

# Android

part 1

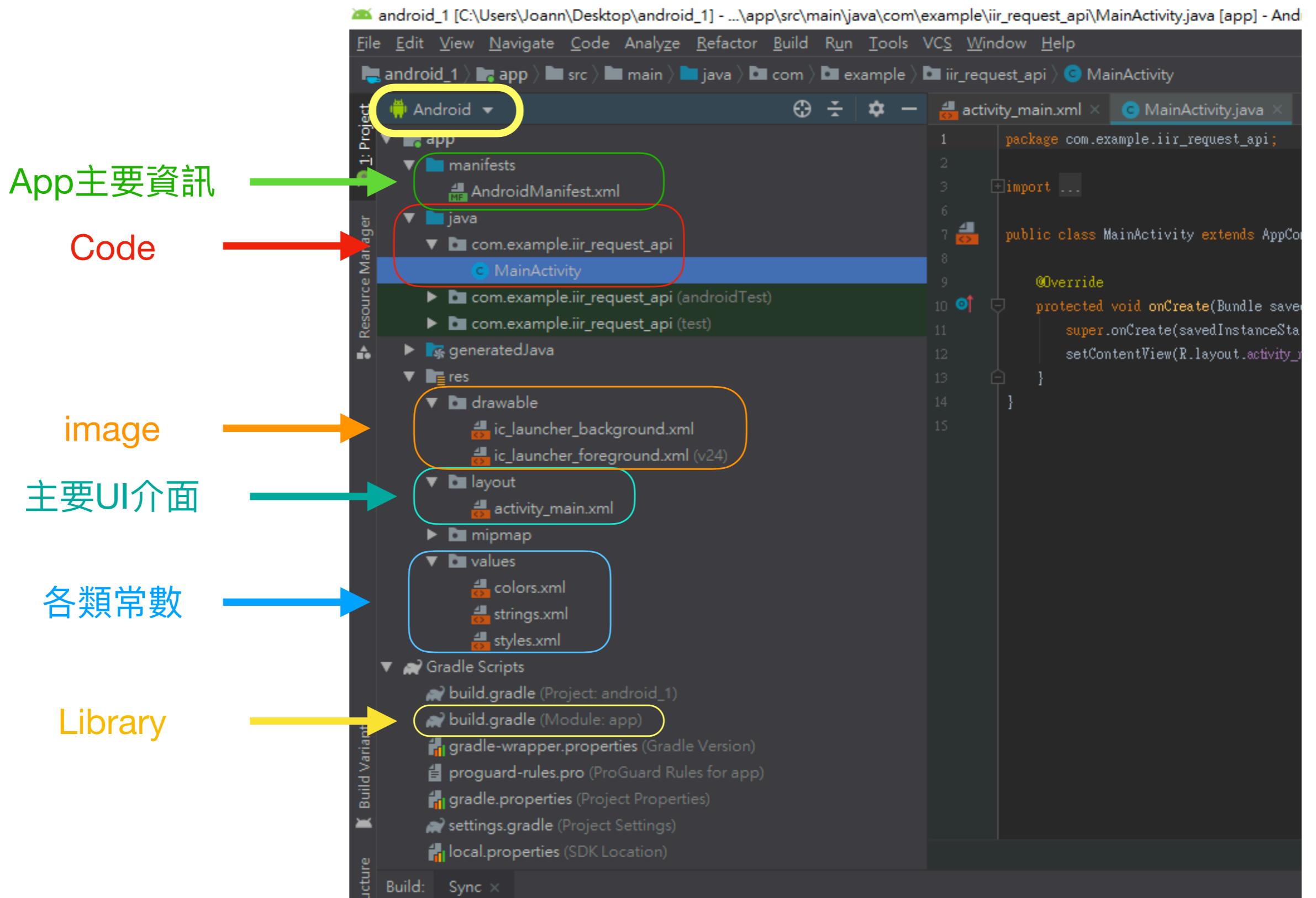
# What is Android?

The screenshot shows the official Android Developers website with the "Android Studio" tab selected in the navigation bar. Below the navigation bar, there are four tabs: "DOWNLOAD", "WHAT'S NEW", "USER GUIDE", and "PREVIEW". The main content area features the "android studio" logo and a brief description: "Android Studio provides the fastest tools for building apps on every type of Android device." It includes two prominent buttons: "DOWNLOAD ANDROID STUDIO" (in green) and "PREVIEW RELEASES" (in white). Below these buttons, download links are provided: "3.4.1 for Mac (1026 MB)", "3.5 Beta 5 (724 MB)", and "3.6 Canary 4 (731 MB)". Further down, there are links for "DOWNLOAD OPTIONS" and "RELEASE NOTES". At the bottom of the screenshot, a portion of the Android Studio IDE interface is visible, showing the project structure and code editor with XML files like "image\_grid.xml" and "AndroidManifest.xml".

# Project

- Github : <https://github.com/ncku-iir/Android-Practice>

# Android Structure



# Android Virtual Device

Virtual Device Configuration X

 System Image  
Android Studio

Select a system image

Recommended   x86 Images   Other Images

Release Name	API Level	ABI	Target
<a href="#">Q Download</a>	29	x86	Android 9.+ (Google APIs)
<a href="#">Pie Download</a>	28	x86	Android 9.0 (Google APIs)
<a href="#">Oreo Download</a>	27	x86	Android 8.1 (Google APIs)
<a href="#">Oreo Download</a>	26	x86	Android 8.0 (Google APIs)
<a href="#">Nougat Download</a>	25	x86	Android 7.1.1 (Google APIs)
<a href="#">Nougat Download</a>	24	x86	Android 7.0 (Google APIs)
<a href="#">Marshmallow Download</a>	23	x86	Android 6.0 (Google APIs)
<a href="#">Lollipop Download</a>	22	x86	Android 5.1 (Google APIs)

**Q** API Level  
**29**  
Android  
**9.+**  
Google Inc.  
System Image  
**x86**

We recommend these images because they run the fastest and support Google APIs.

Questions on API level?  
See the [API level distribution chart](#)

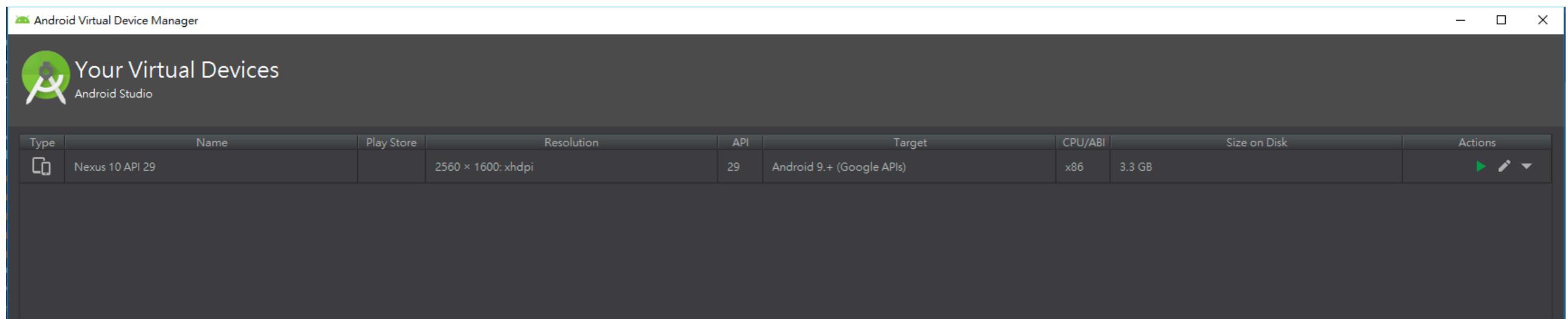
! A system image must be selected to continue.

?

Previous   Next   Cancel   Finish

# Android Virtual Device

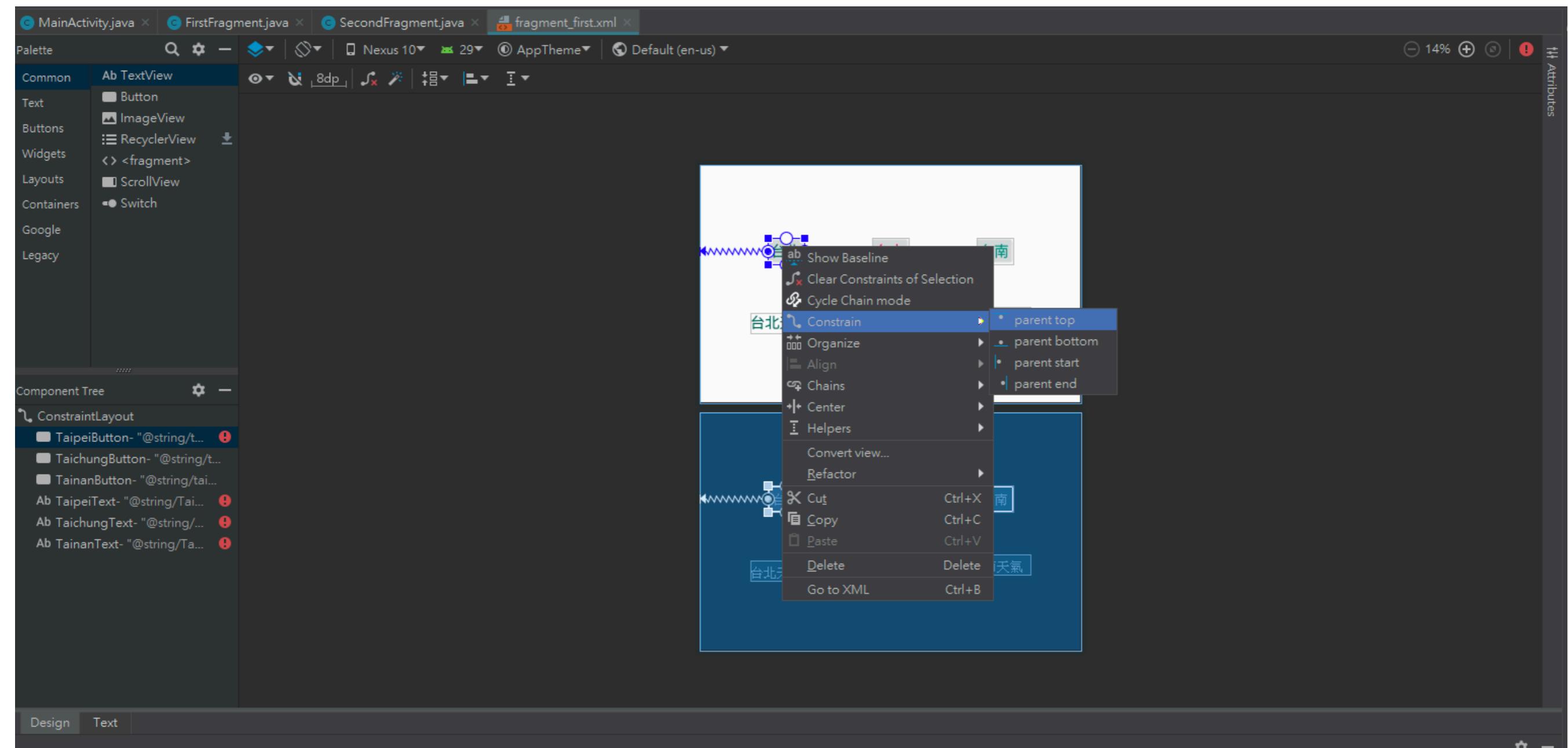
## Nexus 10



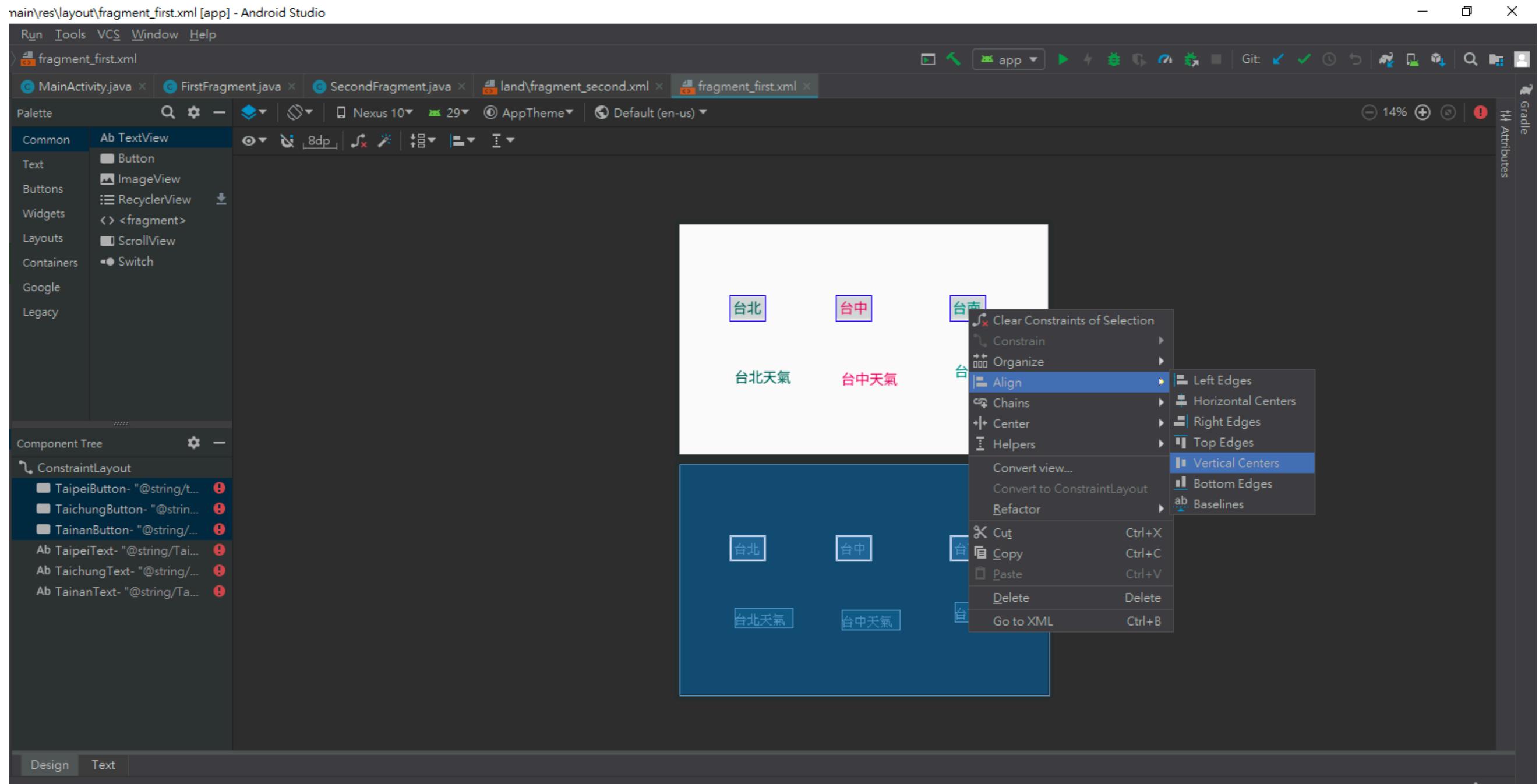
# UI- Layout



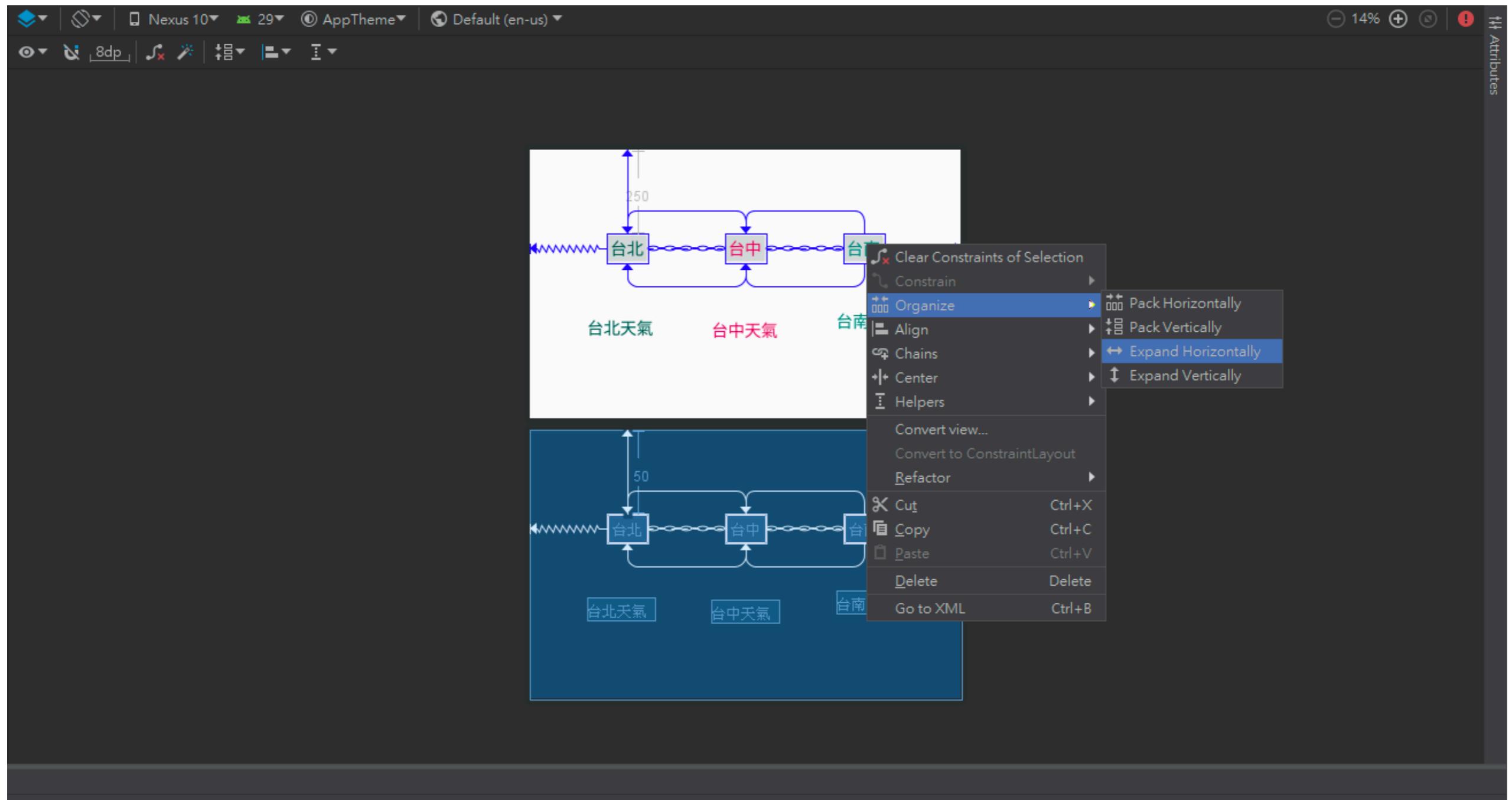
# UI- Constrain



# UI- Align



# UI- Expand



# MainActivity

```
public class MainActivity extends AppCompatActivity {

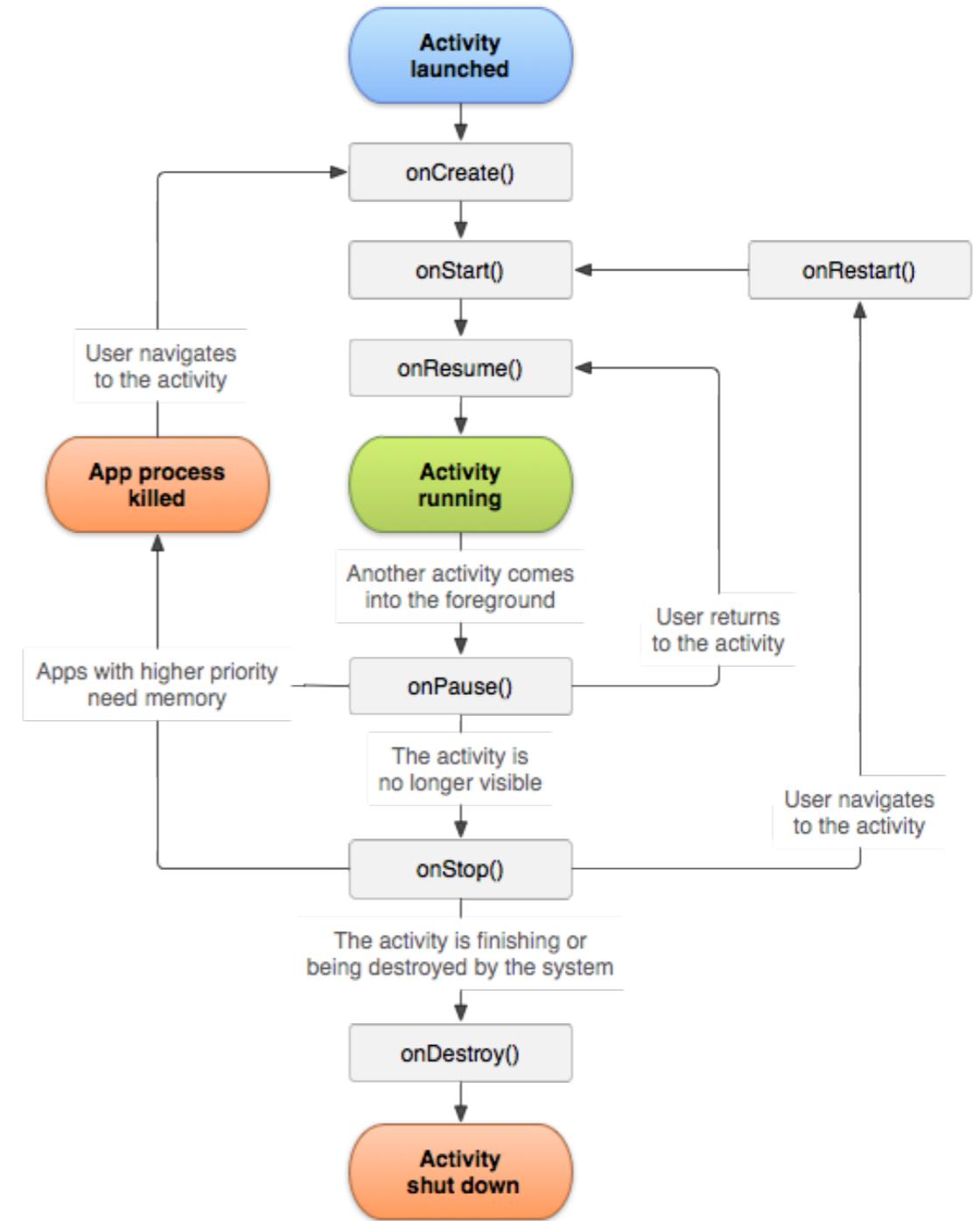
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.container2);

        /*
         * Hide navigate bar
         * https://stackoverflow.com/questions/21724420/how-to-hide-navigation-bar-permanently-in-android-activity
         */
        final int flags = View.SYSTEM_UI_LAYOUT_FLAGS
                | View.SYSTEM_UI_FLAG_HIDE_NAVIGATION | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION
                | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN | View.SYSTEM_UI_FLAG_FULLSCREEN
                | View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY;
        this.getWindow().getDecorView().setSystemUiVisibility(flags);

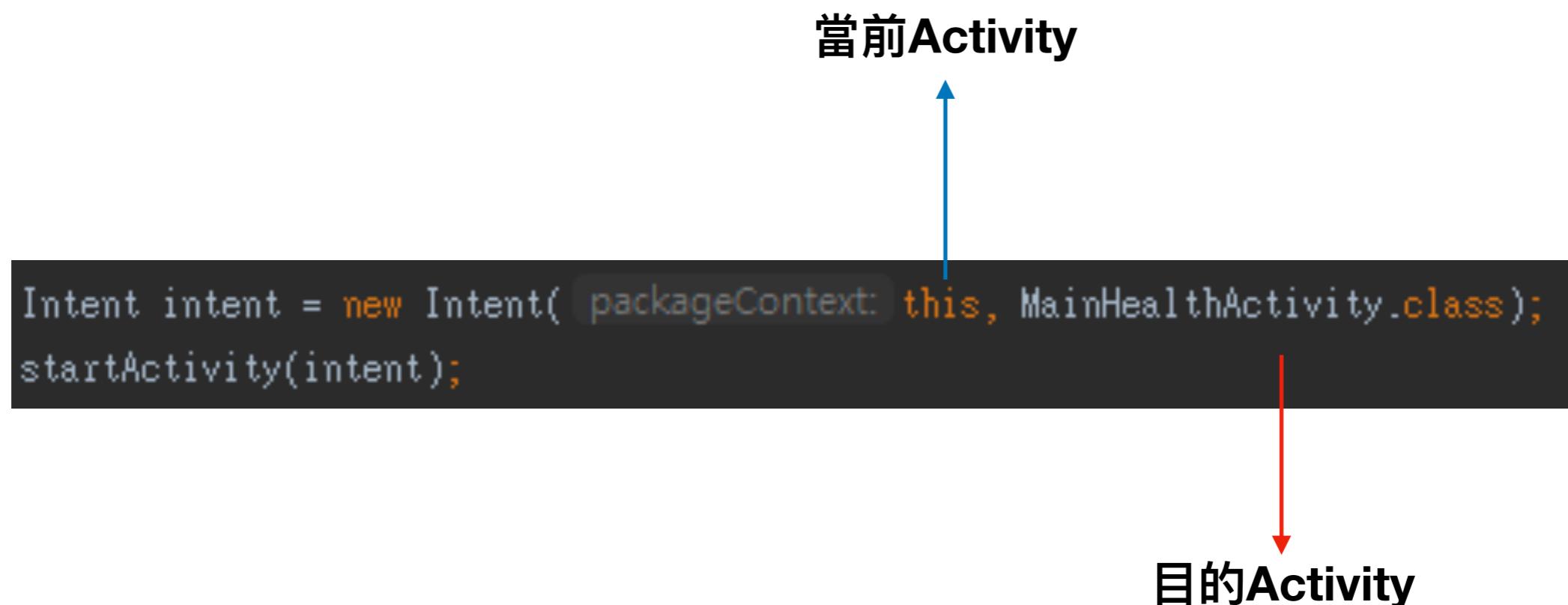
        // go to FirstFragment
        changeFragment(new FirstFragment());
    }
}
```

# Activity

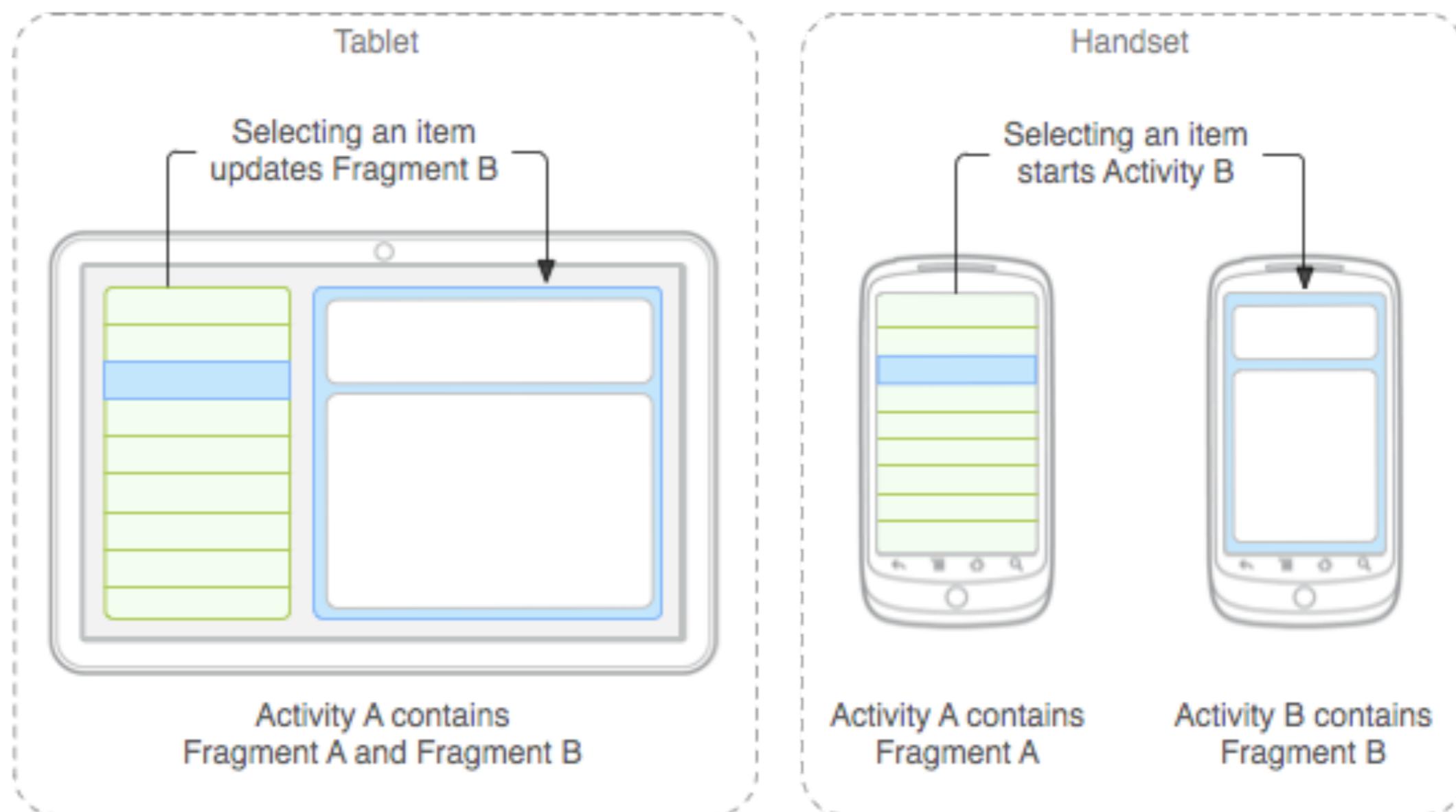
```
public class ExampleActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // The activity is being created.  
    }  
    @Override  
    protected void onStart() {  
        super.onStart();  
        // The activity is about to become visible.  
    }  
    @Override  
    protected void onResume() {  
        super.onResume();  
        // The activity has become visible (it is now "resumed").  
    }  
    @Override  
    protected void onPause() {  
        super.onPause();  
        // Another activity is taking focus (this activity is about to be "paused").  
    }  
    @Override  
    protected void onStop() {  
        super.onStop();  
        // The activity is no longer visible (it is now "stopped")  
    }  
    @Override  
    protected void onDestroy() {  
        super.onDestroy();  
        // The activity is about to be destroyed.  
    }  
}
```



# Activity - Transfer

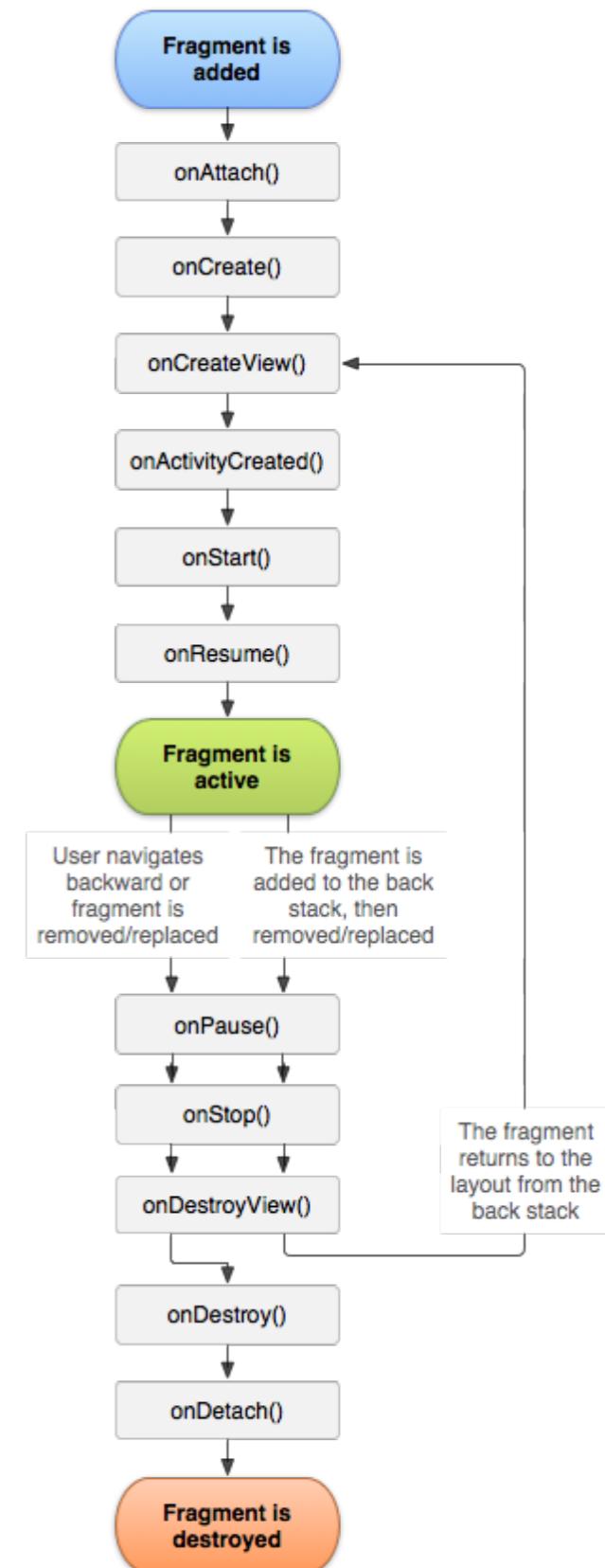


# Fragment



# Fragment

```
public class FirstFragment extends Fragment {  
    // UI components  
    Button taipeiBtn;  
    Button taichungBtn;  
    Button tainanBtn;  
  
    public FirstFragment() {  
        // Required empty public constructor  
    }  
  
    public static FirstFragment newInstance() {  
        FirstFragment fragment = new FirstFragment();  
  
        return fragment;  
    }  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); }  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
                            Bundle savedInstanceState) {  
        // Inflate the layout for this fragment  
        View view = inflater.inflate(R.layout.fragment_first, container, attachToRoot: false);  
  
        // Initialize UI button  
        taipeiBtn = view.findViewById(R.id.TaipeiButton);  
        taichungBtn = view.findViewById(R.id.TaichungButton);  
        tainanBtn = view.findViewById(R.id.TainanButton);  
  
        return view;  
    }  
}
```



# Fragment - Transfer

```
public void changeFragment(Fragment f) {  
    FragmentManager fm = getSupportFragmentManager();  
    fm.beginTransaction().replace(R.id.main_container, f).commit();  
}
```

```
FragmentManager fragmentManager = getSupportFragmentManager();  
fragmentManager.beginTransaction().replace(R.id.main_container, new FirstFragment()).commit();
```

**Without passing bundle**

# Bundle

```
Bundle bundle = new Bundle();
bundle.putString("city", "台南市");

FragmentManager fragmentManager = getFragmentManager();
fragmentManager.beginTransaction().replace(R.id.main_container, SecondFragment.newInstance(bundle)).commit();
```

## With passing bundle

```
//for accepting bundle
public static SecondFragment newInstance(Bundle args){
    SecondFragment fragment = new SecondFragment();
    fragment.setArguments(args);
    return fragment;
}
```

**newInstance for accepting in Fragment**

# Bundle

```
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    if (getArguments() != null) {  
        city = getArguments().getString("key: "city");  
        Log.d("tag: "SecondFragment", city);  
    }  
}
```

**Get data with bundle in onCreate**

# Retrofit

## Retrofit

A type-safe HTTP client for Android and Java

[Download Latest](#)



### *Introduction*

Retrofit turns your HTTP API into a Java interface.

```
public interface GitHubService {  
    @GET("users/{user}/repos")  
    Call<List<Repo>> listRepos(@Path("user") String user);  
}
```

The `Retrofit` class generates an implementation of the `GitHubService` interface.

```
Retrofit retrofit = new Retrofit.Builder()  
    .baseUrl("https://api.github.com/")  
    .build();  
  
GitHubService service = retrofit.create(GitHubService.class);
```

Each `Call` from the created `GitHubService` can make a synchronous or asynchronous HTTP request to the remote webserver.

```
Call<List<Repo>> repos = service.listRepos("octocat");
```

Use annotations to describe the HTTP request:

- URL parameter replacement and query parameter support
- Object conversion to request body (e.g., JSON, protocol buffers)
- Multipart request body and file upload

[Introduction](#)

[API Declaration](#)

[Retrofit Configuration](#)

[Download](#)

[Contributing](#)

[License](#)

[Javadoc](#)

[StackOverflow](#)

# Retrofit Interface

```
public interface WeatherServerClient {  
  
    String BASE_URL = "http://140.116.247.188:9898";  
  
    @GET("/weather/{cityName}/")  
    Call<ArrayList<Weather>> getWeatherInfo(@Path("cityName") String cityName);  
}
```

# Retrofit Request -1

```
public class RetrofitRequest {  
    // 以Singleton模式建立  
    public static RetrofitRequest mInstance = new RetrofitRequest();  
  
    //declare  
    RetrofitInterface RetrofitInterface;  
  
    public RetrofitRequest() {  
        // 設置baseUrl即要連的網站，addConverterFactory用Gson作為資料處理Converter ->轉JsonObject 不是JSONObject  
        Retrofit retrofit = new Retrofit.Builder()  
            .baseUrl(Global_Health.line_prefix) //因為parameter是url，可ignore  
            .addConverterFactory(GsonConverterFactory.create())  
            .build();  
  
        RetrofitInterface = retrofit.create(RetrofitInterface.class);  
    }  
  
    public static RetrofitRequest getInstance() {  
        if (mInstance == null)  
            mInstance = new RetrofitRequest();  
        return mInstance;  
    }  
  
    public RetrofitInterface getAPI_interface() { return RetrofitInterface; }
```

# Retrofit Request -2

```
public void Request_Username( String url) {  
  
    // 2. 透過RetrofitManager取得連線基底  
    RetrofitInterface = RetrofitRequest.getInstance().getAPI_interface();  
    // 3. 建立連線的Call，此處設置call為interface  
    Call<JSONObject> call = getAPI_interface().User_Name(url);  
    Log.d( tag: "current step", msg: "initialize call for get user name");  
  
    // 4. async 執行call  
    call.enqueue(new Callback<JSONObject>() {  
        @Override  
        public void onResponse(Call<JSONObject> call, Response<JSONObject> response) {  
            // 連線成功  
            JSONObject response_object = response.body();  
            Log.d( tag: "response string", response.toString());  
            Log.d( tag: "response msg", response.message());  
            Log.d( tag: "response success", String.valueOf(response.isSuccessful()));  
            Log.d("response body string", response.body().toString() );  
            if (response.isSuccessful()) {  
                Log.d( tag: "response body string", String.valueOf(response.body().isJsonNull()));  
                Log.d( tag: "LINE open id", response_object.get("open_id").getAsString());  
                Global_Health.current_user_id = response_object.get("open_id").getAsString();  
                Log.d( tag: "LINE current user id", Global_Health.current_user_id);  
            } else {  
                checkRequestContent(call.request());  
            }  
        }  
  
        @Override  
        public void onFailure(Call<JSONObject> call, Throwable t) {  
            // 連線失敗  
            Log.d( tag: "response fail", msg: "unsuccessful at getting user open id");  
            checkRequestContent(call.request());  
        }  
    });  
}
```

# Object Class(Self-define)

```
public class Weather {  
    @Expose  
    @SerializedName("city")  
    private String city;  
  
    @Expose  
    @SerializedName("weathertime")  
    private String weatherTime;  
  
    @Expose  
    @SerializedName("pop")  
    private String pop; // 降雨機率  
  
    @Expose  
    @SerializedName("temperature")  
    private String temparature;  
  
    @Expose  
    @SerializedName("rh")  
    private String rh; // 相對溼度  
  
    @Expose  
    @SerializedName("minci")  
    private String minci; // 最低舒適度
```

# Retrofit in Fragment

```
private void getCityWeather(View view, String city){
    DebugLogger.out( where: "WeatherFragment,getCityWeather", what: "create request: " + city);
    Retrofit r = new Retrofit.Builder()
        .baseUrl(WeatherServerClient.BASE_URL)
        .addConverterFactory(GsonConverterFactory.create())
        .build();
    r.create(WeatherServerClient.class).getWeatherInfo(city).enqueue(
        new Callback<ArrayList<Weather>>(){
            @Override
            public void onResponse(Call<ArrayList<Weather>> call, Response<ArrayList<Weather>> response) {
                DebugLogger.out( where: "WeatherFragment,onResponse", what: "isSuccess? " + response.isSuccessful());
                if (response.body() != null){
                    if (response.body().size() != 0){
                        setWeatherInfoToUI(view, response.body());
                        speakWeatherInfo(response.body());
                    }else{
                        Toast.makeText(getContext(), text: "今天天氣預測休息哦", Toast.LENGTH_SHORT).show();
                    }
                }
            }

            @Override
            public void onFailure(Call<ArrayList<Weather>> call, Throwable t) {
                DebugLogger.out( where: "WeatherFragment,onFailure", t.toString());
                new AlertDialog.Builder(getContext()).setMessage("今天天氣預測休息哦").show();
            }
        });
}
```

# Homework

- 連接Django作業的音樂API
  - 根據需求對API做request
  - 顯示歌曲名稱和演唱者

# Homework - Rule

- 命名 eg. weatherButton
- 註解
- ConstraintLayout
- Fragment Transfer
- RequestAPI
  - Retrofit Interface
  - Self-define class
  - Implement
- Bonus
  - App ICON
  - UI/UX