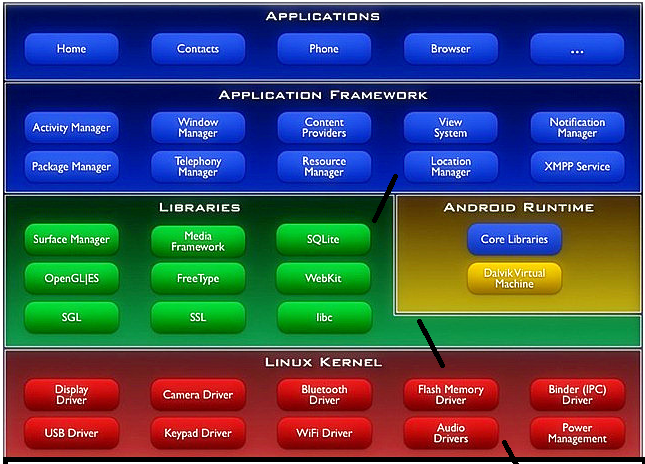
1. Web基础
   1. 了解html+css+js基本语法
   2. 了解安卓版本发展历史；熟悉安卓体系结构



* 1. 了解服务器和servet工作原理；掌握Http协议，包括概念（超文本传输协议、定义了Client/Broswer和Server交互的规则），特点（基于TCP/IP协议），请求方式（Get、Post），请求格式（request-line请求行、headers字段、blank line空行、body请求体），响应格式（状态行、响应头、响应实体）熟悉servlet编程。

1. 安卓原生布局
   1. LinearLayout 线性布局：每一个孩子都会在上一个孩子的下边或者右边，注意必须配合orientation
   2. RelativeLayout 相对布局：所有孩子都可以指定具体位置，层叠堆放，每个孩子默认在左上角
   3. FrameLayout 帧布局:所有孩子都在左上角层叠堆放，最省性能，常用于占位。
   4. TableLayout表格布局、AbsoluteLayout绝对布局，熟悉即可。
2. 入门项目
   1. 短信发送器
   2. Junit单元测试，Log技巧

//注意权限

String destinationAddress = "110";//目标号码

String scAddress = **null**;//服务中心地址

String text = "大王叫我来巡山"; //短信内容

PendingIntent sentIntent = **null**;//回调，给null即可

PendingIntent deliveryIntent = **null**;//回调，给null即可

SmsManager.*getDefault*().sendTextMessage(destinationAddress, scAddress, text, sentIntent, deliveryIntent);

* 1. 掌握点击事件的四种写法

v.setOnClickListener(**new** MyOnClickListener());

}

**private** **class** MyOnClickListener **implements** OnClickListener{

@Override

**public** **void** onClick(View v) {

// 第三种写法，让内部类实现OnClickListener接口

}

}

v.setOnClickListener(**this**);

}

@Override

**public** **void** onClick(View v) {

//第二种写法，让外部类实现OncliClick接口

}

v.setOnClickListener(**new** OnClickListener() {

@Override

**public** **void** onClick(View v) {

//第一种写法,匿名内部类

}

});

android:onClick="click"

**public** **void** click(View v){

//第四种写法，onClick属性

}

1. XML
   1. 了解概念、语法。
   2. 熟悉Pull解析。
   3. 重点掌握SharedPreference。

String name = "config";//sp文件名

**int** mode = Context.***MODE\_PRIVATE***;//SP模式（私有）

SharedPreferences sharedPreferences = context.getSharedPreferences(name , mode);

//写入SP

sharedPreferences.edit().putString("key", "value").commit();

//读取SP

String value = sharedPreferences.getString("key", "defValue");

1. 糗事百科
   1. 熟悉服务器和Android端交互流程。
   2. 重点掌握json解析（JsonObject、Gson选择一种掌握，建议Gson）。

String jsonObject = "{\"name\":\"zhangsan\";\"age\":\"20\"}";

String jsonArray = "[{\"name\":\"zhangsan\";\"age\":\"20\"},{\"name\":\"lisi\";\"age\":\"21\"}]";

Gson gson = **new** Gson();

Student student = gson.fromJson(jsonObject, Student.**class**);//json对象解析

List<Student> students = gson.fromJson(jsonArray, **new** TypeToken<List<Student>>(){}.getType());//json集合解析

}

**public** **static** **class** Student{

**public** String name;

**public** String age;

}

1. 安卓数据储存（重点掌握）
   1. 私有文件储存。
   2. SD卡文件储存。

//私有目录储存

**new** File(getFilesDir(),"privateFile.txt");//私有目录

**new** File(getCacheDir(),"cacheFile.txt");//私有缓存目录

//SD卡

String storageState = Environment.*getExternalStorageState*();

**if**(Environment.***MEDIA\_MOUNTED***.equals(storageState)){

//表示SD卡已挂载

**new** File(Environment.*getExternalStorageDirectory*(), "sdFile.txt");//SD卡目录

}**else**{

//SD卡未挂载，可以提示用户。

}

* 1. SQLite数据库

//通过SQLiteOpenHelper来创建数据库

MySQLiteOpenHelper helper = **new** MySQLiteOpenHelper(**this**);

SQLiteDatabase database = helper.getWritableDatabase();

...

database.close();//关闭数据库

//增-谷歌api

ContentValues values = **new** ContentValues();

values.put("name", "zhangsan");

values.put("age", 13);

database.insert("student", **null**, values);

//增-SQL语句

database.execSQL("insert into student values(null,'lisi',14)");

//查-谷歌api

Cursor cursor1 = database.query("student", **new** String[]{"name","age"}, **null**, **null**, **null**, **null**, **null**);

**while**(cursor1.moveToNext()){

System.***out***.println("name = " + cursor1.getString(cursor1.getColumnIndex("name"))

+ "age = " + cursor1.getString(cursor1.getColumnIndex("age")));

}

cursor1.close();

//查-SQL语句

Cursor cursor2 = database.rawQuery("select \* from student", **null**);

**while**(cursor2.moveToNext()){

System.***out***.println("name = " + cursor2.getString(cursor2.getColumnIndex("name"))

+ " age = " + cursor2.getString(cursor2.getColumnIndex("age")));

}

cursor2.close();

//改-谷歌api

ContentValues updateValues = **new** ContentValues();

updateValues.put("name", "zhangsansan");

updateValues.put("age", 233);

database.update("student", updateValues, "name = ?", **new** String[]{"zhangsansan"});

//改-SQLite语句

database.execSQL("update student set age = 666 where name = 'zhangsan'");

//删-谷歌api

database.delete("student", "name = ?", **new** String[]{"lisi"});

//删-SQL语句

database.execSQL("delete from student where name = 'lisi'");

**public** **static** **class** MySQLiteOpenHelper **extends** SQLiteOpenHelper{

**public** MySQLiteOpenHelper(Context context) {

**super**(context

, "info.db"//数据库文件名

, **null**//数据库工厂

, 1//数据库版本

);

}

@Override

**public** **void** onCreate(SQLiteDatabase db) {

//当数据库第一次创建的时候调用

db.execSQL("create table student (\_id integer primary key autoincrement ,name varchar(20) ,age integer)");

}

@Override

**public** **void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {

//当数据库版本变化时进行的操作

}

}

1. 学生管理系统
   1. ListView（重点）。

**class** MyAdapter **extends** BaseAdapter{

@Override

**public** **int** getCount() {

**return** mList.size();

}

@Override

**public** View getView(**int** position, View convertView, ViewGroup parent) {

//返回条目的布局，重点掌握

**if**(convertView == **null**){

convertView = View.*inflate*(MainActivity.**this**, R.layout.***item***, **null**);

}

TextView tv = (TextView) convertView.findViewById(R.id.***tv\_item***);

tv.setText(mList.get(position));

**return** convertView;

}

@Override

**public** Object getItem(**int** position) {//可以不理会

**return** **null**;

}

@Override

**public** **long** getItemId(**int** position) {//可以不理会

**return** 0;

}

}

ListView listView = (ListView) findViewById(R.id.***lv***);

listView.setAdapter(**new** MyAdapter());

1. 网络编程
   1. HttpConnection。
   2. 理解：网络请求必须在子线程进行，只有主线程才能更新UI
   3. 理解Handler的作用，并掌握Handler的用法。
   4. 掌握第三方网络请求框架的用法。

//待续。。