

안드로이드 SDK 개발 플랫폼 활용하기

TEXTVIEW 활용하기

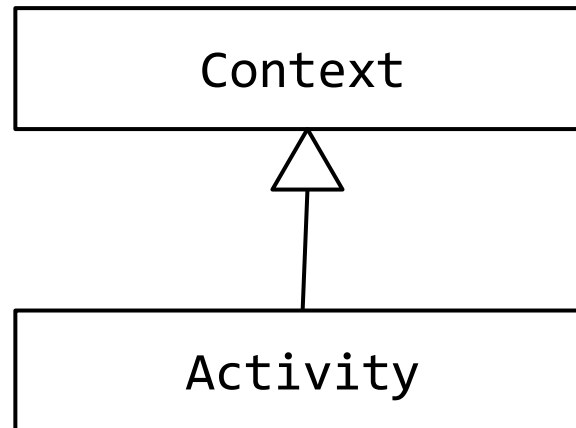
- Project Name
 - 프로젝트를 구분시켜주기 위한 이름
- Create new project in workspace
 - empty project는 생성 안됨
 - 기본 바탕에 기본 문자열을 출력하는 프로그램(Hello, World)코드가 포함되어있는 프로젝트가 생성
 - 프로젝트 생성 후 바로 빌드해서 실행해 볼 수 있음
- Create project from existing source
 - 작업영역(workspace)가 아닌 다른 디렉토리 위치에 있는 안드로이드 프로젝트를 불러와서 작업할 때 사용
- Create project from existing sample
 - 안드로이드 SDK에 포함되어 있는 예제를 불러올 때 사용

- Build Target
 - 타겟보드에 포팅되어 있는 Android SDK 버전에 맞춰서 선택
 - 예를 들면, SDK에 구글지도 API를 사용하는 경우 Google APIs를 선택
- Application Name
 - 단말에서 어플리케이션에 사용되는 이름
 - 단축 아이콘의 레이블, 실행중인프로그램의 제목 표시줄에 나타날 레이블
- Package Name
 - 클래스 이름 충돌 방지용으로 사용
 - APK 파일 배포 시 package name을 사용
 - 안드로이드 프로젝트는 모두 package name이 달라야 함
 - 여기서는 com.iot 패키지 뒤에 프로젝트 이름을 소문자로 붙여서 사용

- Create Activity 화면
 - 기본 생성되는 Activity 클래스에 사용될 이름을 지정
 - Activity는 클래스로부터 상속받아 정의할 subclass 이름을 명시
 - 그러면 ActivityName.java 파일이 생성
- Min SDK Version
 - 안드로이드앱이 작동할 수 있는 최소 버전 지정
 - 만약 Android 1.6을 사용하는 경우 API 레벨4를 입력

- 안드로이드 어플리케이션의 기본단위
 - 보통 사용자가 현재 보고있는 전체화면을 뜻함
 - 화면 구성정보를 가지며 사용자의 입력을 처리
- Activity 클래스로부터 상속받아 사용자정의 subclass를 만들어 사용
 - superclass로부터 필요로 하는 메서드를 오버라이딩해서 사용

- 텍스트뷰는 뷰의 자식
 - 액티비티가 화면이고, 화면은 컨텍스트로 나타냄
 - 화면 위에 뷰를 배치 가능
 - 뷰를 화면에 표시하기 위해서는 컨텍스트 필요
 - 액티비티가 바로 컨텍스트의 일종



- **activity_main.xml 파일을 열고 TextView 추가하기 -1/2**
 - 다시 빌드한 후 AVD에서 텍스트뷰가 표시되는지 확인

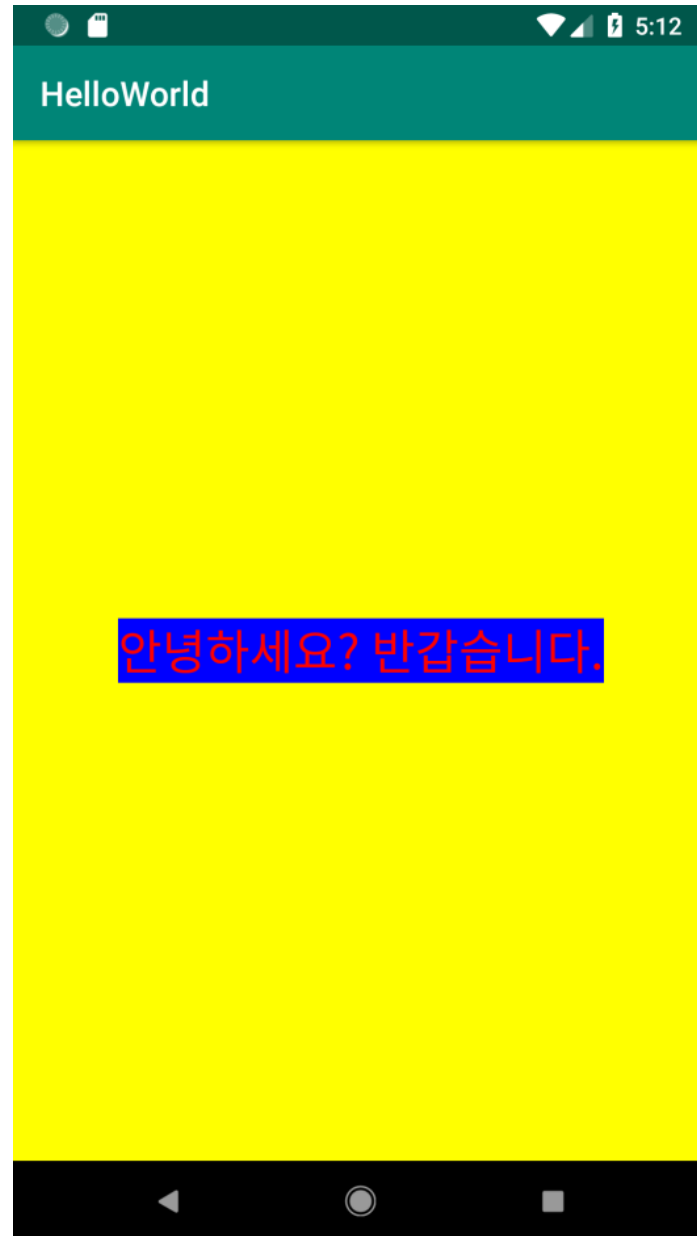
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFF00"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/brother"
        android:layout_width="wrap_content" // 콘텐츠에 맞게 최소화
        android:layout_height="wrap_content" // 콘텐츠에 맞게 최소화
```

- activity_main.xml 파일을 열고 TextView 추가하기-2/2

```
android:text="안녕하세요? 반갑습니다."
android:textSize="28dp"
android:textColor="#FF0000"
android:background="#0000FF"
app:layout_constraintBottom_toBottomOf="parent" // 부모를 기준으로 맞춤
app:layout_constraintLeft_toLeftOf="parent" // 부모를 기준으로 맞춤
app:layout_constraintRight_toRightOf="parent" // 부모를 기준으로 맞춤
app:layout_constraintTop_toTopOf="parent" // 부모를 기준으로 맞춤 />

</android.support.constraint.ConstraintLayout>
```

- 텍스트 뷰를 하나더 추가한 다음 글자의 크기와 색깔 배경 색을 바꿔서 표시

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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" // 부모를 안벗어나게 최대화
    android:layout_height="match_parent" // 부모를 안벗어나게 최대화
    android:background="#FFFF00"
    tools:context=".MainActivity">

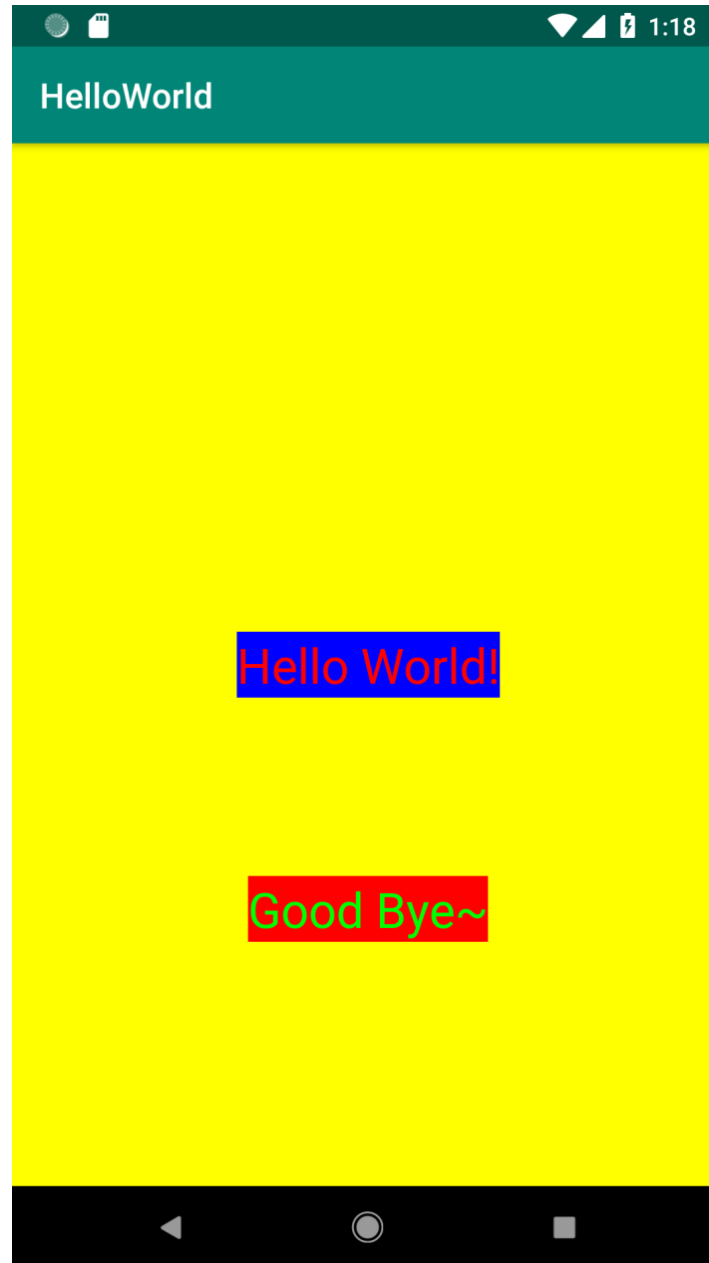
    <TextView
        android:id="@+id/brother"
        android:layout_width="wrap_content" // 컨텐츠에 맞게 최소화
        android:layout_height="wrap_content" // 컨텐츠에 맞게 최소화
        android:text="Hello World!"
        android:textSize="28dp"
        android:textColor="#FF0000"
        android:background="#0000FF"
```

```
app:layout_constraintBottom_toBottomOf="parent" // 부모를 기준으로 맞춤  
app:layout_constraintLeft_toLeftOf="parent" // 부모를 기준으로 맞춤  
app:layout_constraintRight_toRightOf="parent" // 부모를 기준으로 맞춤  
app:layout_constraintTop_toTopOf="parent" // 부모를 기준으로 맞춤 />
```

<TextView

```
android:layout_width="wrap_content" // 콘텐츠에 맞게 최소화  
android:layout_height="wrap_content" // 콘텐츠에 맞게 최소화  
android:text="Good Bye~"  
android:textSize="28dp"  
android:textColor="#00FF00"  
android:background="#FF0000"  
app:layout_constraintBottom_toBottomOf="parent" // 부모를 기준으로 맞춤  
app:layout_constraintLeft_toLeftOf="parent" // 부모를 기준으로 맞춤  
app:layout_constraintRight_toRightOf="parent" // 부모를 기준으로 맞춤  
app:layout_constraintTop_toTopOf="@id/brother"  
/>
```

</android.support.constraint.ConstraintLayout>



- 액티비티의 배경은 핑크색
 - 위쪽 텍스트뷰의 배경은 검은색으로 변경
 - 아래쪽 텍스트뷰의 배경은 흰색으로 변경