Java基础的介绍

https://www.cnblogs.com/rove888/p/9936109.html

Android 加密解密

https://www.jianshu.com/p/a0cf938ce5ab

concatMap 和 flatMap 最大的区别是 concatMap 发射的数据集是有序的，flatMap 发射的数据集是无序的

不同点： 模块化（lib）主要解决问题是重用问题，不强调单独编译

组件化（lib）主要解决问题是功能拆分，强调单独编译

插件化（application）是所有组件都为apk的特殊组件化，特点可热更新

通讯方式的不同：模块化相互引入，需要引入需要的module

组件化通讯方式分为隐式和路由。

插件化本身是不同进程，因此是binder机制进程间通讯。

onSaveInstanceState()和ViewModel的不同点：

onSaveInstanceState()方法只能保存少量的、能支持序列化的数据，但是onSaveInstanceState()方法可以持久化页面的数据

ViewModel能支持页面中所有的数据，但是需要注意的是，ViewModel不支持数据的持久化，当页面被彻底销毁时，ViewModel及其持有的数据就不存在

ViewModel原理：

1.Activity实现了ViewModelStoreOwner接口，创建了ViewModelStore对象。

2.当Activity意外销毁时，onRetainNonConfigurationInstance函数被回调，在此函数中对ViewModelStore对象进行了保存。

3.当Activity重建时，onCreate方法中会先获取getLastNonConfigurationInstance，如果其中的ViewModelStore对象不为空，就直接引用，不再重新创建ViewModelStore对象了。

ViewModelStore就是个HashMap，通过key来获取ViewModel对象。

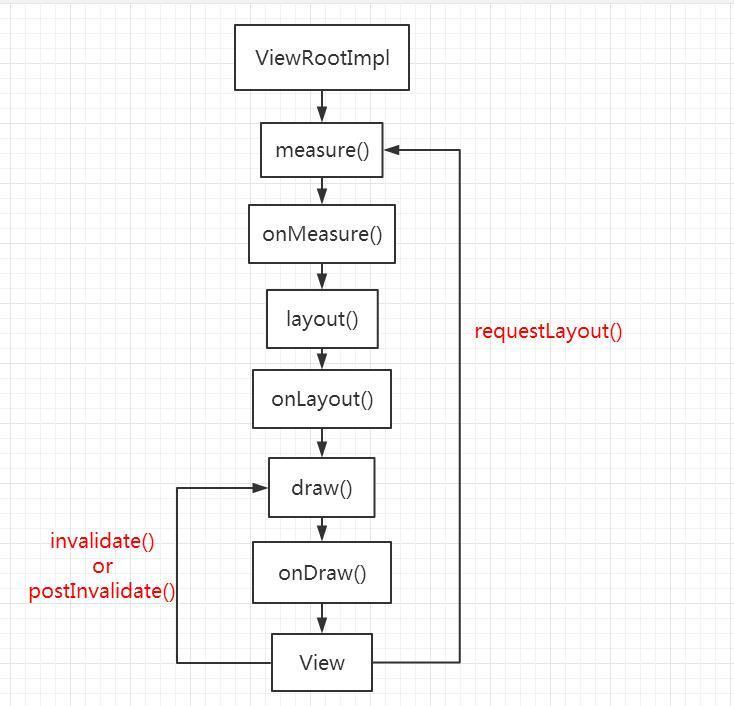
ViewModleProvider的get方法实现也比较简单，总结一下就是：

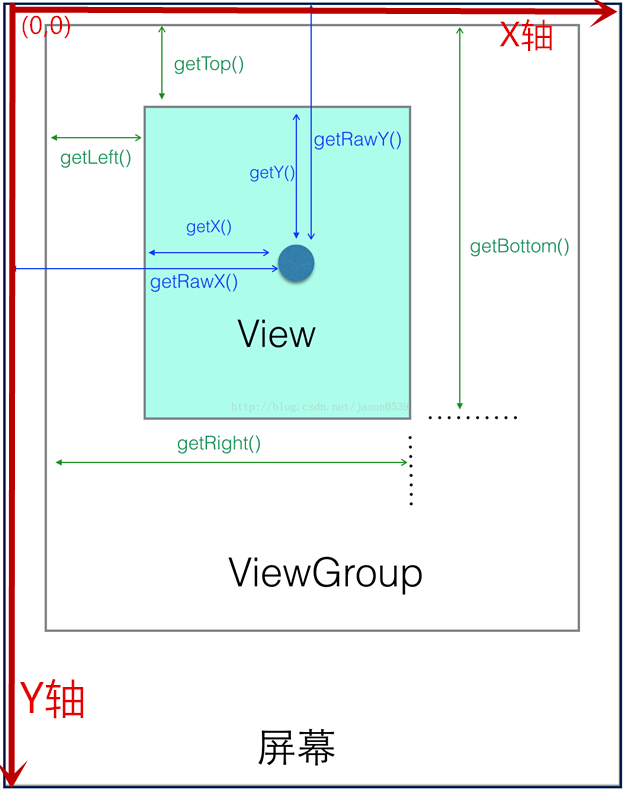
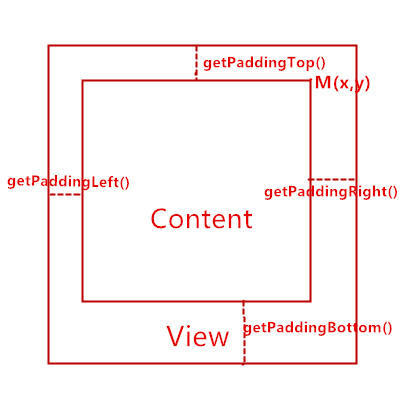
1. 使用ViewModel Class的canonicalName作为ViewModel在ViewModelStore中的唯一标识。
2. 通过唯一标识，先查询一下ViewModelStore中是否有该ViewModel对象，如果有则直接返回。
3. 如果ViewModelStore中没有该ViewModel对象，则通过Factory工厂类反射创建出ViewModel对象，存入ViewModelStore中，并返回给调用者。

invalidate 在UI线程中使用（主线程）。

postInvalidate 在非UI线程中通知重绘。

requestLayout 会标记当前View及父容器，同时逐层向上提交，直到ViewRootImpl处理该事件，ViewRootImpl会调用三大流程，从measure开始，对于每一个含有标记位的view及其子View都会进行测量、布局、绘制。





### 安卓原生与H5混合开发

1.在安卓原生代码中声明

private class JsInterface {  
        // 安卓原生与h5互调方法定义  
        @JavascriptInterface //js接口声明  
        public void takePhoto() {  
          //安卓原生处理逻辑  
        }  
    }

再通过webView控件调用

myWebView.addJavascriptInterface(new JsInterface(), "android");

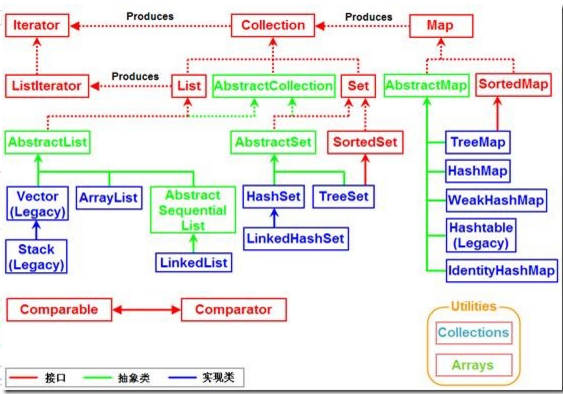
在h5的html中使用window.android.takePhoto();就可以调用上述声明的方法。

1. 安卓原生调用H5方法是

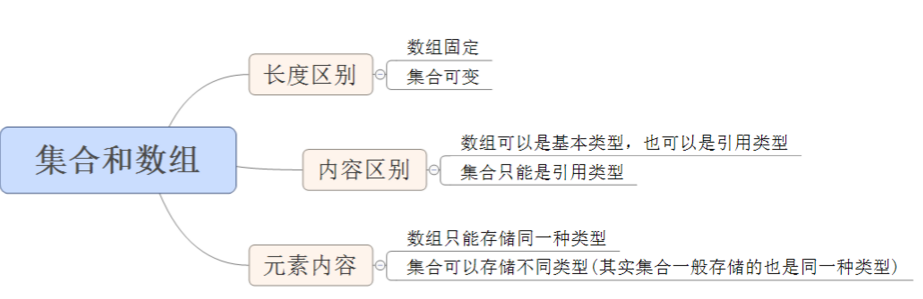
String method = "javascript:testResult('" + content + "')";  
myWebView.loadUrl(method);

通过webview加载，当然testResult()方法需要在html中写好

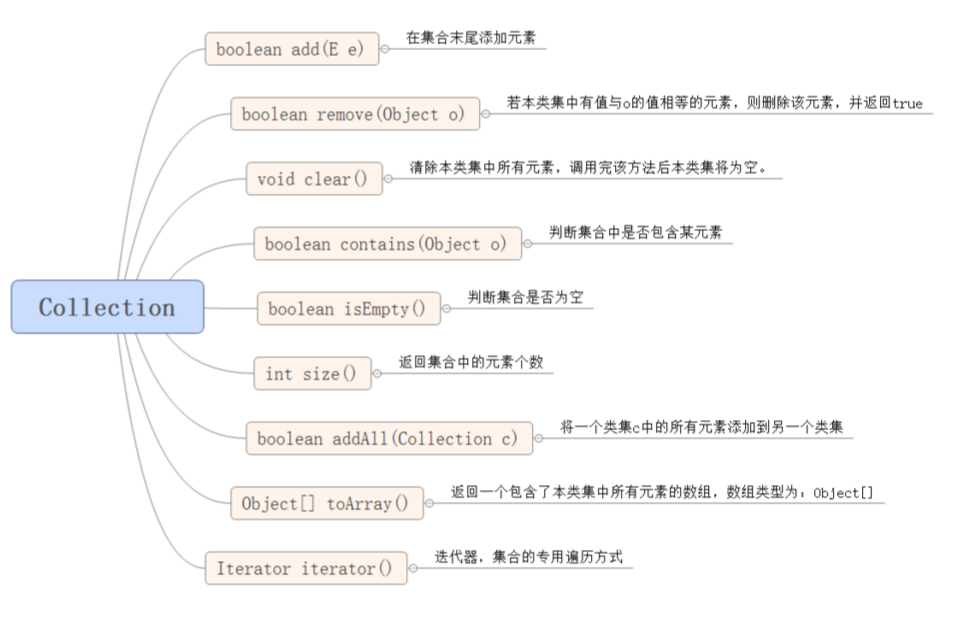
## 数据结构



集合和数组的区别



#### **Collection集合的方法**



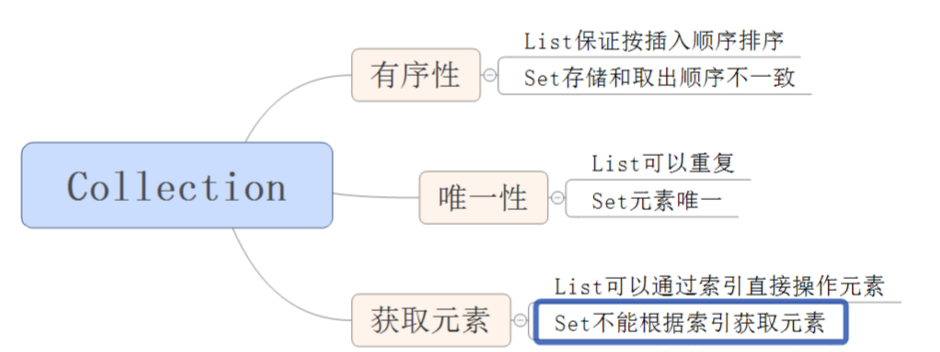
#### 常用集合的分类

**Collection** 接口的接口 对象的集合（单列集合）  
├——-**List** 接口：元素按进入先后有序保存，可重复  
│—————-├ **LinkedList**接口实现类， 链表， 插入删除， 没有同步， 线程不安全  
│—————-├ **ArrayList** 接口实现类， 数组， 随机访问， 没有同步， 线程不安全  
│—————-└ **Vector** 接口实现类 数组， 同步， 线程安全  
│ ———————-└ **Stack** 是Vector类的实现类  
└——-**Set** 接口： 仅接收一次，不可重复，并做内部排序  
├—————-└**HashSet** 使用hash表（数组）存储元素  
│————————└**LinkedHashSet** 链表维护元素的插入次序  
└ —————-**TreeSet** 底层实现为二叉树，元素排好序

**Map** 接口 键值对的集合 （双列集合）  
├———**Hashtable** 接口实现类， 同步， 线程安全  
├———**HashMap** 接口实现类 ，没有同步， 线程不安全-  
│—————–├ **LinkedHashMap** 双向链表和哈希表实现  
│—————–└ **WeakHashMap**  
├ ——–**TreeMap** 红黑树对所有的key进行排序  
└———**IdentifyHashMap**

## **List和Set集合详解**

##### **list和set的区别**



#### List

（1）ArrayList：底层数据结构是数组，查询快，增删慢，线程不安全，效率高，可以存储重复元素  
（2）LinkedList 底层数据结构是链表，查询慢，增删快，线程不安全，效率高，可以存储重复元素  
（3）Vector:底层数据结构是数组，查询快，增删慢，线程安全，效率低，可以存储重复元素