Fragment 生命周期

一. Fragment 中常用的生命周期方法：

当 Fragment 从创建到运行时回调的生命周期方法有：

1. onAttach():当 Fragment 依附到 Activity 时调用的方法
2. onCreate()：当 Fragment 创建时调用的方法
3. onCreateView()：给 Fragment 加载布局时调用的方法
4. onActivityCreated():当该 Fragment 依附的 Activity 创建时调用的方法
5. onStart():当 Fragment 启动时调用的方法
6. onResume():当 Fragment 正在运行时调用的方法

二. 当 Fragment 不在使用时调用的生命周期方法

onPause();当 Fragment 不在交互时调用该方法

onStop();当 Fragment 不再可见时调用该方法

onDestroyView():销毁 Fragment 布局时调用的方法

onDestroy()；当 Frament 销毁时调用的方法

onDetach()；当 Fragment 完全脱离 Fragment 时调用的方法

三. 生命周期方法调用的顺序如下图所示：

案例：使用代码演示 Fragment 生命周期执行顺序

具体要求：MainActivity 的布局中有两个按钮，一个是用于加载 Fragment 的，另外一个是切换到另外一个 Activity 的,界面代码入下所示：

MainActivity 界面：

```
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="com.imooc.demo.fragmentlifedemo.MainActivity">
    <Button
        android:id="@+id/myButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
    <LinearLayout
        android:id="@+id/cotainer"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="2">
    </LinearLayout>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="点击切换"
        android:id="@+id/anotherBut"/></LinearLayout>
```

Fragment 界面：

```
<RelativeLayout 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="com.imooc.demo.fragmentlifedemo.MyFragment">
    <TextView
        android:layout_marginTop="50dp"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_centerInParent="true"
        android:text="Fragment的布局显示" />
</RelativeLayout>
```

AnotherActivity 界面：

```
<RelativeLayout 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"
    android:paddingBottom="16dp"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.imooc.demo.fragmentlifedemo.AnoActivity">
  <TextView
      android:layout_centerInParent="true"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="我是另外的一个Activity"/>
</RelativeLayout>
```

代码部分：

```
public class MainActivity extends Activity {
    private Button myButton;
    private Button anotherButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myButton= (Button)findViewById(R.id.myButton);
        anotherButton= (Button) findViewById(R.id.anotherBut);
        //点击加载Fragment
        myButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FragmentManager mFragManager=getFragmentManager();
                FragmentTransaction fs = mFragManager.beginTransaction();
                MyFragment mFragment=new MyFragment();
                fs.add(R.id.cotainer, mFragment, "myFragment");
                fs.commit();
            }
        });
        //切换其他页
        anotherButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(MainActivity.this,AnoActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

Fragment 部分：

```
public class MyFragment extends Fragment {
    private static final String TAG="MyFragment";
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(TAG,"Fragment创建方法==onCreate() ");
    }
    public View onCreateView(LayoutInflater inflater,ViewGroup container,
                             Bundle savedInstanceState) {
        Log.d(TAG,"Fragment绑定布局==onCreateView() ");
        return super.onCreateView(inflater, container, savedInstanceState);
    }
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        Log.d(TAG,"依附的Activity创建的方法==onActivityCreated() ");
    }
    public void onStart() {
        super.onStart();
        Log.d(TAG,"Fragment启动的方法==onStart() ");
    }
    public void onResume() {
        super.onResume();
        Log.d(TAG,"Fragment正在运行时的方法==onResume() ");
    }
```

```
    public void onPause() {
        super.onPause();
        Log.d(TAG,"Fragment不再交互的方法==onPause() ");
    }
    public void onStop() {
        super.onStop();
        Log.d(TAG,"Fragment停止运行的方法==onStop() ");
    }
    public void onDestroyView() {
        super.onDestroyView();
        Log.d(TAG,"Fragment视图销毁的方法==onDestroyView() ");
    }
    public void onDestroy() {
        super.onDestroy();
        Log.d(TAG,"Fragment销毁的方法==onDestroy() ");
    }
    public void onDetach() {
        super.onDetach();
        Log.d(TAG,"Fragment脱离Activity的方法==onDestroy() ");
    }
}
```

AnotherActivity 代码部分：

```
public class AnoActivity extends Activity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ano);
    }
}
```

运行效果：

```
D/MyFragment: Fragment创建方法==onCreate()
D/MyFragment: Fragment绑定布局==onCreateView()
D/MyFragment: 依附的Activity创建的方法==onActivityCreated()
D/MyFragment: Fragment启动的方法==onStart()
D/MyFragment: Fragment正在运行时的方法==onResume()
D/MyFragment: Fragment不再交互的方法==onPause()
D/MyFragment: Fragment停止运行的方法==onStop()
D/MyFragment: Fragment视图销毁的方法==onDestroyView()
D/MyFragment: Fragment销毁的方法==onDestroy()
D/MyFragment: Fragment脱离Activity的方法==onDestroy()
```