

中山大学移动信息工程学院本科生实验报告

(2017 年秋季学期)

课程名称：移动应用开发

任课教师：郑贵锋

| | | | |
|------|------------------|--------|------------------|
| 年级 | 2015 | 专业（方向） | 移动互联网 |
| 学号 | 15352286 | 姓名 | 任萌 |
| 电话 | 13763361232 | Email | 352600801@qq.com |
| 开始日期 | 2017 年 10 月 10 日 | 完成日期 | 2017 年 10 月 13 日 |

一、 实验题目

实验二：事件处理

二、 实验目的

1. 了解 Android 编程基础
2. 熟悉 ImageView、Button、RadioButton 等基本控件，能够处理这些控件的基本事件
3. 学会弹出基本的对话框，能够定制对话框中的内容，能对确定和取消按钮的事件做处理

三、 实现内容

实现一个 Android 应用，界面呈现与实验一基本一致，要求：

Lab2

中山大学学生信息系统



请输入学号

请输入密码

☒ 学生 ☐ 教职工

登录

注册

- (1) 该界面为应用启动后看到的第一个界面
- (2) 输入学号和密码的控件要求用 **TextInputLayout** 实现
- (3) 点击图片，弹出对话框如下图：



- 点击“拍摄”选项，弹出 Toast 信息“您选择了[拍摄]”；
- 点击“从相册选择”选项，弹出 Toast 信息“您选择了[从相册选择]”；
- 点击“取消”按钮，弹出 Toast 信息“您选择了[取消]”。
- (4) 切换 **RadioButton** 的选项，弹出 **Snackbar** 提示“您选择了 xx”；
例如从选项“学生”切换到选项“教职工”，则提示“您选择了教职工”；
点击 **Snackbar** 上的“确定”按钮，则弹出 Toast 信息“Snackbar 的确定按钮被点击了”。
- (5) 点击登录按钮，**依次判断**学号是否为空，密码是否为空，用户名和密码是否正确（正确的学号和密码分别为“123456”，“6666”）；不正确则给出错误信息，如学号和密码都正确则提示“登陆成功”。
- (6) 点击注册按钮，如果切换选项时，**RadioButton** 选中的是“学生”，那么弹出 **Snackbar** 信息“学生注册功能尚未启用”，如果选中的是“教职工”，那么弹出 Toast 信息“教职工注册功能尚未启用”。

使用到的代表性函数：

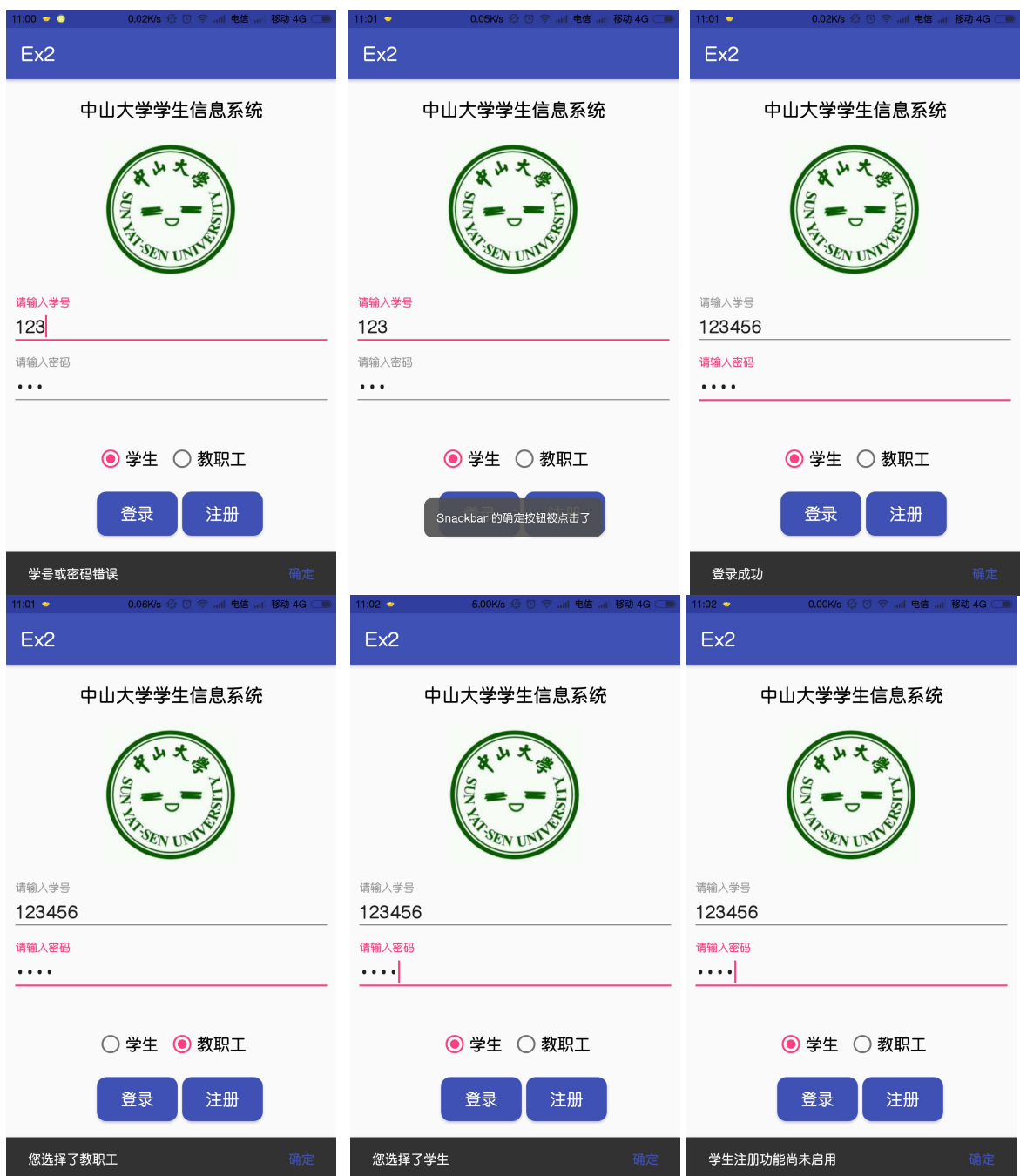
| 函数名称 | 功能 |
|------------------------|--------------|
| alertDialog.setTitle() | 设置对话框标题 |
| alertDialog.setItems() | 设置对话框元素 |
| OnClickListener() | 创建监听器 |
| findViewById() | 根据 id 查找布局元素 |

| | |
|----------------------|--------------------|
| setContentView() | 加载布局 |
| Toast.makeText() | 显示 Toast 信息 |
| setActionTextColor() | 设置右侧按钮文本的颜色 |
| setDuration() | 定义 Snackbar 弹出时间长度 |
| setErrorEnabled() | 设置错误提示信息的显示状态 |

四、 课堂实验结果

(1) 实验截图





结果分析：

(2) 实验步骤以及关键代码

Step 1 复制布局

建一个上一次实验的副本，然后将该副本调试正常，并作为这次实验的基础。

Step 2 布局修改部分

在布局文件中将输入学号和密码的 TextView 等控件改成 TextInputLayout 控件。使用 TableLayout 嵌套布局实现对两行输入框的整体设计。

```
<android.support.design.widget.TextInputLayout
    android:id="@+id/user"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/sysu"
    android:layout_marginBottom="10dp">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintRight_toRightOf="parent"
        android:layout_marginRight="20dp"
        android:id="@+id/user_edit"
        android:hint="请输入学号"
        android:inputType="number"
        android:textSize="18sp" />
</android.support.design.widget.TextInputLayout>
```

Step 3 点击图像部分

打开 Ex2.java 文件，新建一个 alertDialog 并通过 AlertDialog.Builder() 进行初始化，同时定义字符串数组 items 为对话框中的待选项显示内容。然后为 alertDialog 设置标题、元素及触发事件的 Toast 信息，重写 onClick() 函数，通过下标 i 来选择显示内容。最后用 findViewById() 函数，通过在 XML 中定义好的 id，找到 ImageView 控件，并通过 setOnClickListener() 调用 alertDialog 的 show() 来显示 Toast 信息。注意：由于 findViewById() 方法返回的是一个 View 类型，所以需要强制类型转换为 ImageView 类型。

```
// 点击图像部分
final AlertDialog.Builder alertDialog = new AlertDialog.Builder(this);
final String [] items = {"拍摄", "从相册选择"};
alertDialog.setTitle("上传头像").setItems(items,
    (dialogInterface, i) -> {
        Toast.makeText(getApplicationContext(), "您选择了"+items[i], Toast.LENGTH_SHORT).show();
    }).setNegativeButton("取消",
    (dialogInterface, i) -> {
        Toast.makeText(getApplicationContext(), "您选择了[取消]", Toast.LENGTH_SHORT).show();
    }).create();

ImageView logo = (ImageView) findViewById(R.id.sysu);
if(logo != null) {
    logo.setOnClickListener((view) -> { alertDialog.show(); });
}
```

Step 4 切换 RadioButton

用 `findViewById()` 函数，通过 `id` 找到 `RadioButton` 控件和 `RadioGroup` 控件，并对它们进行命名。通过 `setOnCheckedChangeListener` 函数监听并处理 `RadioButton` 切换事件，弹出相应提示。使用 `Snackbar` 显示身份选择部分的提示信息，在其内部重写 `onClick()` 函数实现当“确定”按钮被点击时弹出 `Toast` 提示信息。

```
// 选择radiobutton部分
final RadioGroup radioGroup1 = (RadioGroup) findViewById(R.id.buttonGroup);
radioGroup1.setOnCheckedChangeListener((group, checkedId) -> {
    RadioButton r1 = (RadioButton) findViewById(R.id.student);
    RadioButton r2 = (RadioButton) findViewById(R.id.teacher);
    if (checkedId == r1.getId()) {
        Snackbar.make(r1, "您选择了学生", Snackbar.LENGTH_SHORT)
            .setAction("确定", new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                }
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    } else if (checkedId == r2.getId()) {
        Snackbar.make(r1, "您选择了教职工", Snackbar.LENGTH_SHORT)
            .setAction("确定", new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                }
            })
            .setActionTextColor(getResources().getColor(R.color.colorPrimary))
            .setDuration(5000)
            .show();
    }
});
```

Step 5 登录部分

和 `RadioButton` 部分类似，首先在 `java` 文件中引入 `TextInputLayout`，通过 `findViewById()` 找到 `Button` 控件和两个输入框控件并命名，然后重写 `onClick()` 函数。利用两个字符串变量定义正确的用户名和密码，通过 `TextInputLayout` 获取 `EditText` 中的内容，再通过 `getText()` 和 `toString()` 将输入内容转换为字符串。利用 `setErrorEnabled()` 设定是否显示错误提示信息，初始状态为 `false`。

在“登录”按键被点击时，首先判断学号是否为空，若为空则设置 `user` 的 `error` 触发状态为 `true`，调用 `setError()` 提示“学号不能为空”，然后判断密码是否为空，若为空则设置 `key` 的 `error` 触发状态为 `true`，调用 `setError()` 提示“密码不能为空”。若均非空，则恢复 `user` 和 `key` 的 `error` 触发状态为 `false`，再判断学号与密码是否正确，若正确则通过 `Snackbar` 显示“登录成功”，否则通过 `Snackbar` 显示错误信息。


```
// 登录部分
Button button1 = (Button) findViewById(R.id.button1);
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        TextInputLayout user = (TextInputLayout) findViewById(R.id.user);
        TextInputLayout key = (TextInputLayout) findViewById(R.id.key);
        EditText username = user.getText();
        EditText password = key.getText();
        String s1 = "123456";
        String s2 = "6666";
        String s3 = username.getText().toString();
        String s4 = password.getText().toString();
        user.setErrorEnabled(false);
        key.setErrorEnabled(false);
        if (TextUtils.isEmpty(s3)) {
            user.setErrorEnabled(true);
            user.setError("学号不能为空");
        } else if (TextUtils.isEmpty(s4)) {
            key.setErrorEnabled(true);
            key.setError("密码不能为空");
        } else if (s1.equals(s3) && s2.equals(s4)) {
            user.setErrorEnabled(false);
            key.setErrorEnabled(false);
            Snackbar.make(user, "登录成功", Snackbar.LENGTH_SHORT)
                .setAction("确定", (view) -> {
                    Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                })
                .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                .setDuration(5000)
                .show();
        } else {
            user.setErrorEnabled(false);
            key.setErrorEnabled(false);
            Snackbar.make(key, "学号或密码错误", Snackbar.LENGTH_SHORT)
                .setAction("确定", (view) -> {
                    Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                })
                .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                .setDuration(5000)
                .show();
        }
    }
});
```

Step 6 注册部分

注册部分处理方法和 RadioButton 部分基本一样，只是要先用 `findViewById()` 函数先找到“注册”按钮，在它被点击时再执行类似操作。注意：学生注册的提示信息是由 `Snackbar.make()` 给出，教职工注册的提示信息是由 `Toast.makeText()` 给出。

```
// 注册部分
ImageView img = (ImageView) findViewById(R.id.sysu);
final AlertDialog.Builder alertDialog2 = new AlertDialog.Builder(this);

Button button2 = (Button) findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        RadioButton r1 = (RadioButton) findViewById(R.id.student);
        RadioButton r2 = (RadioButton) findViewById(R.id.teacher);
        if (r1.isChecked())
            Snackbar.make(r1, "学生注册功能尚未启用", Snackbar.LENGTH_SHORT)
                .setAction("确定", new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        Toast.makeText(getApplicationContext(), "Snackbar 的确定按钮被点击了", Toast.LENGTH_SHORT).show();
                    }
                })
                .setActionTextColor(getResources().getColor(R.color.colorPrimary))
                .setDuration(5000)
                .show();
        if (r2.isChecked()) {
            Toast.makeText(getApplicationContext(), "教职工注册功能尚未启用", Toast.LENGTH_SHORT).show();
        }
    }
});
```

(3) 实验遇到困难以及解决思路

问题 1：刚开始全部都用 SetOnClickListener 解决 radiobutton 切换跳出提示的功能，仔细看了课件才发现这里的切换应该用 setOnCheckedChangeListener 来解决，但是我使用该函数的时候用了与 SetOnClickListener 一样的语法会报错。

解决方法：在网上搜关于 SetOnClickListenerChangeListener 函数的用法，寻找功能相近的代码去模仿。后来搞懂了两个问题，一个是 setOnCheckedChangeListener() 函数中括号的参数不是 view.XXX，而是 RadioGroup.OnCheckedChangeListener。第二个是要用 checkedId == r1.getId() 语句去判断是否被选择，而不是 r1.isChecked 语句，然后就搞定了它。

问题 2：代码文件中 ImageView、Button 等关键字报错 cannot resolve symbol xxx。

解决方法：我查阅了各种资料和大量的博客，使用了 Invalidate caches / restart 等一系列方法都没能结果，后来在问同学的过程中我发现是我没有包含库文件，使用 alter+enter 进行补全后代码就可以正常运行了。

问题 3：RadioGroup.OnCheckedChangeListener 中的函数不能 Override。

解决方法：查阅资料后，我发现原来有时自己写的 Override 代码它不会自动识别成它要覆盖的函数，所以我就把代码删了重新打，该让它自己生成的地方让它自己生成。最后在用 alter+enter 补全缺少的库，就可以了。

问题 4：AlertDialog 总是有错误。

解决方法：因为开始那里我把所有事件都放在一起写，所以比较繁杂不好发现错误。后来，我就把 AlertDialog 的创建和里面弹出内容的设定等单独写在一起，然后之后登陆成功或者失败的时候直接去调用它，让它 show() 出来就简单的多，也查出了错误所在。

问题 5：使用 TextInputLayout 布局的时候一直报错。

解决方法：首先我发现我做实验的顺序有点不对，我应该先修改 build.gradle 文件中的配

置，然后再去写 TextInputLayout 布局的，然而我顺序搞反了，然后我就重新按正常顺序写了遍然后同步。结果还是不行，我上网搜了下解决办法，改了 Theme，然后 Clean 下就解决了。

问题 6：对用户名和密码是否正确做不出正确判断。

解决方法：在 “String s3 = username.getText().toString();” 这个语句中，我少了 getText 这一句，然后就会读出莫名其妙的字符串，加上以后就正常了。

问题 7：在某次打开 Project 的时候 java 文件突然报错 cannot resolve symbol R。

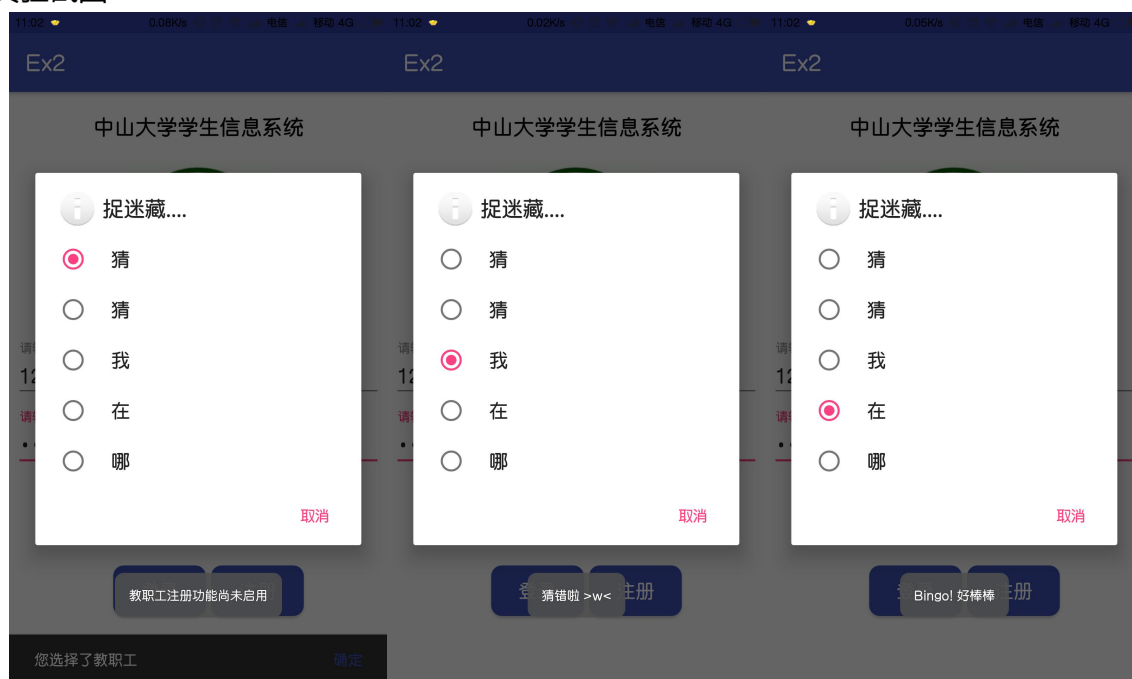
解决方法：直接 build → rebuild project 就好了。

五、 课后实验结果

（1） 实验内容

- ① 增加了对输入框内输入内容的长度限定
- ② 利用对话框设计了“捉迷藏”小游戏

（2） 实验截图



结果分析：在原来的功能基础上增加了“捉迷藏”对话框小游戏。点击教职工的“注册”按键时，自动弹出小游戏，通过触摸选择选项，若猜中则弹出提示信息“Bingo！好棒棒”，否则弹出提示信息“猜错啦>w<”。

（3） 实验步骤以及关键代码

Step 1 增加对输入框内输入内容的长度限定

使用 maxLength 即可。

```
android:maxLength="16"
```

Step 2 利用对话框设计“捉迷藏”小游戏

Step 2.1 新定义一个对话框对象 alertDialog2。

```
final AlertDialog.Builder alertDialog2 = new AlertDialog.Builder(this);
```

Step 2.2 利用 `setIcon()` 设置对话框的图标，利用 `setSingleChoiceItems()` 设置单选选项名称，利用 `OnClickListener()` 监听并处理点击选项事件，根据不同的选项选择显示出不同的 Toast 信息。

```
alertDialog2.setTitle("捉迷藏...")
    .setIcon(android.R.drawable.ic_dialog_info)
    .setSingleChoiceItems(new String[]{"猜", "猜", "我", "在", "哪"}, 0,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                if (which == 3)
                    Toast.makeText(getApplicationContext(), "Bingo! 好棒棒", Toast.LENGTH_SHORT).show();
                else
                    Toast.makeText(getApplicationContext(), "猜错啦 >w<", Toast.LENGTH_SHORT).show();
            }
        })
    .setNegativeButton("取消", null)
    .show();
```

(4) 实验遇到困难以及解决思路

问题 1：在 `setIcon` 时直接从 `mipmap` 中选取了图片文件，导致语法报错。

解决方法：改为选取 AS 自带的图标文件 `android.R.drawable.ic_dialog_info` 即可。

六、实验思考及感想

本次实验是一些基本的事件处理。开始做实验的时候再一次觉得无从下手，后来找到了规律后就轻松了很多。觉得 Android Studio 在提示输入方面做得真的很棒，大部分组件、功能、参数的命名非常好。经常我不知道该怎么写下去时，随便猜个名字，然后 Android Studio 就会给很多提示，在里面选一个就好了。

其实这次实验觉得比较郁闷的地方还是搞软件的功夫都快和打代码的事件一样长了。因为我们用的 AndroidStudio 里面其实大多东西都是开发者定义好的类，我们只要去调用它们就可以实现我们想要的功能，这样很方便也很简单。不过这样直接导致的缺点就是，我们一般不会去看它们的内部实现，一些使用，重载什么的经常会出问题，我遇到的问题在实验报告“问题与讨论”部分已经说了，还是比较浪费时间的。

觉得事件处理这里开始写 java 了，会比界面 XML 难一点，不过也是开始觉得难啦，把刚开始一段艰难地打完，后面进行的还是比较快的。不过有很多细节还是值得去注意的，比如我虽然听了课，没有入“`str1.equals (str2)`”的坑，但是我在之前获得了 `EditText` 的时候并没有用 `getText()` 去获得其中的内容。还有就是 `TextUtils.isEmpty(str1)` 和 `str1.isEmpty()` 的区别在与前者在 `str1` 等于 “ ” 的时候也会返回 `true`。这些都是语法的细节问题，也是未来我们需要掌握的。