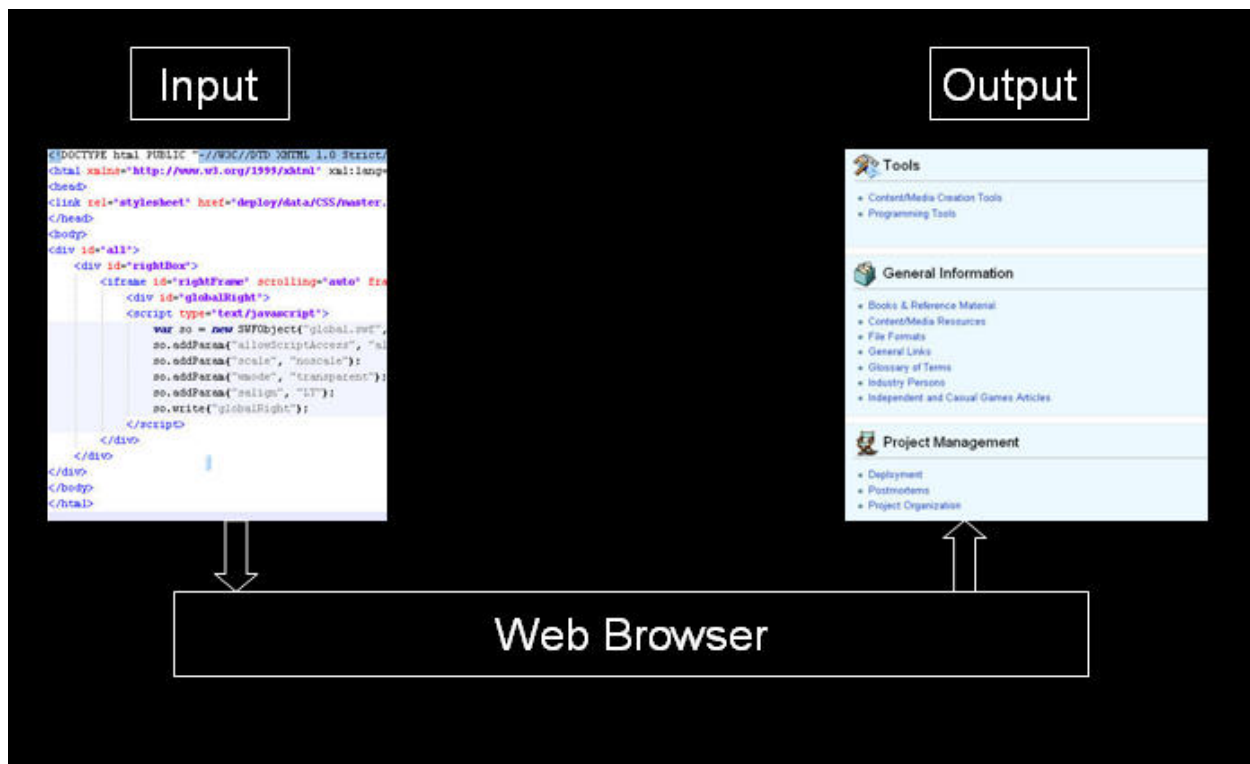


什么是Webkit

Webkit是一个开源的Web浏览器引擎，也就是浏览器的内核。Apple的Safari, Google的Chrome, Nokia S60平台的默认浏览器，Apple手机的默认浏览器，Android手机的默认浏览器均采用的Webkit作为浏览器内核。Webkit的采用程度由此可见一斑，理所当然的成为了当今主流的三大浏览器内核之一。另外两个分别是Gecko和Trident，大名鼎鼎的Firefox便是使用的Gecko 内核，而微软的IE系列则使用的是Trident内核。虽然目前市场上IE内核浏览器仍然占据大多数市场份额，但其他内核浏览器也大有后来居上之势。其中,Webkit作为一款优秀的浏览器内核，它众多优秀的特性引起业内的广泛关注。尤其是近来，google的加入更是让Webkit有所升温，从 Goole Chrome浏览器, Goole Anroid手机操作系统内置浏览器均采用Webkit作为内核， 近来推出的Chrome OS更是让人期待

Webkit做了什么？

作为浏览器的内核，Webkit做了哪些工作？为了了解这些，先让我们来看下一个Web浏览器究竟做了什么。我们可以从输入输出的角度来看一个 Web浏览器为我们做了哪些工作。先看一个简单的例子，Web浏览器的输入是一个HTML文档，输出则是一个我们用眼睛所看到的一个Web页面, 就普通用户而言它的输入和输出就是这么简单，如下图所示。



那么Webkit的输入和输出又是什么呢？如果能明白这个，那我们就能很清楚的知道Webkit到底是做什么的了。不过现在要说清楚这个还有点困难，因为Webkit的输出就要复杂些了，因为它的输出本来就非直接面向用户，现在简单点来说，Webkit的输入是web 文档，输出是一些看不见的模型，浏览器上层借助于这些模型来绘制出我们所看到的实际页面。后面适当的时候会更具体的探讨这些模型

Webkit组成

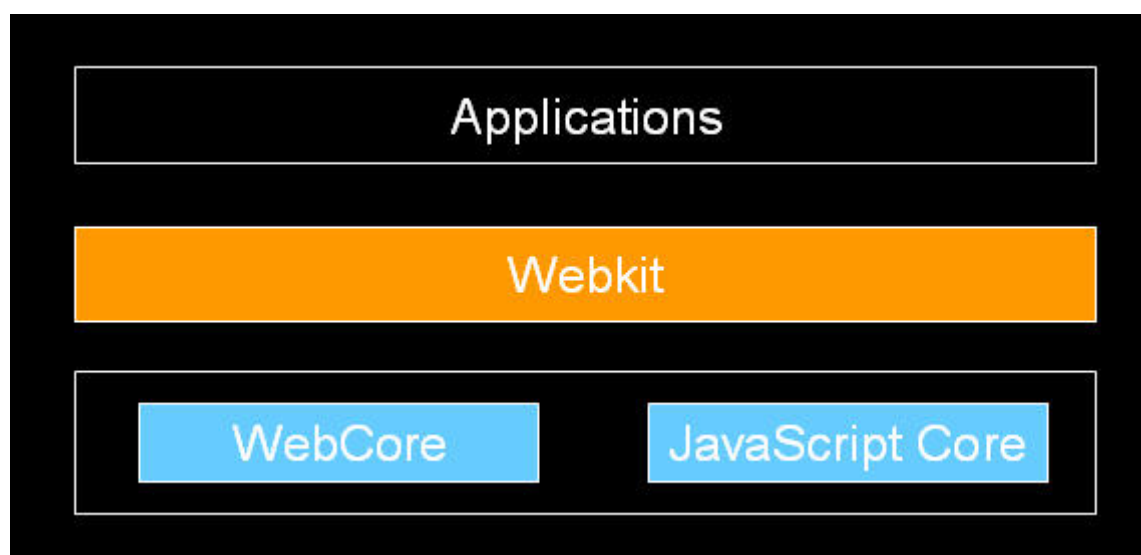
Webkit实际上包含三大部分，至少从代码结构上来说是这样的，当然，如果细分的话还能够划分出更多的模块。如下图所示



其中，WebCore是Webkit的核心部分，它实现了对文档的模型化，包括了CSS, DOM, Render等的实现，JavaScript Core显然是对JavaScript支持的实现。而橘黄色标注的Webkit部分包含了很多不同平台对Webkit封装的实现，即抽象出了与浏览器所能直接对应的一些概念的实现，如，WebView, WebPage, WebFrame等。这三部分共同构成了Webkit，在源码中，它们分别对应这个这三个目录，即Webkit三大部分为WebCore, JavaScript Core, Webkit

应用程序如何利用Webkit的

从下图可以看出，利用Webkit的应用程序的位置处于Webkit模块之上，应用程序并不直接和WebCore以及JavaScript Core来打交道，也不需要和它们打交道，而是同我前面提到过的Webkit模块来交互，从而屏蔽了WebCore和JavaScript Core部分。Webkit模块实际上抽象出了大多数应用程序所需要的那一部分，这里所说的大多数应用程序其实就是指浏览器，对于实现一个浏览器而言，Webkit模块所提供的接口已经足够。不过对于某些应用，可能得对其实施一定的改造，但是话又说回来，Webkit本来就是被设计成为Browser的内核而被实现的。



看到这里，也就是说如果我们要想利用Webkit来实现一个功能完善Browser或其他类浏览器应用程序的话，你只需要了解Webkit模块部分就足够了，更确切的说只需要了解它所提供的接口就够了。事实上，大多数平台，如果采用了Webkit内核作为浏览器内核组件的话，所提供的对应文档部分都是关于Webkit模块所提供的接口的描述。很多平台都提供了Browser组件，其使用方法就像是其他可视组件一

样，在应用程序中应用起来非常方便。比如，某平台提供了叫做WebView的可视化组件，我们就可以通过如下几行代码来展示一个网页，使用的代码仅仅三行。

```
WebView *view = new WebView();  
view->load(Url("http://www.google.cn"));  
view->show();
```