一、原生语言：

苹果手机：ios系统，由swift和c++/object-c语言编写，后缀名为 ipa（ios开发的安装包的后缀名）swift语言是一种开源的语言（半开半闭）

安卓手机：android系统，由java（android）语言编写，后缀名为apk 开源的

wp手机：windowphone系统，由c#语言编写，后缀名分为两类（wp7 wp8的是xap wp8.1以后用8.1的sdk开发的是appx ） 一般说到混合开发不考虑这个版本

序号	系统	语言	后缀名
1	ios	oc/c++/swift	ipa
2	android	android	apk
3	wp	c#	appx

二、混合开发：

原生语言开发+js的开发

缘由：

js无法调动系统的原生功能（拍照、短信、打电话、通讯录），但是原生（android、ios、wp）可以

原生有很大的适配问题（特别是android），js可以很好的解决这个问题

术语：

hybridapp ---- js+ android/ios写的

webapp m站 touch端项目 手机网站 ---- 纯js写的

nativeapp -- 纯android/ios编写的程序

三、android + js的开发模式

准备工具：

java jdk

链接：https://pan.baidu.com/s/1_LsDWTLTa7Hnj3PtRFzkGg 提取码：b2ke

**jdk配置文档 **

https://jingyan.baidu.com/article/6dad5075d1dc40a123e36ea3.html

**android-studio **

intellij编辑器插件扩展而来（intellij在13-14年之前特别火爆，直到现在还被很多人青睐）

链接：https://pan.baidu.com/s/1bFXhgpC2XNJqLdZsmyBLtg 提取码：rzv6

**安装教程 **

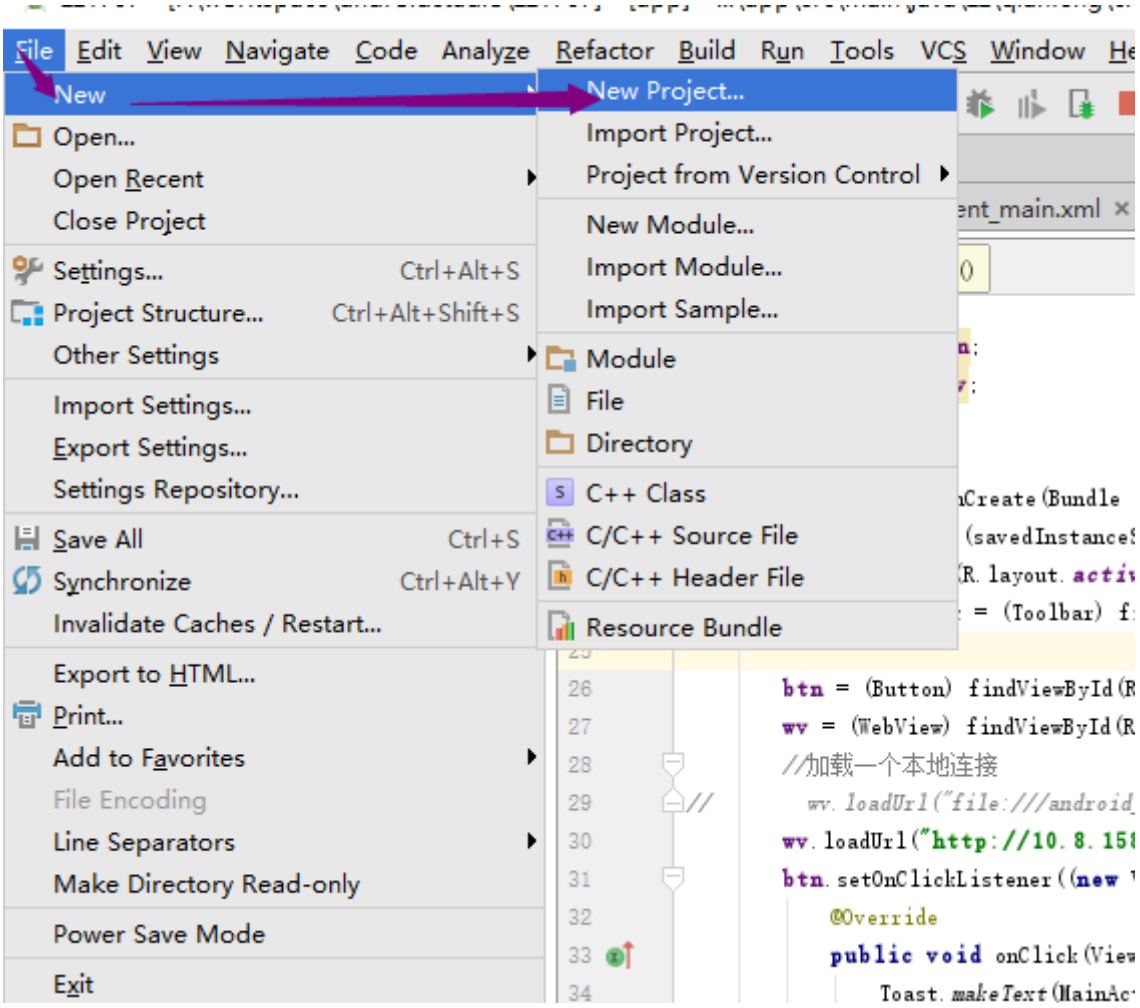
https://www.runoob.com/android/android-studio-install.html

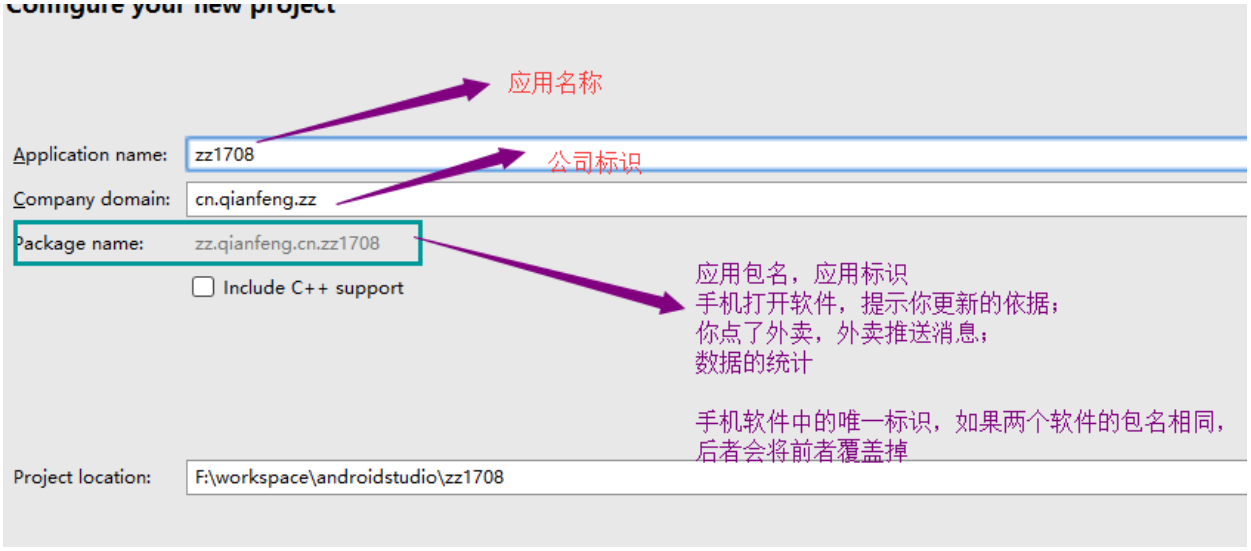
android手机或者android模拟器

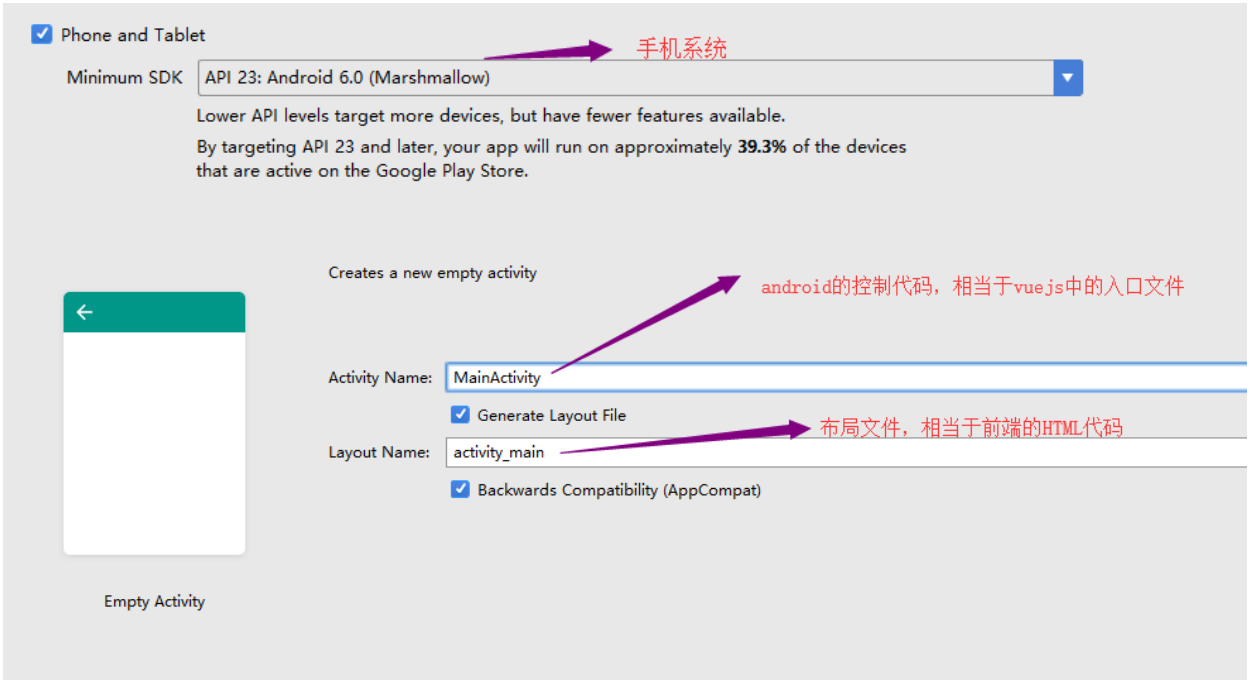
http://droid4x.haimawan.com/

数据线

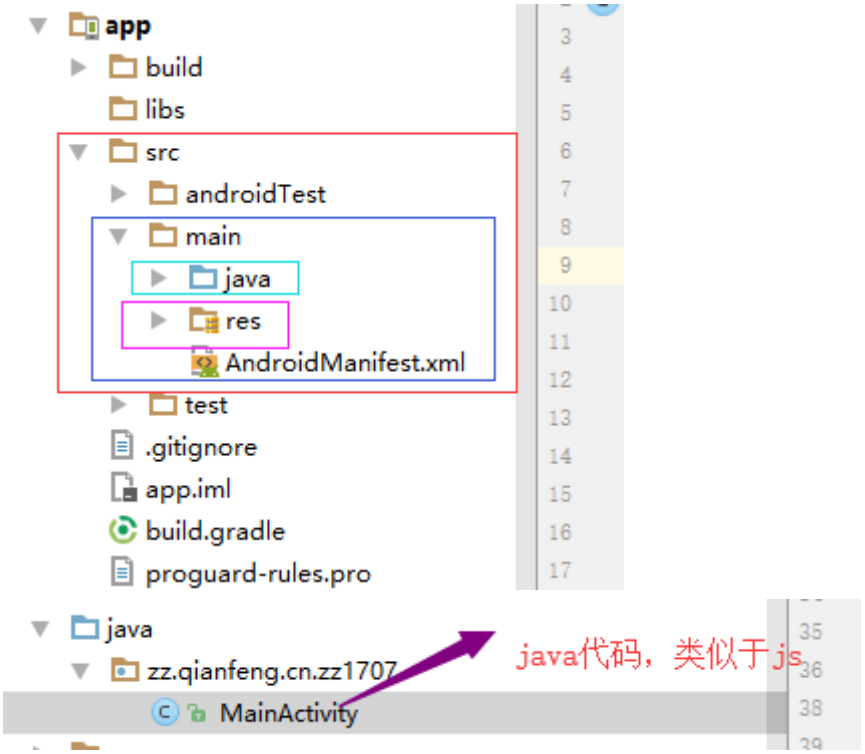
四、代码分析

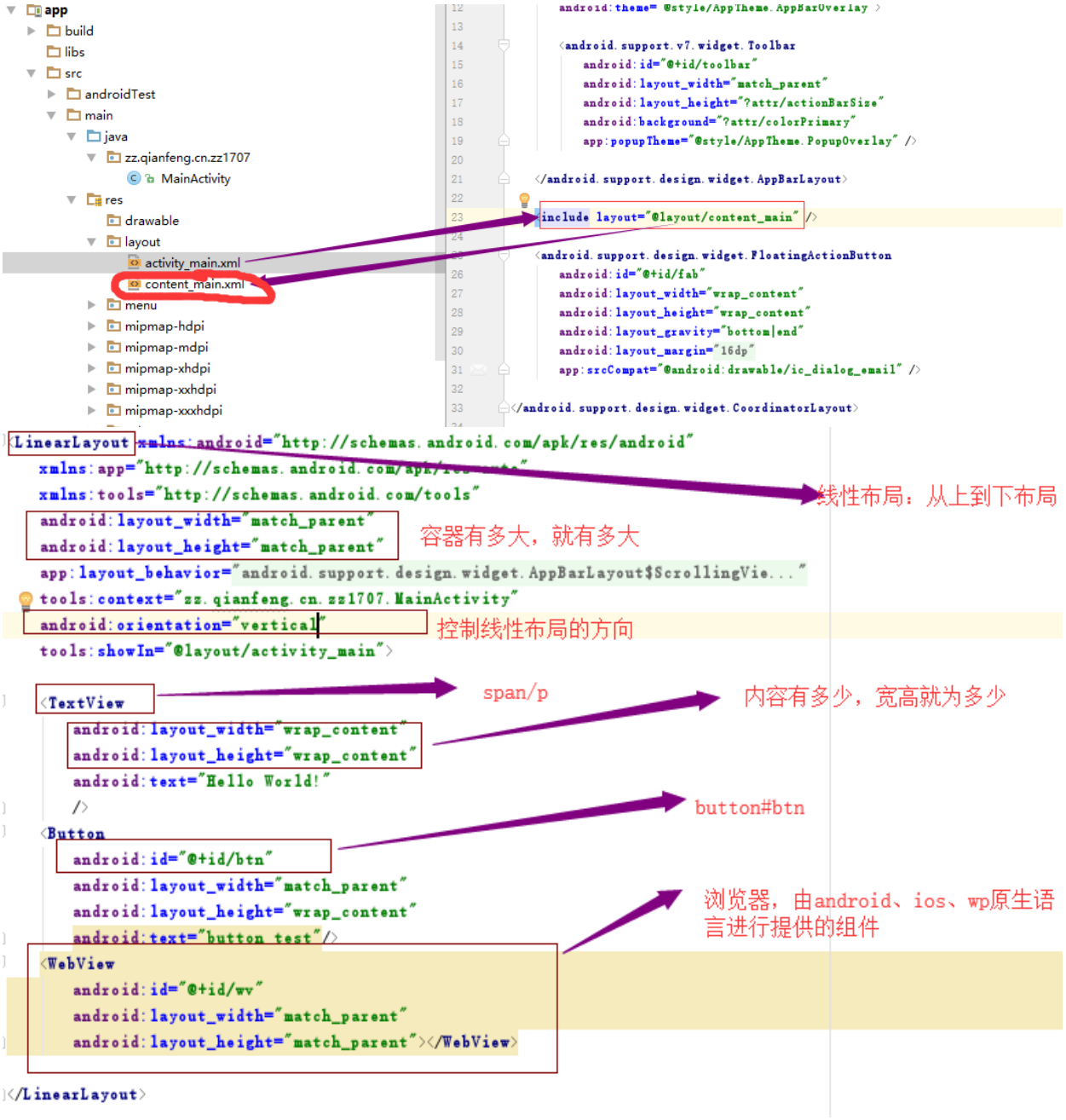

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window He
New
Open...
Open Recent
Close Project
Settings... Ctrl+Alt+S
Project Structure... Ctrl+Alt+Shift+S
Other Settings
Import Settings...
Export Settings...
Settings Repository...
Save All Ctrl+S
Synchronize Ctrl+Alt+Y
Invalidate Caches / Restart...
Export to HTML...
Print...
Add to Favorites
File Encoding
Line Separators
Make Directory Read-only
Power Save Mode
Exit
New Project...
Import Project...
Project from Version Control
New Module...
Import Module...
Import Sample...
Module
File
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Resource Bundle
btn = (Button) findViewById(R
wv = (WebView) findViewById(R
//加载一个本地连接
wv.loadUrl("http://10.8.15
btn.setOnClickListener((new
@Override
public void onClick(View
Toast.makeText(MainAc


应用名称
Application name: zz1708
公司标识
Company domain: cn.qianfeng.zz
Package name: zz.qianfeng.cn.zz1708
Include C++ support
应用包名，应用标识
手机打开软件，提示你更新的依据；
你点了外卖，外卖推送消息；
数据的统计
手机软件中的唯一标识，如果两个软件的包名相同，
后者会将前者覆盖掉
Project location: F:\workspace\androidstudio\zz1708



代码展示为java文件夹下为控制文件，相当于js文件，res文件相当于资源文件以及布局文件




app
build
libs
src
androidTest
main
java
zz.qianfeng.cn.zz1707
MainActivity
res
drawable
layout
activity_main.xml
content_main.xml
menu
mipmap-hdpi
mipmap-mdpi
mipmap-xhdpi
mipmap-xxhdpi
mipmap-xxxhdpi
android:theme="@style/AppTheme.AppBarOverlay"
<android.support.v7.widget.Toolbar
android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="?attr/colorPrimary"
app:popupTheme="@style/AppTheme.PopupOverlay" />
</android.support.design.widget.AppBarLayout>
<include layout="@layout/content_main" />
<android.support.design.widget.FloatingActionButton
android:id="@+id/fab"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="bottom|end"
android:layout_margin="16dp"
app:srcCompat="@android:drawable/ic_dialog_email" />
</android.support.design.widget.CoordinatorLayout>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingVie..."
tools:context="zz.qianfeng.cn.zz1707.MainActivity"
android:orientation="vertical"
tools:showIn="@layout/activity_main">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
/>
<Button
android:id="@+id/btn"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="button test"/>
<WebView
android:id="@+id/wv"
android:layout_width="match_parent"
android:layout_height="match_parent"></WebView>
</LinearLayout>
线性布局：从上到下布局
容器有多大，就有多大
控制线性布局的方向
span/p
内容有多少，宽高就为多少
button#btn
浏览器，由android、ios、wp原生语言进行提供的组件

如需需要给页面添加一个网页

```
wv.loadUrl("file:///android_asset/index.html");   ///加载本地的页面        ----- 代码在手机，更新需要下载替换
wv.loadUrl("https://h5.m.taobao.com/#index"); /// 加载远程的页面   -----   更新无需下载
```

默认的webviw浏览器是纯洁的，不支持js

```
/**
* webView目前还不具备真正浏览器的功能
*/
webView.getSettings().setJavaScriptEnabled(true);//相当于google浏览器中的内容设置中的js的设置
```

//默认情况下，点击webview中的链接，会使用Android系统自带的浏览器打开这个链接。

//如果希望点击链接继续在我们自己的Browser中响应，必须覆盖 webview的WebViewClient对象：

//新建两个对象MWebViewClient 和 MWebChromeClient，他们分别继承自WebViewClient和WebChromeClient /* Android WebView 做为承载网页的载体控件，他在网页显示的过程中会产生一些事件， 并回调给我们的应用程序，以便我们在网页加载过程中做应用程序想处理的事情。 比如说客户端需要显示网页加载的进度、网页加载发生错误等等事件。

WebView提供两个事件回调类给应用层，分别为WebViewClient,WebChromeClient开发者可以继承这两个类，接手相应事件处理。

WebViewClient 主要提供网页加载各个阶段的通知，比如网页开始加载onPageStarted，网页结束加载onPageFinished等； WebChromeClient主要提供网页加载过程中提供的数据内容，比如返回网页的title,favicon等。*/

```
//处理页面加载各个阶段
MWebViewClient mWebViewClient = new MWebViewClient(webView, getApplicationContext());
webView.setWebViewClient(mWebViewClient);
//提供网页加载过程中提供的数据内容
MWebChromeClient mWebChromeClient = new MWebChromeClient( getApplicationContext());
webView.setWebChromeClient(mWebChromeClient);
```

```
MWebViewClient.java
public class MWebViewClient extends WebViewClient {
     private WebView webView;
      private Context context;

      public MWebViewClient(WebView webView) {
        super();
        this.webView = webView;
      }

      public MWebViewClient(WebView webView, Context context) {
        super();
        this.webView = webView;
        this.context = context;
      }


      @Override
      public boolean shouldOverrideUrlLoading(WebView view, String url) {
        webView.loadUrl(url);
        return true;
      }

      @Override
      public void onPageStarted(WebView view, String url, Bitmap favicon) {
        super.onPageStarted(view, url, favicon);
      }
```

```
      @Override
      public void onPageFinished(WebView view, String url) {
        super.onPageFinished(view, url);
      }


      @Override
      public void onLoadResource(WebView view, String url) {
        super.onLoadResource(view, url);
      }

      @Override
      public void onReceivedSslError(WebView view,
          SslErrorHandler handler, SslError error) {
        super.onReceivedSslError(view, handler, error);
      }
}
```

```
MWebChromeClient.java

public class MWebChromeClient extends WebChromeClient {
    private Context context;

      public MWebChromeClient(Context context) {
        super();
        this.context = context;
      }

      @Override
      public boolean onJsAlert(WebView view, String url, String message,
          JsResult result) {
        return super.onJsAlert(view, url, message, result);
      }

      @Override
      public void onReceivedTitle(WebView view, String title) {
        super.onReceivedTitle(view, title);
      }
      @Override
      public boolean onJsConfirm(WebView view, String url, String message,
          JsResult result) {
        return super.onJsConfirm(view, url, message, result);
      }
      @Override
      public boolean onJsPrompt(WebView view, String url, String message,
          String defaultValue, JsPromptResult result) {
        return super.onJsPrompt(view, url, message, defaultValue, result);
      }
}
```

假设你现在需要调用系统的拍照功能，android工程师会为你提供一个对象和他对应的方法WebViewTakePhoto为对象，takePhoto为拍照的方法，

```
webView.addJavascriptInterface(new MJavascriptInterface1(getApplicationContext()),
"WebViewTakePhoto");
```

通过这样的形式将地址传递给前端，testParams方法由前端定义

webView.loadUrl("javascript:testParams(http://img4.imgtn.bdimg.com/it/u=103061881,2842093305&fm=26&gp=0.jpg')");

```
MJavascriptInterface1
class MJavascriptInterface1{
    private Context context;


    public MJavascriptInterface1(Context context) {
        // TODO Auto-generated constructor stub
        this.context = context;
    }
    @JavascriptInterface
    public void takePhoto(){
        Toast.makeText(context, "拍照", Toast.LENGTH_SHORT).show();
        //android调用拍照功能

//
webView.loadUrl("javascript:testParams('http://img4.imgtn.bdimg.com/it/u=103061881,2842093305&fm=26&gp=0.jpg')");
    }

}
```

前端调用拍照方法，在自己的写的触发函数中

```
window.WebViewTakePhoto.takePhoto()
```

说： webview --- android、ios的组件 ---- js运行的环境

进行配置

```
---  支持js

---  链接本组件内跳转
```

根据需求先写页面，假设你的一个按钮需要调用系统的功能，在你的按钮事件中调用由android工程师提供的相对应的对象和其方法即可，同时前端也会定义一些方法，但是前端自己不调用，android会根据webview的loadUrl方法进行调用，并且传参（根据需求看）

百度网盘链接地址

四、IOS + js的混合开发

参考百度网盘观看

链接：https://pan.baidu.com/s/19fZvDn2NWjoLHTa2L09MRQ 提取码：ogvx