

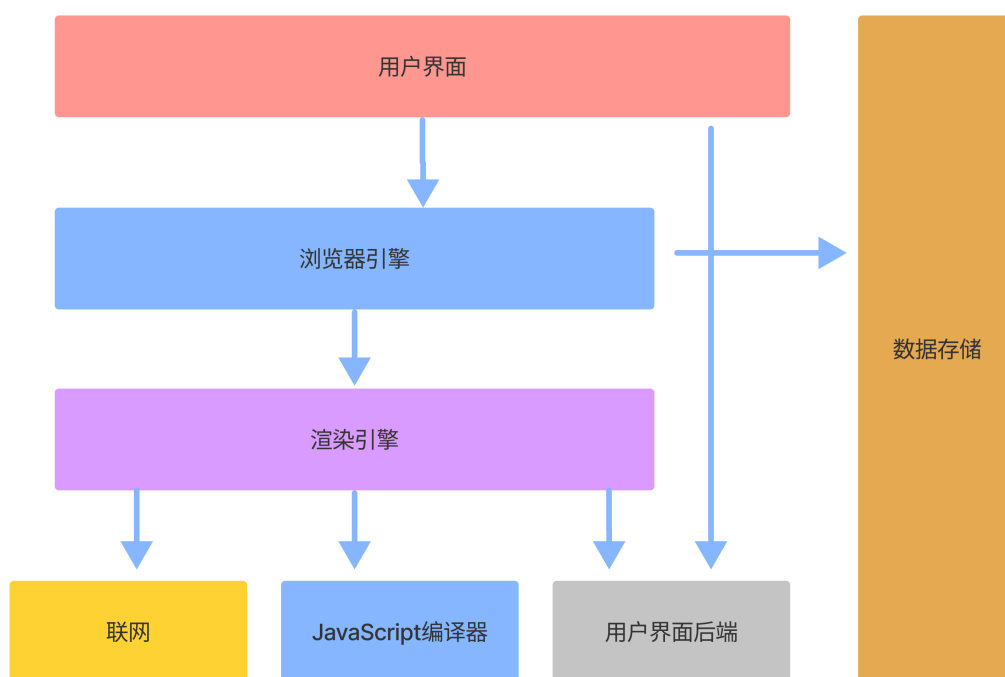
# 了解渲染引擎在浏览器中的作用

BrowserStack 技术内容作家-Jash Unadkat-2019年11月15日

在深入探讨渲染引擎在浏览器中的作用之前，让我们快速了解Web浏览器的底层架构。

## 了解Web浏览器的架构

Web浏览器是一种软件应用程序，它使用户能够通过图形用户界面访问和显示网页或其他在线内容。通过下图了解构建Web浏览器所涉及的关键组件。



图形：浏览器组件

- **用户界面:** 这个组件允许终端用户与网页上所有可用的视觉元素进行交互。视觉元素包括 [地址栏](#)、[主页按钮](#)、[下一步按钮](#) 和所有其他获取和显示终端用户所请求的网页的元素。
- **浏览器引擎:** 它是每个Web浏览器的核心组件。浏览器引擎充当用户界面和渲染引擎之间的中介或桥梁。它根据用户界面接收到的输入，[查询](#)和[处理](#)渲染引擎。
- **渲染引擎:** 顾名思义，该组件负责在屏幕上呈现用户请求的特定网页。它解释HTML和XML文档以及使用CSS设置样式或格式化的图像，并生成最终布局，显示在用户界面上。

**注意:** 每个浏览器都有自己独特的渲染引擎。不同浏览器版本的渲染引擎也可能不同。下面的列表提到了一些常见浏览器使用的浏览器引擎。

1. 谷歌浏览器和欧朋浏览器v.15+: **Blink**
2. ie浏览器: **Trident**
3. 火狐浏览器: **Gecko**
4. iOS端谷歌浏览器和苹果浏览器: **Webkit**

- **联网:** 该组件负责管理使用标准协议（如HTTP或FTP）的网络调用。它还负责处理与互联网通信相关的安全问题。

- **JavaScript解释器:** 顾名思义，她负责解析和执行嵌入网站中的JavaScript代码。一旦生成了解释结果，它们就会被转发到渲染引擎以显示在用户界面上。
- **用户界面后端:** 该界面使用底层操作系统的用户界面方法。它主要用于绘制基本小部件（窗口和组合框）。
- **数据存储:** 它是一个持久层。Web浏览器需要在本地存储各种类型的数据。历史cookie。因此，浏览器必须兼容WebSQL、IndexedDB、FileSystem等数据存储机制。

现在我们已经了解了构建Web浏览器所涉及的关键组件，让我们来深入了解渲染引擎的角色。

## 渲染引擎的角色：

一旦用户请求特定文档，渲染引擎就会开始获取请求文档的内容。这是通过网络层完成的。渲染引擎开始以8KB每秒的数据块从网络层接收该特定文档的内容。之后渲染引擎的基本流程就开始了。



图形：渲染引擎基本流程

这四个基本步骤：

1. 请求的HTML页面由渲染引擎以块的形式解析，包括外部CSS文件和样式元素。然后将HTML元素转换为DOM节点已形成“内容树”或“DOM树”。
2. 同时，浏览器也会创建一个渲染树。该树包括样式信息以及定义元素显示顺序的视觉说明。渲染树确保内容以所需的顺序显示。
3. 此外，渲染树会经历布局过程。创建渲染树时，不会分配配置或大小值。计算用于评估所需位置的值的整个过程称为布局过程。在这个过程中，每个节点都被分配了精确的坐标。这确保了每个节点都出现在屏幕上的精准位置。
4. 最后一步是绘制屏幕，其中遍历渲染树，并调用渲染器的paint()方法，该方法使用UI后端层绘制屏幕上的每个节点。

如前所述，每个浏览器都有自己独特的渲染引擎。所以很自然，每个浏览器都有自己的方式来解释用户屏幕上的网页。这就是Web开发人员在跨浏览器的兼容性方面遇到的挑战。

这就是跨浏览器测试的意义所在。

[跨浏览器测试](#) 是一种质量保证方法，用于验证Web应用程序在跨多个浏览器的功能和设计方面的一致性。这些测试使QA团队能够通过进行[响应式测试](#)来探索任何问题，这可能在通过不同浏览器或浏览器版本访问他们的网站时发生。

通过了解渲染引擎的工作原理，Web开发人员可以更深入地了解网站的运行方式。因此，他们可以更有效地开发、设计和部署内容。如果一个人理解不同浏览器如何在用户屏幕上显示Web内容的细微差别，那么一个人就更有能力创建与多个浏览器兼容的内容。