[**Android webview使用详解**](http://www.cnblogs.com/zgz345/p/3768174.html)

1. 打开网页时不调用系统浏览器， 而是在本WebView中显示：

[复制代码](javascript:void(0);)

mWebView.setWebViewClient(new WebViewClient(){

@Override

public boolean shouldOverrideUrlLoading(WebView view, String url) {

view.loadUrl(url);

return true;

}

});

[复制代码](javascript:void(0);)

2. 通过java代码调用javascript

[复制代码](javascript:void(0);)

WebSettings webSettings = mWebView .getSettings();

webSettings.setJavaScriptEnabled(true);

mWebView.addJavascriptInterface(new Object() {

public void clickOnAndroid() {

mHandler.post(new Runnable() {

public void run() {

webview.loadUrl("javascript:wave()");

}

});

}

}, "demo");

[复制代码](javascript:void(0);)

3. 按返回键时， 不退出程序而是返回上一浏览页面：

[复制代码](javascript:void(0);)

public boolean onKeyDown(int keyCode, KeyEvent event) {

if ((keyCode == KeyEvent.KEYCODE\_BACK) && mWebView .canGoBack()) {

webview.goBack();

return true;

}

return super.onKeyDown(keyCode, event);

}

[复制代码](javascript:void(0);)

4. 打开页面时， 自适应屏幕：

WebSettings webSettings = mWebView .getSettings();

webSettings.setUseWideViewPort(true);//设置此属性，可任意比例缩放

webSettings.setLoadWithOverviewMode(true);

5. 便页面支持缩放：

WebSettings webSettings = mWebView .getSettings();

webSettings.setJavaScriptEnabled(true);

webSettings.setBuiltInZoomControls(true);

webSettings.setSupportZoom(true);

6.如果webView中需要用户手动输入用户名、密码或其他，则webview必须设置支持获取手势焦点。

webview.requestFocusFromTouch();

7.WebView 加载界面主要调用三个方法：LoadUrl、LoadData、LoadDataWithBaseURL.

1、LoadUrl            直接加载网页、图片并显示.（本地或是网络上的网页、图片、gif）

2、LoadData           显示文字与图片内容 （模拟器1.5、1.6）

3、LoadDataWithBase  显示文字与图片内容（支持多个模拟器版本）

8.WebSettings 的常用方法介绍

[复制代码](javascript:void(0);)

setJavaScriptEnabled(true);  //支持js

setPluginsEnabled(true);  //支持插件 

setUseWideViewPort(false);  //将图片调整到适合webview的大小 

setSupportZoom(true);  //支持缩放 

setLayoutAlgorithm(LayoutAlgorithm.SINGLE\_COLUMN); //支持内容重新布局  

supportMultipleWindows();  //多窗口 

setCacheMode(WebSettings.LOAD\_CACHE\_ELSE\_NETWORK);  //关闭webview中缓存 

setAllowFileAccess(true);  //设置可以访问文件 

setNeedInitialFocus(true); //当webview调用requestFocus时为webview设置节点  
  
webview webSettings.setBuiltInZoomControls(true); //设置支持缩放 

setJavaScriptCanOpenWindowsAutomatically(true); //支持通过JS打开新窗口   
  
setLoadWithOverviewMode(true); // 缩放至屏幕的大小

setLoadsImagesAutomatically(true);  //支持自动加载图片

[复制代码](javascript:void(0);)

9.WebViewClient 的方法全解

[复制代码](javascript:void(0);)

doUpdateVisitedHistory(WebView view, String url, boolean isReload)  //(更新历史记录)

onFormResubmission(WebView view, Message dontResend, Message resend) //(应用程序重新请求网页数据)

onLoadResource(WebView view, String url) // 在加载页面资源时会调用，每一个资源（比如图片）的加载都会调用一次。

onPageStarted(WebView view, String url, Bitmap favicon) //这个事件就是开始载入页面调用的，通常我们可以在这设定一个loading的页面，告诉用户程序在等待网络响应。 

onPageFinished(WebView view, String url) //在页面加载结束时调用。同样道理，我们知道一个页面载入完成，于是我们可以关闭loading 条，切换程序动作。

onReceivedError(WebView view, int errorCode, String description, String failingUrl)// (报告错误信息)

onReceivedHttpAuthRequest(WebView view, HttpAuthHandler handler, String host,String realm)//（获取返回信息授权请求）

onReceivedSslError(WebView view, SslErrorHandler handler, SslError error) //重写此方法可以让webview处理https请求。

onScaleChanged(WebView view, float oldScale, float newScale) // (WebView发生改变时调用)

onUnhandledKeyEvent(WebView view, KeyEvent event) //（Key事件未被加载时调用）

shouldOverrideKeyEvent(WebView view, KeyEvent event)//重写此方法才能够处理在浏览器中的按键事件。

shouldOverrideUrlLoading(WebView view, String url)

//在点击请求的是链接是才会调用，重写此方法返回true表明点击网页里面的链接还是在当前的webview里跳转，不跳到浏览器那边。这个函数我们可以做很多操作，比如我们读取到某些特殊的URL，于是就可以不打开地址，取消这个操作，进行预先定义的其他操作，这对一个程序是非常必要的。