浏览器的工作原理

1. 浏览器的主要功能：将用户的web资源呈现出来，需要向服务器请求资源，并将其显示在浏览器的窗口中。资源的格式通常是HTML、pdf、image或其他格式。

用户用URI（统一资源标识符）来规定所求资源的位置。

HTML和CSS规范了浏览器解释html文档的方式。

浏览器的用户界面：

用户输入URL的地址栏

前进、后退按钮

刷新按钮

收藏、书签按钮

主页按钮

1. 浏览器的主要构成
2. 用户界面：地址栏、前进后退、刷新、书签、主页。
3. 浏览器引擎：在用户界面和渲染引擎之间传送命令。
4. 渲染引擎：显示内容，如果呈现的是HTML内容，就负责解析HTMl和CSS内容，将解析的内容呈现在屏幕上。
5. 网络：网络调用，比如Http请求
6. 用户界面后端：用于绘制基本的窗口小组件。
7. JavaScript解释器：用于解析和执行JavaScript代码。
8. 数据存储：持久层，浏览器需要在硬件保存数据（cookie）。
9. 渲染引擎

渲染引擎的职责：在浏览器窗口显示所请求的内容。

可以显示html、xml及图片，也可以借助插件，显示其他类型数据。

开源浏览器：IE、Firefox、Safari、Chrome

渲染引擎：Firefox Geoko

Safari和Chrome Webkit

1. 渲染的流程

渲染引擎首先通过网络获得请求的内容，通常以8k分块的方式完成。

取得内容之后流程：

解析html构建dom树 》构建render树（渲染）》布局render树》绘制render树

1. 渲染引擎开始解析html，并将标签转化为内容树中的dom节点。它解析外部css文件及style标签中的样式信息。
2. 样式信息以及html中的可见性指令被用来构建另一颗——render树
3. 构建render树：Render树由一些包含颜色、大小等属性的矩形组成，它们将正确的顺序显示在屏幕上。
4. 布局render树：会确定每个节点在屏幕上的确切坐标。
5. 绘制render树：遍历render树，并使用UI后端层绘制每个节点。

注意：渲染引擎不会等所有的html内容解析完才构建和布局render树。它是解析完一部分内容就显示一部分内容，同时还在通过网络下载其余内容。

1. 解析

解析一个文档即将其转换为具有一定意义的结构——编码可以理解和使用的东西。

解析的结果通常是表达文档结构的节点树，称为解析树或语法树。

1. 文法

解析基于文档依据的语法规则——文档的语言或格式。

每种可被解析的格式必须具有由词汇及语法规则组成的特定的文法，称为上下文无关文法。

1. 解析器——词法分析器

解析分为两个过程——语法分析及词法分析

词法分析：将输入分解为符号，符号是语言的词汇表——基本有效单元的集合。（相当于单词）

语法分析：对语言应用语法规则。

解析器一般将工作分配给两个组件——词法分析器负责将输入分解成合法的符号，解析器则根据语言的语法规则分析文档结构，从而构建解析器，词法解析器知道怎么跳过空白和换行之类的无关字符。

解析过程是迭代的，解析器从词法分析器处得到一个新的符号，并试着用这个符号匹配一条语法规则，如果匹配了一条语法规则，这个符号对应的节点将被添加到解析树上，然后解析器去请求另一个符号。如果没有匹配到规则，解析器将会在内部保存该符号，并从词法解析器去下一个符号，直到所有内部保存的符号能够匹配一项语法规则。如果最终没有找到匹配的规则，解析器将会跑出异常，这就意味着文档无效或是包含语法错误。

1. 转换：解析树并不是最终结果。解析一般在转换中使用——将输入文档转换成另一种格式。编译就是个例子编译将一段源码编译为机器码的时候，先将源码解析为解析树，然后将该树转换为一个机器码文档。
2. 解析实例

2+3-1

词汇表：包括整数、加号和减号

语法：1.该语言的语法基本单元包括表达式、term及操作符

2.语言可以包括多个表达式

3.一个表达式定义为两个term通过一个操作符连接

4．操作符可以是加号或减号

5.term可以是一个整数或一个表达式

第一个匹配规则的子字符串是‘2’，根据规则5，它是一个term，第二个匹配的是‘2+3’，符合规则3。‘2+3-1’是一个表达式，符合规则3，因为‘2+3’是一个term。

10、词汇表及语法的定义

词汇表通常利用正则表达式来定义

INTEGER：0|[1-9][0-9]\*

PLUS：+

MINUS：-

语法通常是BNF格式定义，我们的语言可以定义为：

expression：= term operation term

operation：= PLUS| MINUS

term：= INTEGER | expression

如果一个语言的文法是上下文无关的，则它可以用正则解析器来解析。对上下文无关文法的一种定义是，该文法可以用BNF来完整的表达。

11、解析器类型

两种：自顶向下解析及自底向上解析。

自顶向下解析：查看语法的最高层结构并试着匹配其中一个；

自底向上解析：从输入开始，逐步将其转换为语法规则，从底层规则开始直到匹配高层规则。

12、自动化解析

解析器生成器这个工具可以自动生成解析器，只需要指定语言的文法——词汇表及语法规则，它就可以生成一个解析器。创建一个解析器需要对解析有深入的理解，而且手动的创建一个由较好性能的解析器并不容易，所以解析生成器很有用。Webkit使用两个知名的解析生成器——用于创建语法分析器的Flex及创建解析器的Bison。Flex的输入是一个包含了符合定义的正则表达式，Bison的输入是用BNF格式表示的语法规则。