

浏览器应该有的功能

网络：

浏览器通过网络模块来下载各式各样的资源，例如html文本；javascript代码；样式表；图片；音视频文件等。

网络部分尤为重要，因为它耗时长，而且需要安全访问互联网上的资源。

资源管理：

从网络下载，或者本地获取到的资源需要有高效的机制来管理它们。

例如如何避免重复下载，资源如何缓存等等

网页浏览：

这是浏览器的核心也是最基本的功能，最重要的功能。

如何将资源转变为可视化的结果。

多页面管理

插件与管理

账户和同步

安全机制

开发者工具

...

...

浏览器的主要功能总结起来就是一句话：将用户输入的url转变成可视化的图像。

浏览器的内核(渲染引擎)

在浏览器中有一个最重要的模块，它主要的作用把一切请求回来的资源变为可视化的图像。

这个模块就是浏览器内核，通常它也被称为渲染引擎。

浏览器内核总结：

IE----->Trident

Safari----->WebKit

WebKit本身主要是由两个小引擎构成的，

一个正是渲染引擎“WebCore”，

另一个则是javascript解释引擎“JSCore”，

它们均是从KDE的渲染引擎KHTML及javascript解释引擎KJS衍生而来。

Chrome----->WebKit的分支引擎----->Blink

在13年发布的Chrome 28.0.1469.0版本开始，Chrome放弃Chromium引擎转

而使用最新的Blink引擎（基于WebKit2---苹果公司于2010年推出的新的WebKit引擎），

Blink对比上一代的引擎精简了代码、改善了DOM框架，也提升了安全性。

Opera

旧版Opera 4至6版本 :Elektra排版引擎

Opera7.0 :Presto渲染引擎

Opera在2013年2月宣布放弃Presto:

采用Chromium引擎；

又转为Blink引擎；

Firefox----->Gecko

进程与线程

进程：程序的一次执行，它占有一片独有的内存空间，是操作系统执行的基本单元。

一个进程中至少有一个运行的线程：主线程，进程启动后自动创建

一个进程中也可以同时运行多个线程，我们会说程序是多线程运行的

一个进程内的数据可以供其中的多个线程直接共享，多个进程之间的数据是不能直接共享的

线程：是进程内的一个独立执行单元，是CPU调度的最小单元。程序运行的基本单元

线程池(thread pool)：保存多个线程对象的容器，实现线程对象的反复利用

JS引擎是单线程运行的！（回忆事件轮询机制）

现代浏览器：多进程、多线程模型

1. 不堪回首的过去：

当你通过浏览器打开很多页面的时候，如果其中一个页面不响应了或者崩溃了，那么随之而来的将会是更不幸的事情，你开打的所有页面都会得不到响应，最让人不能忍受的是，其中的一些页面可能还包含了未保存或者未发送的信息

2. 浏览器厂商如何解决？

采用多进程模型，该模型可以带来的好处

- ①. 避免因单个页面的不响应或者崩溃影响整个浏览器的稳定性
- ②. 当第三方插件崩溃时，也不会影响整个浏览器的稳定性
- ③. 安全

3. 浏览器到底有些什么进程

①. Browser进程：

浏览器的主进程，负责浏览器界面的显示，和各个页面的管理，
浏览器中所有其他类型进程的祖先，负责其他进程的创建和销毁
它有且只有一个！！！！

②. Renderer进程：

网页渲染进程，负责页面的渲染，可以有多个
当然渲染进程的数量不一定等于你开打网页的个数

③. 各种插件进程

④. GPU进程

移动设备的浏览器可能不太一样：

Android不支持插件，所以就没有插件进程

GPU演化成了Browser进程的一个线程

Renderer进程演化成了操作系统的一个服务进程，它仍然是独立的

4. 每个进程内部又有很多线程

多线程的目的主要是保持用户界面的高度响应

例如：为了不让Browser进程的UI线程被其他耗时的操作（大文件的加载，本地文件读写）所阻塞，
那么我们就把这些操作放到分线程中去处理。

在Renderer进程中，为了不让其他操作阻止渲染线程的高速执行，我们通常会将渲染过程【管线化】，
利用计算机的多核优势，让渲染的不同阶段在不同的线程中执行