

HTML DOM 加载

网页构成

HTML -- 超文本标记语言 Hyper Text Markup Language

HTML 文档 = 网页

HTML 文档包含 HTML 标签和纯文本

HTML 教程--- <https://www.w3school.com.cn/html/index.asp>

HTML 参考手册 <https://www.w3school.com.cn/tags/index.asp>

JavaScript DOM 对象参考手册-- <https://www.w3school.com.cn/jsref/index.asp>

HTML 重要的两个标签:

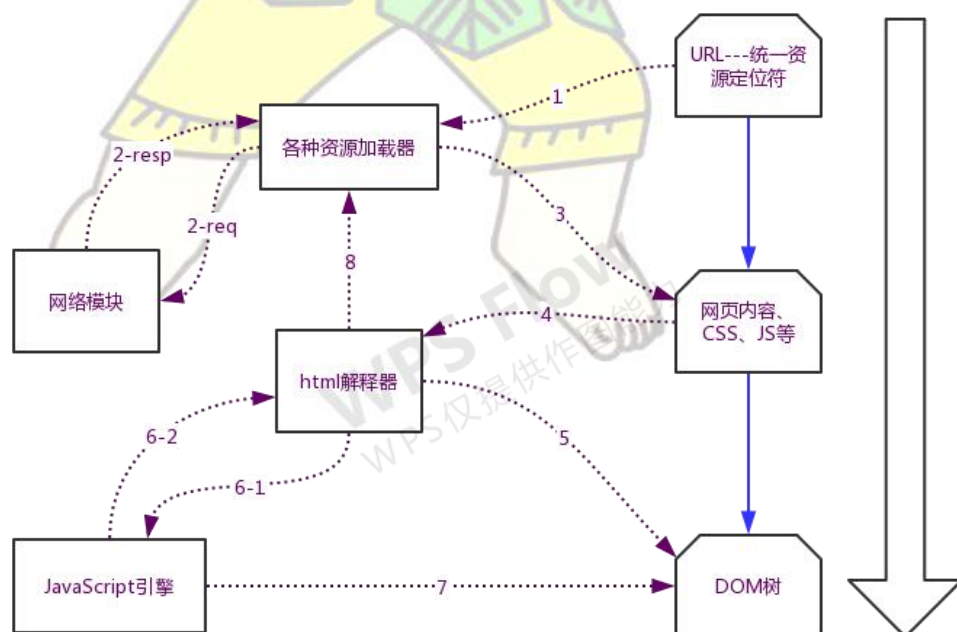
`<link>` 定义文档与外部资源的关系(如 CSS), 这个标签是空(内容)元素, 只包含属性。只能存在于 head 部分

`<script>` 定义客户端脚本(如 JavaScript) 必需的 type 属性规定脚本的 MIME 类型。

网页加载与渲染（详细内容后续补充）

web 浏览器的主要作用是将用户输入的 URL 转变成可视化图像。

这里描述 DOM 树的构建过程 下面给出 visio 流程图（请参考流程图符号含义及标准）。



过程说明

html 解释器是将 [HTML 文档](#) (网页) 解释成 HTML DOM 对象, CSS 为 DOM 的各个元素对象计算出样式信息, 从而为计算最后网页的布局提供支持, JavaScript 引擎解释 JS 代码并通过 [JavaScript DOM 接口](#) 和 [CSSOM](#) (CSS Object Model) 接口来修改网页内容和样式信息

1. 当用户输入网页 URL 时, WebKit 调用资源加载器加载该 URL 对应的网页。
2. 加载器依赖网络建立连接, 发送请求并接受答复。
3. WebKit 接受到各种网页或者资源的链接, 其中某些资源通过同步或者异步获取。
4. 网页被交给 HTML 解释器解释为一系列词语。
5. 解释器根据词语构建节点 (NODE), 形成 DOM 树。
6. 如果节点是 JavaScript 代码就调用 JavaScript 引擎解释并执行。
7. JavaScript 代码可能会修改 DOM 树的结构。
8. 如果节点依赖其他资源, 则调用资源加载器加载, 但他们是异步的, 不会阻碍当前 DOM 树的继续构建; 如果是 JavaScript 资源 URL (没有标志异步方式), 则需要停止当前 DOM 树构建, 直到 JavaScript 资源加载完毕并被 JavaScript 引擎执行后才继续 DOM 树构建。

上述过程中, 网页会发出“DOMContentLoaded”和 DOM“onload”事件, 分别在 DOM 树构建完之后, 以及 DOM 树构建完并且网页所依赖的资源都加载完之后发生, 因为某些资源的加载并不会阻碍 DOM 树的创建, 所以这两个事件多数时候不是同时发生的。

