

# 浏览器渲染机制

## 1.什么是DOCTYPE以及其作用？

DOCTYPE是document type（文档定义）的简写，用来说明web设计中所用的html或xhtml的类型，指出浏览器或者其他阅读程序按照什么样的规则（W3C所发布的一个文档类型定义即DTD）集去解释文档中的标记。

## 浏览器渲染过程

**为什么要了解浏览器的渲染机制，主要还是页面优化：**

1. 了解浏览器如何进行加载，我们可以在引用外部样式文件，外部JS时，将它们放到合适的位置，是浏览器以最快的速度，将文件加载完毕。
2. 了解浏览器如何进行解析，我们可以在构建DOM结构，组织CSS选择器的时候，选择最优的写法，提高浏览器的解析速率。
3. 了解浏览器如何进行渲染，明白渲染的过程，我们在设置元素属性，编写JS文件时，可以减少“重绘”，“重新布局”的消耗。

### 渲染机制的几个概念

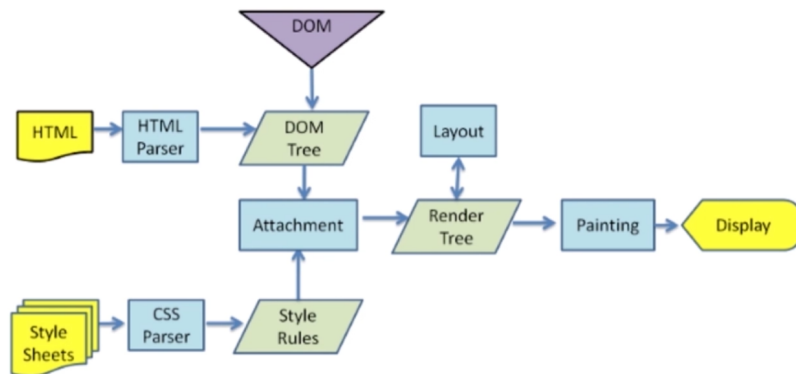
1. DOM: Document Object Model，浏览器将HTML解析成树形的数据结构，简称DOM。
2. CSSOM: CSS Object Model，浏览器将CSS解析成树形的数据结构，简称CSSOM。
3. Render Tree: DOM和CSSOM合并后生成Render Tree
4. Layout: 计算出Render Tree每个节点的具体位置。
5. Painting: 通过显卡，将Layout后的节点内容分别呈现到屏幕上。

### 注意

1. 当我们浏览器获得HTML文件后，会自上而下的加载，并在加载过程中进行解析和渲染。
2. 加载说的就是获取资源文件的过程，如果在加载过程中遇到外部CSS文件和图片，浏览器会另外发送一个请求，去获取CSS文件和相应的图片，这个请求是异步的，并不会影响HTML文件的加载。
3. 但是如果遇到Javascript文件，HTML文件会挂起渲染的进程，等待JavaScript文件加载完毕后，再继续渲染。（为什么HTML需要等待JavaScript呢？因为JavaScript可能会修改DOM，导致后续HTML资源白白加载，所以HTML必须等待JavaScript文件加载完毕后，再继续渲染，这也就是为什么JavaScript文件在写在底部body标签前的原因。）

## 二面/三面

### 渲染机制类 浏览器渲染过程



慕课网

#### 浏览器渲染过程解释如下：

1. 当用户输入一个URL时，浏览器就会向服务器发出一个请求，请求URL对应的资源
2. 接受到服务器的响应内容后，浏览器的HTML解析器，会将HTML文件解析成一棵DOM树，DOM树的构建是一个**深度遍历**的过程，当前节点的所有子节点都构建完成以后，才会去构建当前节点的下一个兄弟节点。
3. 将CSS解析成CSSOM树（CSS Rule Tree）
4. 根据DOM树和CSSOM树，来构建Render Tree（渲染树），注意**渲染树，并不等于DOM树，因为一些像head或display:none的东西，就没有必要放在渲染树中了。**
5. 有了Render Tree，浏览器已经能知道网页中有哪些节点，各个节点的CSS定义，以及它们的从属关系，下一步操作就是Layout,顾名思义，就是计算出每个节点在屏幕中的位置。
6. Layout后，浏览器已经知道哪些节点要显示，每个节点的CSS属性是什么，每个节点在屏幕中的位置是哪里，就进入了最后一步painting,按照算出来的规则，通过显卡，把内容画到屏幕上。

#### Reflow（回流）和Repaint（重绘）

Reflow：改变某个元素的背景色、文字颜色、边框颜色等等不影响它周围或内部布局的属性时，屏幕的一部分要重画，但是元素的几何尺寸没有变

Repaint：元件的几何尺寸变了，我们需要**重新验证并计算Render Tree**。是Render Tree的一部分或全部发生了变化。

`display:none`会触发`reflow`，而`visibility:hidden`只会触发`repaint`，因为没有发现位置变化。