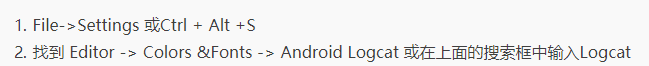
Android Studio3.X(目前包括3.0.0,3.0.1不支持32位Windos!注意!,使用2.3.3即可)

导入项目

修改build.gradle

## 查看Logcat try/catch , sout

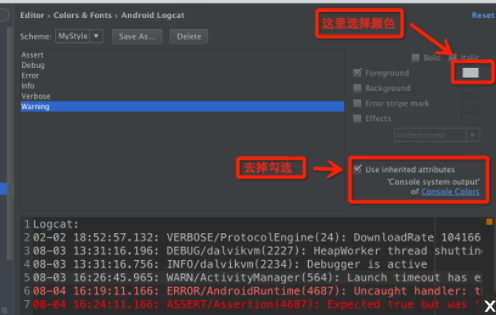


1. 级别verbose:用于打印那些最为繁琐的、意义最小的日志信息。是Android日志里面级别最低；

2. 级别debug:用于打印一些调试信息，有助于调试程序和分析问题，比verbose高一级；

3. 级别info：用于打印一些比较重要的数据，这些数据是你非常想看到的数据、可帮你分析数据。

4.对应warn：用于打印一些警告信息，提示程序在这个地方存在潜在风险，最好去修复出现警告的地方。

5.对应级别error。：用于打印程序中错误信息，比如程序进入到catch语句当中，当有错误信息打印出来时，一般代表你的程序出现严重问题了，必须尽快修复。

private final String TAG = "MainActivity";

Log.e(TAG , name);

## 运行debug,

## 权限,

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>

//Activity开启的时候执行，我们需要在ACtivity开启的时候为按钮设置点击事件。

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//1.找到输入框和按钮控件的对象

final EditText et\_number = (EditText) findViewById(R.id.et\_number);

Button bt\_callphone = (Button) findViewById(R.id.bt\_callphone);

//2.为按钮设置点击事件 bt\_callphone.setOnClickListener(new OnClickListener() {

//当控件被点击的时候调用

@Override

public void onClick(View v) {

//3.当用户点击按钮时，我们需要获取输入框上输入的电话号码，判断号码是否是kong

String number = et\_number.getText().toString().trim();

if(TextUtils.isEmpty(number)){

//context:上下文,当前应用的一个对象 text:提示的内容 duration:提示的持续时间

Toast.makeText(getApplicationContext(), "号码不能为null", Toast.LENGTH\_SHORT).show();

return;

}

//4.不是空就将电话打出去。 number Intent :意图对象， 理解成一个奴隶，让他干啥他干啥

//a.创建一个意图对象 创建了个奴隶

Intent intent = new Intent();

//b.为意图对象设置个ａｃｔｉｏｎ　动作; 告诉奴隶要干什么事

intent.setAction("android.intent.action.CALL");

//c.为意图设置数据 相当于告诉奴隶要打哪个电话号码

intent.setData(Uri.parse("tel:"+number));

//d.开启意图， 奴隶打电话了

startActivity(intent);

}

});

5.在androidmanifest中加拨打电话的权限。  
 <uses-permission android:name="android.permission.CALL\_PHONE"/>

## 按钮的4中点击事件

//2.为按钮设置点击事件

/\* //☆☆☆☆ 按钮点击事件1 匿名内部类

bt\_callphone.setOnClickListener(new OnClickListener() {

//当控件被点击的时候调用

@Override

public void onClick(View v) {

callPhone();

}

});

\*/

/\* //☆☆☆☆ 按钮点击事件2 写个继承OnclickListner

bt\_callphone.setOnClickListener(new MyOnclickListener());\*/

class MyOnclickListener implements OnClickListener{

@Override

public void onClick(View v) {

callPhone();

}

}

/\* //☆☆☆☆ 按钮点击事件3 让当前类直接实现OnclickListner

//注意: class MainActivity extends AppCompatActivity implements OnClickListener{

bt\_callphone.setOnClickListener(this);//this:当前类的一个对象，我MainACtivity已经具备了OnClickListener所有属性和方法

Button button1 = (Button) findViewById(R.id.button1);

Button button2 = (Button) findViewById(R.id.button2);

button1.setOnClickListener(this);

button2.setOnClickListener(this);\*/

}

//按钮点击时调用

@Override

public void onClick(View v) {

//view ：当用户点击了哪个控件这个view就代表谁

//获取被点击的控件的名字

int id = v.getId();

switch (id) {

case R.id.bt\_callphone:

callPhone();

break;

case R.id.button2:

Toast.makeText(getApplicationContext(), "bt2点击", 0).show();

break;

default:

break;

}

}

//按钮的点击事件第4种; 只要为要点击的控件设置一个Onclick属性，回到布局所在的ACtivity中创建一个以属性值为名的方法，该方法就是按钮的点击事件;底层采用反射

/\* public void bttt1(View v){

Toast.makeText(getApplicationContext(), "bt2点击", 0).show();

}

\*/

//XML文件里面写: android:onClick="bttt1"

//总结:

1.匿名内部类  
2.写个类继承OnclickListner ， setOnclickListener时传该类的对象  
3.让当前类直接实现OnclickListner，setOnclickListener时传this；该方式是在公司开发中使用的方式。  
4.只要为要点击的控件设置一个Onclick属性，回到控件所在布局的ACtivity中创建一个以属性值为名的方法，该方法就是按钮的点击事件;底层采用反射;开发中用的不多，一般测试时使用比较方便.

## 保存Sharepreference

SharedPreferences sharedPreferences = getSharedPreferences("Test", Context.***MODE\_PRIVATE***); ***//私有数据***SharedPreferences.Editor editor = sharedPreferences.edit();***//获取编辑器***editor.putString("name", "张三");  
editor.putInt("age", 4);  
editor.putBoolean("Married" , false);  
editor.commit();***//提交修改***String name = sharedPreferences.getString("name", "");  
int age = sharedPreferences.getInt("age", 1);  
boolean married = sharedPreferences.getBoolean("Married", true);  
  
System.***out***.println("name"+name+"age"+age +"Married"+married );

## ###启动界面的两种方式

\* 显式意图

1. 直接从intent身上能够知道跳转到哪一个界面

2. 只需要有一个intent对象即可

Intent intent = new Intent(this , OtherActivity.class);

startActivity(intent);

3. 效率高一点

\* 隐式意图

1. 无法直接从intent身上知道跳转的目的， 必须依赖于后面的action设置才可以。

2. 必须对intent设置action动作 或者 data ...

Intent intent = new Intent();

intent.setAction(Intent.ACTION\_CALL);

intent.setData(Uri.parse("tel://13"));

startActivity(intent);

3. 效率低一点，因为要寻找匹配的activity

action : 动作

data ： 数据 scheme

category ： 分类

我们在开发当中很少去定义隐式意图。 隐式意图的最大作用就是 。。。

> 用来启动其他应用程序的界面

# butterknife (插件的使用(Android ButterKnife Zelezny), github(https://github.com/))

# sqlite(greendao)