

● **FrameLayout**: 레이아웃에 포함된 뷰들을 같은 영역에 겹쳐서 배치할 때 사용한다. 뷰가 포함된 순서대로 배치되어, 맨 마지막에 포함된 뷰가 가장 위에 보인다. 이때 맨 마지막에 포함된 뷰의 크기가 가장 크다면, 전체를 가려서 마지막 뷰만 화면에 보인다. FrameLayout을 이용하는 목적은, 대부분 같은 영역에 여러 뷰를 겹치게 한 다음, 한 순간에 하나의 뷰만 보이기 위함이다. 이를 위해서는 뷰의 `visibility` 속성으로 특정 순간에 뷰가 보이거나 안 보이게 제어하면 된다.

● **TabLayout**: 탭(Tab) 화면을 구성하기 위한 레이아웃이다. material 라이브러리가 제공되기 전에는 TabHost라는 클래스를 이용하여 탭 화면을 만들었지만, 복잡한 앱이 개발되기 시작하면서 더 다양한 기능을 처리하는 탭 화면이 필요해졌고, 이를 지원하기 위해 TabLayout 뷰가 제공되었다. 즉, TabLayout은 탭 화면에서 탭 버튼을 다양하게 구성하기 위한 뷰이다. 단순히 탭 버튼을 동일한 간격으로 나열하는 것뿐만 아니라, 탭 버튼이 많을 때는 스크롤을 이용할 수도 있고 다양한 방식으로 정렬할 수 있다. 또한 TabLayout과 ViewPager2를 연동할 수도 있다.

● **Tab 화면 구현하기**: 탭 화면은 TabLayout으로 구성하고, 탭 버튼을 클릭했을 때 출력할 화면은 FrameLayout을 사용한다. TabLayout의 탭 버튼이 클릭이 되었을 때 나오는 화면은 FrameLayout 이외에 어떤 뷰를 이용해도 상관 없지만 탭 화면의 특성상, 탭 버튼이 나올 때 하나의 뷰만 출력하는 것이 일반적이므로, 대부분 FrameLayout을 이용한다.

#### TabLayout으로 탭 버튼 구성하기

1) 탭 버튼을 동적으로 구성하기: 코드로 구현하기

★ 뷰 결합: `build.gradle (:app)`에 다음 코드를 추가해 주어야 한다.

```
android {  
    ...  
    viewBinding {  
        enabled = true  
    }  
}
```

그런 다음 자바 코드에 변수를 선언해 준다.

```
private ActivityMainBinding binding;
```

MainActivity.java

```

TabLayout.Tab tab1 = binding.tabLayout.newTab();
tab1.setText("tab1");
binding.tabLayout.addTab(tab1);

TabLayout.Tab tab2 = binding.tabLayout.newTab();
tab2.setText("tab2");
binding.tabLayout.addTab(tab2);

TabLayout.Tab tab3 = binding.tabLayout.newTab();
tab1.setText("tab3");
binding.tabLayout.addTab(tab3);

```

Tab은 TabLayout에 들어갈 탭 버튼 하나를 지칭하는 클래스이며, newTab() 함수로 만든다. Tab 객체에 setText(), setIcon() 함수로 버튼을 구성하는 문자열과 이미지를 지정하고, 이렇게 만들어진 Tab 객체를 addTab() 함수의 매개변수로 전달하면, 탭 버튼이 TabLayout에 추가된다.

## 2) 탭 버튼을 정적으로 구성하기: XML 파일에서 구현(대부분)

★ Fragment 만들기: new > Fragment > Fragment (blank)

fragment\_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainFragment"
    android:background="#F44336">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="40sp"
        android:textColor="#FFFFFF"
        android:text="Fragment 1" />

</FrameLayout>

```

MainFragment.java

```

package com.example.tabpractice;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class MainFragment extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_main, container, false);
    }
}

```

#### activity\_main.xml

- ▶ tabMode 속성: scrollable로 하면, 탭 버튼이 많을 때 스크롤을 사용할 수 있다. scrollable로 지정하면 탭 버튼이 왼쪽부터 배치되며, 탭 버튼이 많아 화면을 벗어나면 자동으로 스크롤을 사용해 탭 화면을 움직일 수 있도록 해준다.
- ▶ tabGravity 속성: 정렬 위치를 지정할 수 있다. 기본은값은 fill이며, 탭 화면을 동일한 간격으로 나누어 가로 방향으로 꽉 들어차게 탭 버튼을 배치한다. fill 이외에도 center, start로 tabGravity 값을 지정할 수 있다.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical">

    <com.google.android.material.tabs.TabLayout
        android:id="@+id/tabLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:tabGravity="center">

        <com.google.android.material.tabs.TabItem
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="tab1" />

        <com.google.android.material.tabs.TabItem
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="tab2" />

        <com.google.android.material.tabs.TabItem
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="tab3" />
    </TabLayout>

```

```

</com.google.android.material.tabs.TabLayout>

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/container">

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/fragment_main"
        android:name="com.example.tabpractice.MainFragment"
    />

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/fragment_two"
        android:name="com.example.tabpractice.TwoFragment"
    />

    <fragment
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/fragment_three"
        android:name="com.example.tabpractice.ThreeFragment"
    />

</FrameLayout>

</LinearLayout>

```

#### MainActivity.java

각 탭 버튼을 클릭했을 때 이벤트 프로그램을 통해 원하는 화면이 나오게 제어해주어야 하고, 이를 위해 OnTabSelectedListener가 제공된다.

```

package com.example.tabpractice;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentTransaction;
import android.os.Bundle;
import com.google.android.material.tabs.TabLayout;

public class MainActivity extends AppCompatActivity {
    TabLayout tabLayout;
    MainFragment oneFragment;
    TwoFragment twoFragment;
    ThreeFragment threeFragment;

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

        tabLayout = findViewById(R.id.tabLayout);
        oneFragment = new MainFragment();
        twoFragment = new TwoFragment();
        threeFragment = new ThreeFragment();

        // oneFragment부터 화면에 나타나기
        getSupportFragmentManager().beginTransaction().add(R.id.container, oneFragment).

        // 탭 버튼을 클릭하면 원하는 화면으로 화면 바꾸기
        tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener() {
            // 탭 버튼을 선택할 때 이벤트
            @Override
            public void onTabSelected(TabLayout.Tab tab) {
                FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();
                if (tab.getPosition() == 0) {
                    transaction.replace(R.id.container, oneFragment);
                } else if (tab.getPosition() == 1) {
                    transaction.replace(R.id.container, twoFragment);
                } else if (tab.getPosition() == 2) {
                    transaction.replace(R.id.container, threeFragment);
                }
                transaction.commit();
            }

            // 다음 탭 버튼을 눌러 선택된 탭 버튼이 해제될 때 이벤트
            @Override
            public void onTabUnselected(TabLayout.Tab tab) {
            }

            // 선택된 탭 버튼을 다시 선택할 때 이벤트
            @Override
            public void onTabReselected(TabLayout.Tab tab) {
            }
        });
    }
}

```

