>
<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/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="96dp"
        android:background="#6200EE"
        android:gravity="center"
        android:padding="48dp"
        android:text="로그인 페이지"
        android:textColor="@color/white"
        android:textSize="36sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editTextTextEmailAddress"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="56dp"
        android:hint="email"
        android:inputType="textEmailAddress"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/textView" />

    <EditText
        android:id="@+id/editTextTextPassword"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:hint="password"
        android:inputType="textPassword"
        app:layout_constraintEnd_toEndOf="@id/editTextTextEmailAddress"
        app:layout_constraintStart_toStartOf="@id/editTextTextEmailAddress"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextEmailAddress" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:padding="16dp"
        android:text="로그인하기"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextTextPassword"
        app:layout_constraintBottom_toBottomOf="parent" />

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

▼ 7) 자주 쓰이는 LinearLayout, ScrollView 연습하기

[연습할 것들]

LinearLayout, ScrollView

▼ [코드스니펫] - 자주 쓰이는 LayoutView - 완성본

[illegible]

```

        android:layout_height="wrap_content"
        android:text="3"
        android:textSize="32sp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="4"
            android:textSize="32sp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="5"
            android:textSize="32sp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="6"
            android:textSize="32sp" />

    </LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

06. Quiz_성격 유형 조회 화면 뼈대잡기!

▼ 8) 📌 성격 유형 조회 화면 뼈대잡기

👉 프로젝트를 'PersonalityTest'로 해서 프로젝트를 생성해주세요.

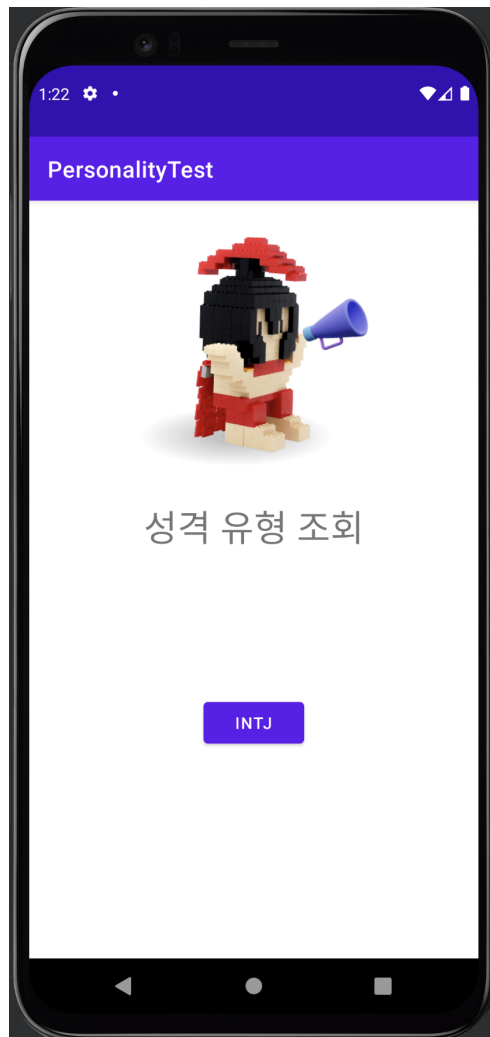
1. File > New > New Project
2. Empty Activity
3. 프로젝트 이름 'PersonlaityTest' 설정
4. 폴더 경로 확인 및 개발언어 Kotlin 확인
5. Finish!

▼ [코드스니펫] - 르탄이 이미지 파일

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/12d60a1e-962a-4556-afc2-01aa471456ed/rtanny.png>

▼ Q. 퀴즈설명

▼ 모습 보기



💡 힌트: 위의 xml 기초에서 배운 내용을 활용해 적절히 수정해보세요!

▼ A. 함께하기(완성본)

💡 어때요, 할만했나요? 다만 조금씩 다른 방법으로 해결하셨더라도, 다음 강의 진행을 위해 아래 코드를 복사→붙여넣기 해주세요!

▼ [코드스니펫] - 성격 유형 조회 화면 뼈대잡기(완성)

```
<?xml version="1.0" encoding="utf-8"?>
<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">

    <ImageView
        android:id="@+id/introImageView"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_marginTop="32dp"
        android:src="@drawable/rtanny"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

```

        app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/introTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:layout_marginTop="32dp"
    android:text="성격 유형 조회"
    android:textSize="32sp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/introImageView" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="INTJ"
    android:layout_marginBottom="56dp"
    app:layout_constraintTop_toBottomOf="@id/introTextView"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

07. Quiz_성격 유형 조회 화면 완성하기!

▼ 9) 🛠️ 성격 유형 조회 버튼 넣어보기!

▼ [코드스니펫] - 성격 유형 조회 화면 뼈대

```

<?xml version="1.0" encoding="utf-8"?>
<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">

    <ImageView
        android:id="@+id/introImageView"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_marginTop="32dp"
        android:src="@drawable/rtanny"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/introTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:gravity="center"
        android:text="성격 유형 조회"
        android:textSize="32sp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@id/introImageView" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="ISTJ" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"

```

```

        android:layout_margin="4dp"
        android:text="ISFJ" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="INFJ" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="INTJ" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ISTP" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ISFP" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="INFP" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="INTP" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ESTP" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ESFP" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ENFP" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ENTP" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ESTJ" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ESFJ" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"

```

```

        android:text="ENFJ" />

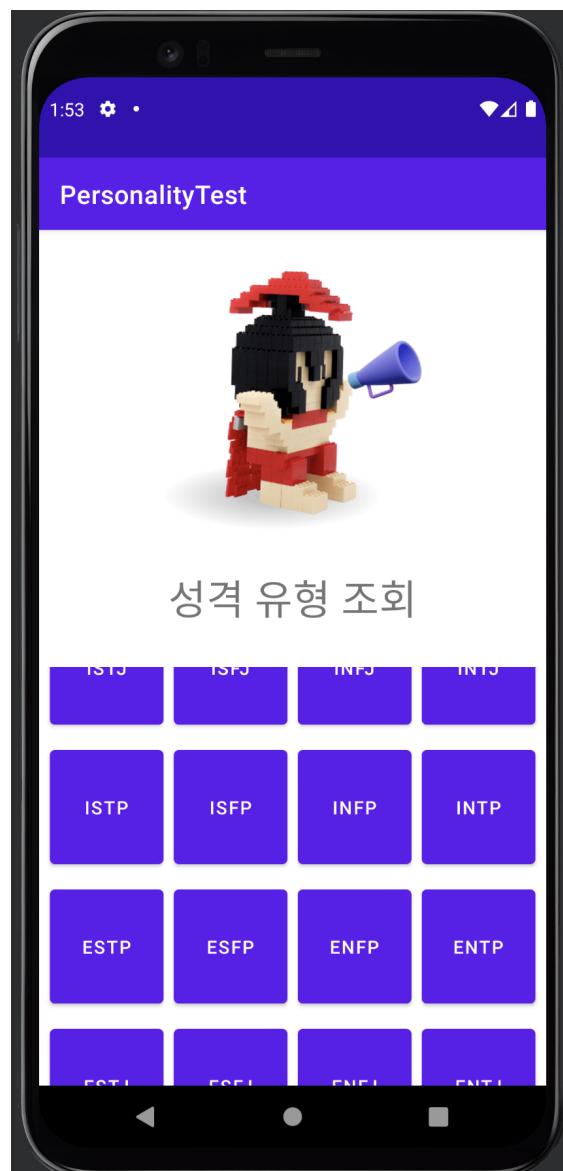
        <Button
            android:layout_width="wrap_content"
            android:layout_height="100dp"
            android:layout_margin="4dp"
            android:text="ENTJ" />

    </androidx.constraintlayout.widget.ConstraintLayout>

```

▼ Q. 퀴즈설명

▼ 모습 보기





힌트:

ScrollView를 Layout처럼 일부만 적용할 수 있습니다.

LinearLayout의 android:orientation을 모두 활용해보세요.

코드를 활용하면 반복하기 좋아요!

▼ A. 함께하기(완성본)



어때요, 할만했나요? 다만 조금씩 다른 방법으로 해결하셨더라도,
다음 강의 진행을 위해 아래 코드를 복사→붙여넣기 해주세요!

▼ [코드스니펫] - activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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">

    <ImageView
        android:id="@+id/introImageView"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_marginTop="32dp"
        android:src="@drawable/rtanny"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/introTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:gravity="center"
        android:text="성격 유형 조회"
        android:textSize="32sp"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@id/introImageView" />

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginTop="32dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/introTextView">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:orientation="horizontal">

                <Button
                    android:layout_width="wrap_content"
                    android:layout_height="100dp"
```



```

        android:layout_margin="4dp"
        android:text="ISTJ" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="ISFJ" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="INFJ" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="100dp"
    android:layout_margin="4dp"
    android:text="INTJ" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="ISTP" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="ISFP" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="INFP" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="INTP" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="ESTP" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="ESFP" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="ENFP" />

    <Button

```

```

        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_margin="4dp"
        android:text="ENTP" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:orientation="horizontal">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="100dp"
            android:layout_margin="4dp"
            android:text="ESTJ" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="100dp"
            android:layout_margin="4dp"
            android:text="ESFJ" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="100dp"
            android:layout_margin="4dp"
            android:text="ENFJ" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="100dp"
            android:layout_margin="4dp"
            android:text="ENTJ" />

    </LinearLayout>
</LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

08. 새로운 화면 추가

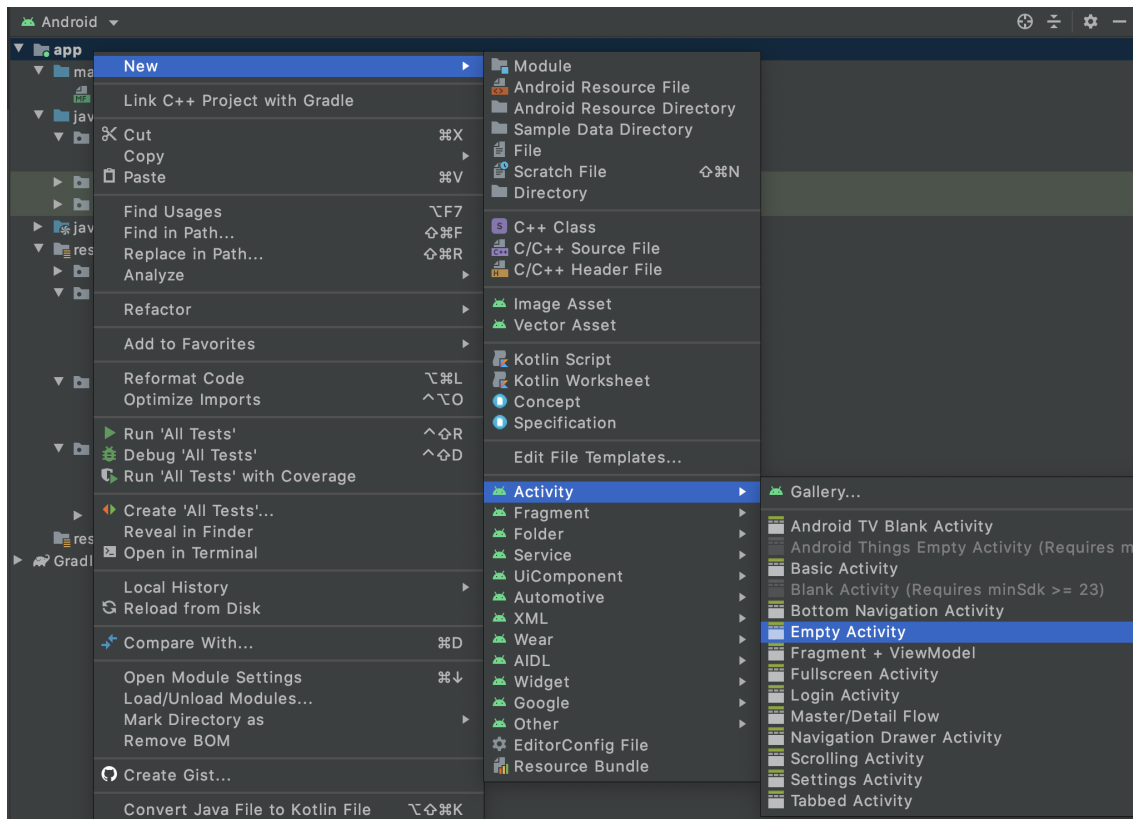
▼ 10) 성격 유형 상세정보 액티비티 생성

👉 화면 연습할때마다 프로젝트를 만드셔서 번거로우셨다고요?
다 연습을 위한 과정이었답니다!
이번에는 프로젝트에 액티비티를 추가해볼게요.

(1) Project 창에서 app 폴더를 마우스 오른쪽 버튼으로 클릭합니다.

(2) New > Activity > Empty Activity 를 선택합니다.

👉 프로젝트 처음 생성할 때 항상 Empty Activity를 선택한거 기억하시나요?
우리는 이제 화면을 직접 구성할 수 있기에 빈 액티비티를 활용해서 채워나갈거예요.



👉 Blank Activity는 왜 선택할 수 없나요?

minSdk 때문이에요!

(만약 선택가능하시다면 현재 프로젝트는 23보다 높은 버전을 선택하신거예요.)

1주차 프로젝트 생성할 때 minSdk가 낮으면 많은 버전을 지원하지만 최신 기술은 사용할 수 없다는 트레이드 오프가 있단 것 기억하시나요?

Blank Activity를 쓰고 싶다면 21, 22 버전을 위해 우회하는 작업이 필요하답니다.

(3) Configure Activity 화면에서 Activity Name에 'DetailActivity'를 입력해주세요.

👉 Layout Name이 자동으로 activity_detail로 바뀐 것이 보이나요?

안드로이드 작명 규칙은 액티비티는 '[이름]Activity' 형태이며 해당 액티비티에서 쓰는 레이아웃 파일의 이름은 'activity_[이름]'을 사용해요.

(4) Finish를 누르고 기다리면 끝!

▼ 11) 생성한 액티비티 확인하기

- app > java > com.example.personalitytest 에 'DetailActivity'가 생성되었네요.
- app > res > layout 에 'activity_detail.xml' 이 추가되어 있네요.



앗, 뭔가 익숙하지 않나요?

맞아요! 1주차에서 배웠던 프로젝트 기본 구성요소들이예요.

프로젝트 기본 구성요소에는 앱 컴포넌트, 앱 리소스 그리고 매니페스트 파일이 있단 것 기억하시나요?
액티비티는 앱 컴포넌트, 레이아웃 xml은 앱 리소스라고 했었죠.
그럼, 매니페스트 파일도 변화가 있는지 확인해볼까요?

- app > manifests > AndroidManifest.xml 파일을 열어볼게요.



AndroidManifest도 xml 파일입니다. 그래서 레이아웃때 작성했던 코드와 유사한 형태를 띄고 있어요.
매니페스트 파일도 element 와 attribute로 구성되어있네요.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.personalitytest">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.PersonalityTest">
        <activity android:name=".DetailActivity"></activity>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

- application 안에 activity가 2개 있는데 'MainActivity'도 보이고 방금 추가한 'DetailActivity'도 들어있네요!
- 매니페스트 파일에는 앱에서 사용하는 모든 화면 정보 즉, 사용하는 액티비티들이 모두 기록되어야합니다. 매니페스트 파일에 기록되지 않은 액티비티를 앱에서 사용하려면 버그가 발생해요.



MainActivity는 왜 DetailActivity와 다르게 왜 복잡하냐구요?

눈썰미가 좋으시네요!

사용자가 앱 아이콘을 클릭하면 안드로이드 시스템은 매니페스트 파일을 확인해서 앱을 동작한다고 한 것 기억하시나요?

매니페스트 파일이 '우리 앱은 MainActivity가 첫 화면이야'라고 알려주는 정보예요.



만약 DetailActivity를 시작하는 화면으로 바꾸고 싶으시다면,

<intent-filter> ~ </intent-filter> 의 내용을 그대로 DetailActivity 안으로 넣으시면 돼요.

우리는 MainActivity가 시작화면이기에 바꾸지 않을거예요.



위의 방식으로 액티비티를 만들면 안드로이드 스튜디오는 세가지 작업을 자동으로 해줘요.

1. 'DetailActivity' 파일을 만들고,
2. 'DetailActivity'에서 사용할 레이아웃 파일인 'activity_detail'을 만들어요.
3. 그리고 필수 요소인 AndroidManifest.xml에 'DetailActivity' 정보를 추가한답니다.

09. 레이아웃에 생명 불어넣기

▼ 12) Activity.kt 의 생김새 살펴보기



오늘은 코틀린 문법에 대해서 자세히 다루지 않아요. 살짝 이런 형태를 갖고 있다 정도로만 다뤄볼거예요. 코틀린 문법을 자세히 배우고싶으시다면 다음주 강의를 기대해주세요!

- 우선 MainActivity.kt를 열어 살펴볼까요?

```
package com.example.personalitytest

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

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

- 뭐가 뭔지 모르시겠나요? 그럼 한번 DetailActivity와 비교하면서 봐봐요

```
package com.example.personalitytest

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

class DetailActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail)
    }
}
```

- 차이점이 눈에 들어오시나요?
 - (1) 'class ~: AppCompatActivity() {' 사이에 액티비티 이름이 들어있네요.
 - (2) setContentView(R.layout.~) 안에 사용하는 레이아웃 xml 파일명이 들어있네요.

☞ 차이점 2가지를 제외하고 나머지는 액티비티를 만들면 생성되는 가장 기본적인 틀이에요. 액티비티를 만들면 항상 위의 형태를 지닌다는 것 기억해두시면 좋아요.

살짝만 추가 설명을 해드리면,
package 는 파일이 들어있는 경로를 나타내요.
import 는 사용하려는 다른 파일을 적어둔 것이에요. 안드로이드 스튜디오 기능을 사용하면 필요한 것을 자동으로 추가할 수 있습니다.
override fun onCreate 무슨말인지 모르겠는데, onCreate 뭔가 만드나봐요.

☞ 여러 궁금증이 몰려오시겠지만 자세히 설명하고 완벽히 이해하려면 사실 많은 배경지식이 필요해요. 그러니 우리 게임한다 생각하고 필요한 수준만 학습하고 사용해보면서 하나씩 지식을 쌓아봐요. 마치, 매니페스트 파일이 어떤 역할을 하는지, xml 코드를 몰랐어도 우리 1주차 때 레이아웃 에디터로 앱을 구성했잖아요! 모두 알아두면 좋겠지만 모든 것을 다 미리 알아야할 필요는 없습니다.

▼ 13) MainActivity.kt 버튼 동작 함께 연동하기

☞ PersonalityTest 프로젝트를 이어서 사용할게요.

(1) MainActivity.kt 에 **ManActivity clickIstj** 버튼 연동 코드를 위치에 맞춰서 추가해주세요.

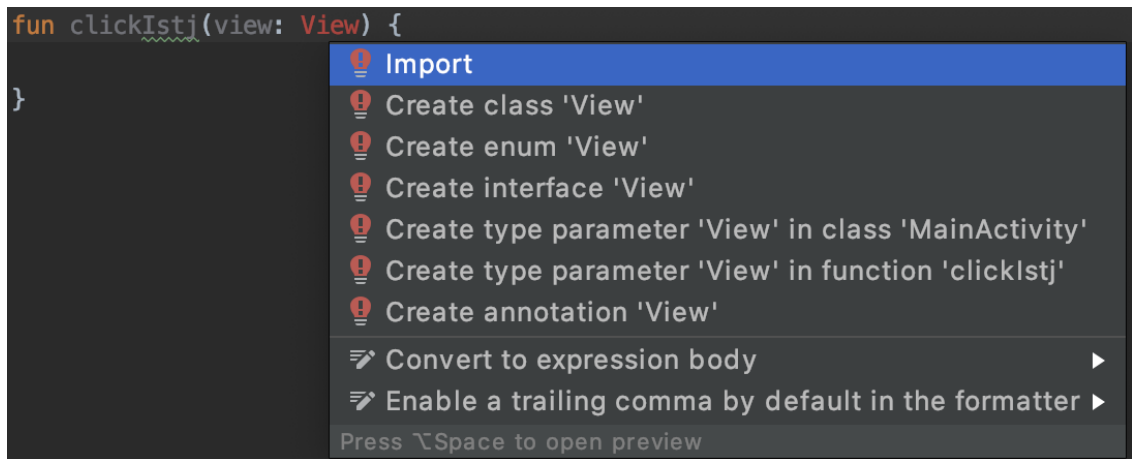
▼ [코드스니펫] - ManActivity - clickIstj 버튼 연동

```
fun clickIstj(view: View) {  
  
}
```

☞ 추가한 clickIstj 앞에 **fun** 보이시나요? function의 약자예요. 주로 '함수'라고 불리며 기능이나 동작을 위해 사용해요.

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
  
    fun clickIstj(view: View) {  
  
    }  
}
```

(2) 그러면 View라는 코드가 빨강게 표시되는게 보이시나요? 마우스커서를 올리고 Alt+Enter(Mac의 경우 Option+Enter)를 눌러주세요. 그러면 Import 라는 단어가 있을 거예요. Import에 커서를 올리고 엔터를 누르거나 마우스로 클릭해주세요.



☞ Alt+Enter(Mac: Option+Enter)는 빠른 수정 실행 단축키입니다. 코틀린 코드에서 빨간색으로 표시되는 Error 코드를 마주하게되면 빠른 수정을 실행해보세요. 안드로이드 스튜디오가 우리에게 해결할 수 있는 방법을 제안해줘요.

(3) 빨간색이 없어졌나요? 그러면 이제 레이아웃으로 이동해볼게요. activity_main.xml을 열어주세요.

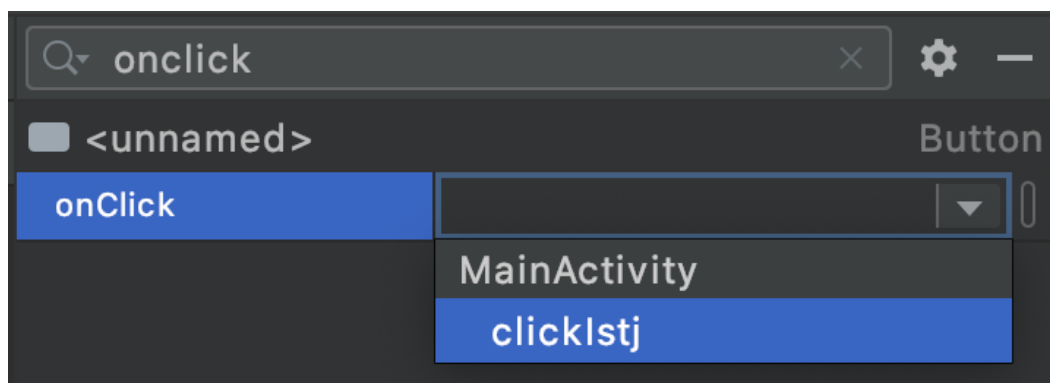
☞ 레이아웃 에디터 사용법도 되살릴 겸 레이아웃 에디터를 이용해 수정할게요. 'Split'으로 설정되어 있다면 오른쪽 위의 'Design'을 눌러 레이아웃 에디터로 변경해주세요.

(4) 레이아웃 에디터에서 ISTJ가 적힌 버튼을 클릭해주세요.

(5) 버튼의 Attributes 에서 onClick을 찾아주세요.

☞ 검색버튼을 이용하면 더 빠르게 찾을 수 있어요.

(6) 드롭다운 항목에서 MainActivity 아래에 있는 'clickIstj'를 선택해주세요.



☞ MainActivity.kt로 이동해 코드를 확인해보면 clickIstj가 회색에서 노란색으로 변경되었네요. 안드로이드 스튜디오에서는 다른 곳에서 사용되면 코드가 색상을 갖게 돼요.

(7) 거의 다왔어요! 버튼이 클릭되면 '청렴결백한 논리주의자'라고 Toast 메시지를 띄울거예요.



토스트 메시지는 동작에 대한 간단한 피드백을 제공하는 작업 팝업이에요. 토스트 메시지는 시간이 지나면 자동으로 사라진다는 특징이 있어요.

(8) clickIstj 함수안에 **ManActivity - Toast 메시지 띄우기** 코드를 붙여넣어주세요.

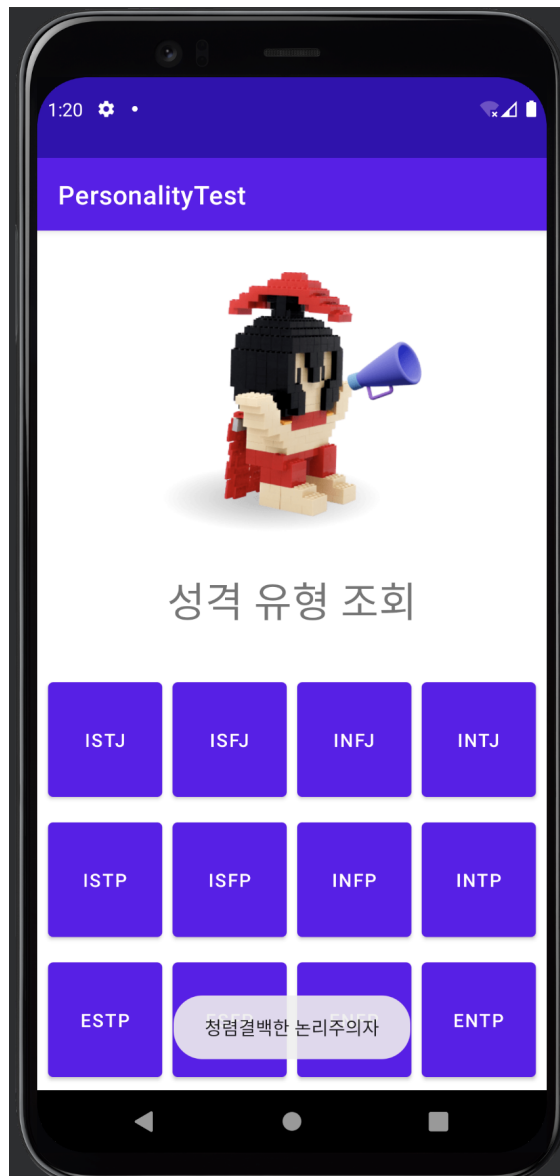
▼ [코드스니펫] - ManActivity - Toast 메시지 띄우기

```
Toast.makeText(view.context, "청렴결백한 논리주의자", Toast.LENGTH_LONG).show()
```

```
fun clickIstj(view: View) {  
    Toast.makeText(view.context, "청렴결백한 논리주의자", Toast.LENGTH_LONG).show()  
}
```

(9) 앗, Toast라는 글자가 빨강다구요? Alt+Enter(Mac의 경우 Option+Enter) 눌러 빠른 수정을 실행해주세요. 그리고 Import를 하시면 완성!

(10) 프로젝트를 실행해주세요. 'Run' 버튼 잊으신거 아니겠죠? 앱이 실행되면 ISTJ 버튼을 눌러보세요! 메시지가 나오네요!



10. 2주차 끝 & 숙제 설명



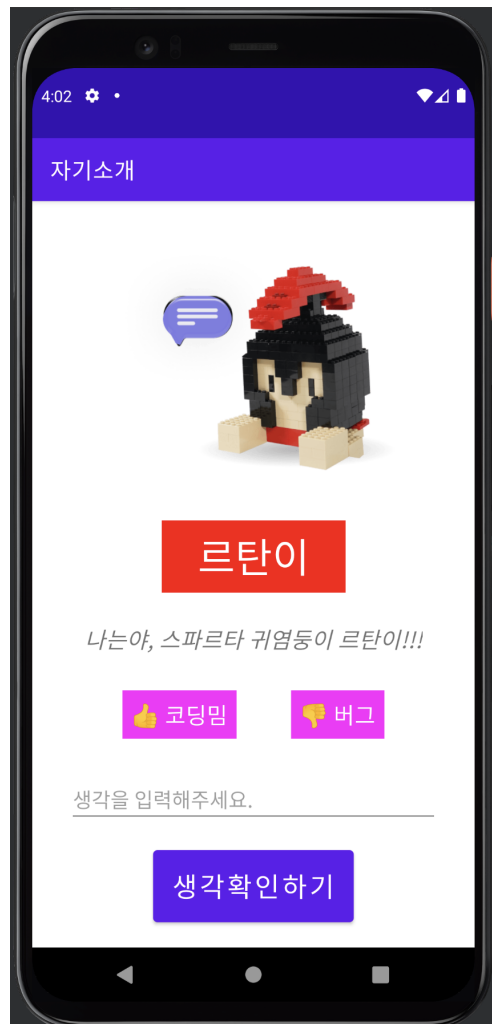
1주차에 이어서 셀프 브랜딩 앱을 만들어볼게요. 아래 기획서를 보고, 자기소개 앱에 정보를 더 추가해 자기소개 앱의 메인 화면을 완성해주세요. (개인정보가 민감하다면, 가상 인물을 소개하는 앱을 만드셔도 좋습니다.)

기능: 스크롤이 가능한 화면으로 구현해주세요. 예시화면보다 더 아름답게 꾸며주시면 좋아요.

▼ 기획서(레이아웃) 보기

자기소개
<div> <div>프로필사진</div> <div>이름</div> <div>나를 한마디 또는 한단어로 정의</div> <div>좋아하는 것 & 싫어하는 것</div> <div>생각입력창</div> <div>생각확인 버튼</div> </div>

▼ 예시 화면



HW. 2주차 숙제 답안

▼ [코드스니펫] activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TextView
                android:id="@+id/textView2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="#E43B3B"
                android:padding="16dp"
                android:text="르탄이"
```

```

        android:textColor="#FFFFFF"
        android:textSize="28sp"
        android:layout_marginTop="32dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/imageView" />

<ImageView
    android:id="@+id/imageView"
    android:layout_width="213dp"
    android:layout_height="217dp"
    android:layout_marginTop="80dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/rtanny" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:text="나는아, 르탄이!!"
    android:textSize="20sp"
    android:textStyle="italic"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />

<TextView
    android:id="@+id/textViewLike"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="#f0f"
    android:padding="4dp"
    android:text="👍코딩맘"
    android:textColor="#fff"
    android:textSize="18sp"
    app:layout_constraintEnd_toStartOf="@id/textViewDislike"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/textView" />

<TextView
    android:id="@+id/textViewDislike"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="#f0f"
    android:padding="4dp"
    android:text="👎버그"
    android:textColor="#fff"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@id/textViewLike"
    app:layout_constraintTop_toBottomOf="@id/textView" />

<EditText
    android:id="@+id/editTextThink"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:hint="생각을 입력해주세요."
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/textViewLike" />

<Button
    android:id="@+id/buttonThink"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="16dp"
    android:layout_margin="32dp"
    android:text="생각확인하기"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"

```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/editTextThink" />

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

Copyright © TeamSparta All rights reserved.