

方法一：传递对象的方法

1、在Activity中声明公共方法：

```
private List<View> v = new ArrayList<>();  
TextView mTEXT = findViewById(R.id.text);  
v.add(mTEXT);
```

```
public List<View> getViews(){  
    return v;  
}
```

2、在Fragment中接收

```
MainActivity activity = (MainActivity) getActivity();  
List<View> views = activity.getViews();  
TextView view = (TextView) views.get(0);
```

方法二：传递数据的方法

1、在Activity进入Fragment之前：

```
FragmentManager mfm = getSupportFragmentManager();  
FragmentTransaction transaction = mfm.beginTransaction();  
Fragment f = new Fr();  
Bundle bundle = new Bundle();  
bundle.putString("AS","5252");  
f.setArguments(bundle);  
transaction.replace(R.id.layout,f);  
transaction.commit();
```

2、在Fragment中接收

//接收数据

```
Bundle arguments = getArguments();  
String as = arguments.getString("AS");
```

回传问题 Fragment打开Activity，Activity操作完回传数据到Fragment

1、Fragment要做的事：

```
Intent intent = new Intent(getActivity(),ManualChangLineActivity.class);
```

```
intent.putExtra("worknum",mCurrentWork );  
startActivityForResult(intent, 0);
```

重写接收数据

@Override

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {  
if ((requestCode == 0 && resultCode == 0) && data != null) {  
String string = data.getStringExtra("worknum");  
//存为sp
```

```
Tools.putString(context, "listdata", string);  
mCurrentWorkNums.setText(string);  
}  
super.onActivityResult(requestCode, resultCode, data);  
}
```

2、Activity要做的事：获取数据回传

```
String selctDate = mSelectworknum.getText().toString().trim();  
Intent intent = new Intent();  
intent.putExtra("worknum", selctDate);  
ManualChangLineActivity.this.setResult(0, intent);  
ManualChangLineActivity.this.finish();
```