

季诗筱的博客

一个"前端攻城小狮子" 的攻城历程。

个人资料



季诗筱

关注

发私信

访问：111491次

积分：2167

等级：BLOG > 5

排名：第15181名

原创：97篇

转载：6篇

译文：1篇

评论：98条

文章搜索

文章分类

jQuery (12)

前端工具 (8)

学习计划 (1)

Bootstrap (1)

学习收获 (5)

javascript (34)

css (16)

js库 (4)

PhotoShop (1)

阶段性总结 (1)

react (3)

代码积累 (1)

移动web (4)

CSS3 (3)

面试总结 (9)

MySQL (1)

Yii (2)

php (2)

Less (2)

Ajax (1)

程序员3月书讯

开发者应该如何谈薪资

Python数据分析与机器学习

博客搬家，有礼相送

浏览器力

渲染的过程

标签：性能优化 渲染

2016-07-24 17:05

4698人阅读

评论

分类：学习收获 (4)

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)

[+]

最近在学习性能优化，学习了[雅虎军规](#)，可是觉着有点云里雾里的，因为里面有些东西虽然自己也一直在使用，但是感觉不太明白，css和js文件的顺序。所以就花了时间去了解浏览器的工作，有一篇经典的文章[《how browsers work》](#)，讲的很详细，文章有点太长，也讲了一堆东西，还是自己总结一下。

为什么要了解浏览器加载、解析、渲染这个过程？

好，我们先说一下，为什么要了解这些呢？如果想写出一个最佳实践的页面，就要好好了解。

- 了解浏览器如何进行加载，可以在引用外部样式文件，外部js时，将他们放到合适的位置，使浏览器以最快的速度将文件
- 了解浏览器如何进行解析，可以在构建DOM结构，组织css选择器时，选择最优的写法，提高浏览器的解析速率。
- 了解浏览器如何进行渲染，明白渲染的过程，在设置元素属性，编写js文件时，可以减少“ reflow ”“ repaint ”的消耗。

正文开始

一、浏览器的主要功能

浏览器的主要功能是将用户选择的web资源呈现出来，它需要从服务器请求资源，并将其显示在浏览器窗口中，资源的格式PDF、image及其他格式。用户用URI（Uniform Resource Identifier统一资源标识符）来指定所请求资源的位置，通过DNS查址。整个浏览器工作的流程，之前博客中有论述：

1. 输入网址。
2. 浏览器查找域名的IP地址。
3. 浏览器给web服务器发送一个HTTP请求
4. 网站服务的永久重定向响应
5. 浏览器跟踪重定向地址 现在，浏览器知道了要访问的正确地址，所以它会发送另一个获取请求。
6. 服务器“处理”请求，服务器接收到获取请求，然后处理并返回一个响应。
7. 服务器发回一个HTML响应

- git (2)
- ES6 (3)
- 正则表达式 (2)
- 学习感悟 (1)
- JS设计模式 (1)
- HTML5 (2)
- 计算机网络 (1)
- Http原理 (2)
- 算法 (1)
- 性能优化 (1)
- Gulp (1)
- R语言 (5)

文章存档

2017年03月 (4)

2017年02月 (1)

2016年11月 (1)

2016年09月 (3)

2016年08月 (8)

展开

阅读排行

【webpack】使用webpack... (6671)

【CSS3】transition过渡和an... (6517)

Sublime Text3 支持Less (6113)

css中设置table中的td内容自... (6019)

【JavaScript】DOM (4858)

【JavaScript】事件 (4819)

浏览器加载、解析、渲染的过程 (4656)

HTTP请求头部+响应码 (3723)

【JavaScript】Cookie and ... (3613)

正则表达式 (3607)

评论排行

移动web学习--淘宝 flexible.js (15)

浏览器加载、解析、渲染的过程 (9)

【学习感悟】找工作之前的焦虑 (8)

Sublime Text3 支持Less (7)

react.js学习（1） webpack (7)

继承再学习 (5)

布局 (4)

Drag, js实现鼠标拖拽元素 (4)

正则表达式 (4)

【JavaScript】事件 (4)

推荐文章

\* CSDN日报20170402 —— 《我希望进入大学时就能知道的一些事儿》

\* 最全面总结 Android WebView与 JS 的交互方式

\* 蓝牙DA14580开发：固件格式、二次引导和烧写

\* 你不知道的 Android WebView 使用漏洞

最新评论

css中设置table中的td内容自动换行 kfengqingyangk : 转载一下，谢谢啦

Swiper.js 实现移动端元素左右滑动 伴随风飞 : 有男朋友了没有~？

移动端屏幕自适应 风雨中的人们 : 谢谢您的代码

Sublime Text3 支持Less 季诗筱 : @qq\_35969766:安装插件，你的不可以吗？安装完了重启sublime

8. 浏览器开始显示HTML

9. 浏览器发送请求，以获取嵌入在HTML中的对象。在浏览器显示HTML时，它会注意到需要获取其他地址内容的标签。这取请求来重新获得这些文件。这些文件就包括CSS/JS/图片等资源，这些资源的地址都要经历一个和HTML读取类似的过程。所些域名，发送请求，重定向等等...

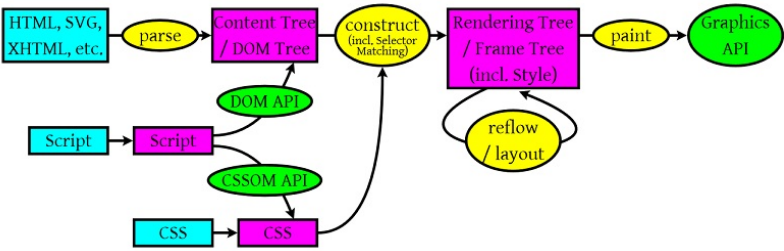
那么，一个页面，究竟是如何从我们输入一个网址到最后完整的呈现在我们面前的呢？还需要了解一下浏览器是如何渲染的：

二、浏览器的渲染

下面是渲染引擎在取得内容之后的基本流程：

解析html以构建dom树 -> 构建render树 -> 布局render树 -> 绘制render树

先来看个图：



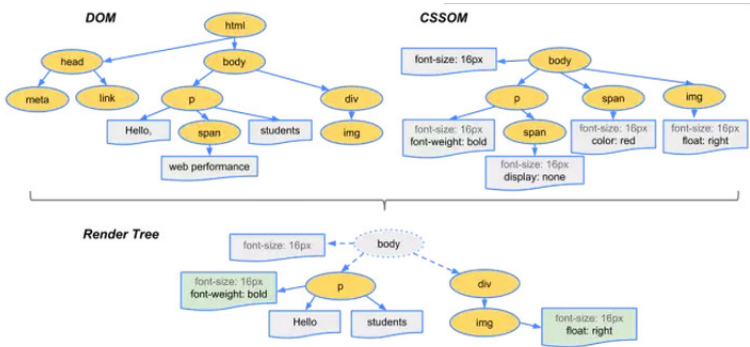
所以，浏览器会解析三个东西：

- (1) HTML/SVG/XHTML，解析这三种文件会产生一个 DOM Tree。
- (2) CSS，解析 CSS 会产生 CSS 规则树。
- (3) JavaScript脚本，主要是通过 DOM API 和 CSSOM API 来操作 DOM Tree 和 CSS Rule Tree。

我今天又纠结了一上午，到底是怎么解析怎么渲染的，我的疑问在于，浏览器到底是先解析生成了DOM树，然后再加载CSS JS: DOM的过程中，遇到了 link script 然后就加载CSS JS，边加载边渲染。我有这种疑问的原因在于，看网上的帖子，说的根本不说，这个写的让我直接懵逼，真的是直接懵逼啊，学习的过程中，总会遇到困难，但这次，让我真的好难啊。不过正因为不懂才==！我又查了一上午，自己测试测试测试，然后觉着，我好像是明白点了。真的推荐大家去认真看《how browsers work》这的时候，还是要从比较权威的资料看起比较好，也不要像我今天这样，无头苍蝇乱查。

那么就来说一下图中的过程，我是按照自己的理解来说，如果有误，欢迎指正。

当浏览器获得一个html文件时，会“自上而下”加载，并在加载过程中进行解析渲染。  
解析：  
1. 浏览器会将HTML解析成一个DOM树，DOM 树的构建过程是一个深度遍历过程：当前节点的所有子节点都构建好后才：一个兄弟节点。  
2. 将CSS解析成 CSS Rule Tree 。  
3. 根据DOM树和CSSOM来构造 Rendering Tree。注意：Rendering Tree 渲染树并不等同于 DOM 树，因为一些像 He的东西就没必要放在渲染树中了。



4.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定义以及他们的从属关系。下一步操作称之为

Sublime Text3 支持Less  
qq\_35969766 : @anythang: 在么? LESS  
编译后一行显示CSS 最后你解决了吗?

闭包再学习  
季诗筱 : @u012609915:感谢你发现了我的  
错误~~

闭包再学习  
季诗筱 : @qjyue77:谢谢~~

闭包再学习  
Somei : 博主, 你的文章写的挺好的。不过  
有个小小的地方要纠正下哈, 就是那个Exam  
ple1中show函数, 你写...

浏览器加载、解析、渲染的过程  
wangfuqiang721 : 详细~看了就要实践

浏览器加载、解析、渲染的过程  
CHENzh0213 : 写得很详细, 谢谢

计算出每个节点在屏幕中的位置。  
5.再下一步就是绘制，即遍历render树，并使用UI后端层绘制每个节点。

重点来了：

上述这个过程是 逐步完成 的，为了更好的用户体验，渲染引擎将会尽可能早的将内容呈现到屏幕上，并不会等到所有的html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，可能还在通过网络下载其余内容。（这段话是《how browser works》我茅塞顿开）

几个概念：

- （1）Reflow（回流）：浏览器要花费时间去渲染，当它发现了某个部分发生了变化影响了布局，那就需要倒回去重新渲染。
- （2）Repaint（重绘）：如果只是改变了某个元素的背景颜色，文字颜色等，不影响元素周围或内部布局的属性，将只会引起部分。

Reflow要比Repaint更花费时间，也就更影响性能。所以在写代码的时候，要尽量避免过多的Reflow。

reflow的原因：

- （1）页面初始化的时候；
- （2）操作DOM时；
- （3）某些元素的尺寸变了；
- （4）如果 CSS 的属性发生了变化了。

减少 reflow/repaint

- （1）不要一条一条地修改 DOM 的样式。与其这样，还不如预先定义好 css 的 class，然后修改 DOM 的 className。
- （2）不要把 DOM 结点的属性值放在一个循环里当成循环里的变量。
- （3）为动画的 HTML 元件使用 fixed 或 absolute 的 position，那么修改他们的 CSS 是不会 reflow 的。
- （4）千万不要使用 table 布局。因为可能很小的一个小改动会造成整个 table 的重新布局。

我应该是已经把网上所有的关于浏览器加载 解析 渲染过程的文章都看全了，其中写的比较好的一个版本是[下面](#)这个：

HTML页面加载和解析流程

1. 用户输入网址（假设是个html页面，并且是第一次访问），浏览器向服务器发出请求，服务器返回html文件；
2. 浏览器开始载入html代码，发现< head > 标签内有一个< link > 标签引用外部CSS文件；
3. 浏览器又发出CSS文件的请求，服务器返回这个CSS文件；
4. 浏览器继续载入html中< body > 部分的代码，并且CSS文件已经拿到手了，可以开始渲染页面了；
5. 浏览器在代码中发现一个< img > 标签引用了一张图片，向服务器发出请求。此时浏览器不会等到图片下载完，而是继续
6. 服务器返回图片文件，由于图片占用了一定面积，影响了后面段落的排布，因此浏览器需要回过头来重新渲染这部分代码
7. 浏览器发现了一个包含一行Javascript代码的< script > 标签，赶快运行它；
8. Javascript脚本执行了这条语句，它命令浏览器隐藏掉代码中的某个< div > （style.display=" none" ）。突然少了这，得不重新渲染这部分代码；
9. 终于等到了</html>的到来，浏览器泪流满面.....
10. 等等，还没完，用户点了一下界面中的“换肤”按钮，Javascript让浏览器换了一下< link > 标签的CSS路径；
11. 浏览器召集了在座的各位< div > < span > < ul > < li > 们，“大伙儿收拾收拾行李，咱得重新来过.....”，浏览器向服务器，重新渲染页面。

与讨论主题相关的其他思考

编写CSS时应该注意：

CSS选择符是从右到左进行匹配的。从右到左！所以，#nav li 我们以为这是一条很简单的规则，秒秒钟就能匹配到想要的元素，右往左匹配啊，所以，会去找所有的li，然后再去确定它的父元素是不是#nav.，因此，写css的时候需要注意：

1. dom深度尽量浅。
2. 减少inline javascript、css的数量。
3. 使用现代合法的css属性。
4. 不要为id选择器指定类名或是标签，因为id可以唯一确定一个元素。
5. 避免后代选择符，尽量使用子选择符。原因：子元素匹配的概率要大于后代元素匹配符。后代选择符:#tp p{} 子选择符tp p{} 子选择符
6. 避免使用通配符，举一个例子，.mod .hd \*{font-size:14px;} 根据匹配顺序,将首先匹配通配符,也就是说先匹配出通配符,再匹配

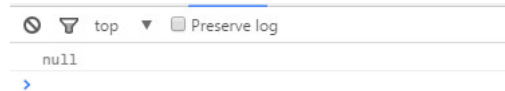
dom树上的所有节点进行遍历他的父级元素),然后匹配.mod,这样的性能耗费可想而知.

## 关于script标签的位置

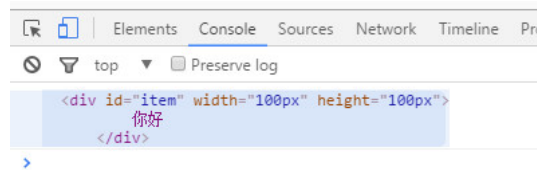
现在,我们大都会将script标签放在body结束标签之前,那原因是什么呢?我今天也做了一个测试。

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>测试js代码位置</title>
6   <script type="text/javascript">
7     var item = document.getElementById("item");
8     console.log(item);
9   </script>
10
11 </head>
12 <body>
13   <div id="item" width="100px" height="100px">
14     你好
15   </div>
16
17 </body>
18 </html>
```

上述代码中有一段js代码,要在控制台打印一个元素,我把script标签放在head里,控制台里打印出来的是null。



我又把js代码放在body结束标签之前,打印出来的就是div元素了

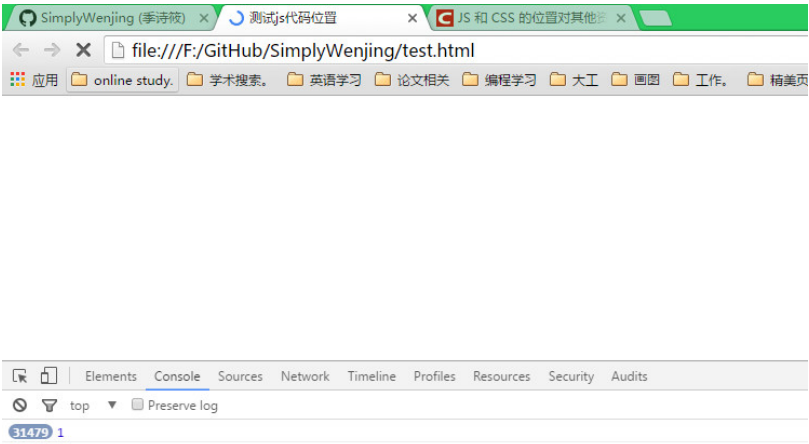


所以,通过这个简单的例子我们可以看到,js代码在加载完后,是立即执行的。

我又做了一个测试,在js代码里面写了一个死循环,把它放在head标签中,

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>测试js代码位置</title>
6   <script type="text/javascript">
7     var item = document.getElementById("item");
8     while(true) {
9       console.log(1);
10    }
11  </script>
12 </head>
13 <body>
14   <div id="item" width="100px" height="100px">
15     你好
16   </div>
17 </body>
18 </html>
```

页面是这样的:



一直在执行那个打印1的死循环，后面的body都没有加载渲染出来。所以，这个小例子，我们可以看出，js的下载和执行会阻塞

所以，Javascript的加载和执行的特点：

- （1）载入后马上执行；
- （2）执行时会阻塞页面后续的内容（包括页面的渲染、其它资源的下载）。原因：因为浏览器需要一个稳定的DOM树结构，而改变了DOM树结构，比如使用 document.write 或 appendChild,甚至是直接使用的location.href进行跳转，浏览器为了防止出重新构建DOM树的情况，所以 就会阻塞其他的下载和呈现。

减少 JavaScript 对性能的影响的方法：

1. 将所有的script标签放到页面底部，也就是body闭合标签之前，这能确保在脚本执行前页面已经完成了DOM树渲染。
2. 尽可能地合并脚本。页面中的script标签越少，加载也就越快，响应也越迅速。无论是外链脚本还是内嵌脚本都是如此。
3. 采用无阻塞下载 JavaScript 脚本的方法：
  - （1）使用script标签的 defer 属性（仅适用于 IE 和 Firefox 3.5 以上版本）；
  - （2）使用动态创建的script元素来下载并执行代码；

好了，写到这里吧，我想静静。

顶 13 踩 0

- 上一篇 【CSS3】transition过渡和animation动画
- 下一篇 【JavaScript】精准定位参数学习

我的同类文章

学习收获（4）			
• 【移动端】基于rem实现手机端页面自...	2016-04-26	阅读 1599	• 【今日总结】 2016-02-
• 【second day】学习精准切图+高度自...	2016-02-22	阅读 496	• 【first day】学会用不同方法实现页面... 2016-02-

参考知识库



JavaScript知识库  
14076 关注 | 1514 收录



软件测试知识库  
4285 关注 | 318 收录



jQuery知识库  
8502 关注 | 1014 收录



AngularJS知识库  
4597 关注 | 542 收录

猜你在找

WEB前端整套教程html+divcss+ja...  
HTML5之新增标签，CSS3，js特效  
Jquery入门到精通（备html、css...  
HTML&CSS;实战教程  
全网首发：HTML5+CSS3速成实战...

HTML基础和JSP了解及JSP中代码...  
meta link之预加载图片文件  
JavaScript基础浏览器对象模型BOM  
QQ浏览器X5内核问题汇总  
CSS 加载新方式

查看评论



wangfuqiang721  
详细~看了就要实践

6楼 2017-04-05 10:00



CHENzh0213  
写得很详细，谢谢

5楼 2017-04-05 09:59



vince12580  
很棒

4楼 2017-04-05 09:58



Rubylo  
厉害了我的姐。

3楼 2017-04-05 09:57



季诗筱  
回复u012273867：O(∩\_∩)O~

Re: 2017-04-05 09:56



lyj12300  
写的不错~内容很详细。教程很好。赞一个！

2楼 2017-04-05 09:55



季诗筱  
回复lyj12300：谢谢~~

Re: 2017-04-05 09:54



季诗筱  
学习的最好的方法是实践，查这么多资料，不如自己亲自实践一下，打开谷歌浏览器，在Timeline里，所有的加载和渲染看的一清二楚。悔恨自己花了一天的宝贵时间查资料！！首先会发送请求给服务器，得到响应后，加载完毕html，之后开始解析html，解析过程中遇到了css文件或者js文件，接着发请求，得到响应后，加载完毕css文件，然后解析css文件，继续解析html，然后计算样式，形成render tree，然后layout，周后paint

1楼 2017-04-05 09:53



SuperThreeX  
回复XIAOZHUXMEN：最新版chrome的timeline咋看顺序???网上很多教程都是旧版的timeline。html文件要加载完毕才会进行解析吗???但是有时候我看network的时间线，html的load期间内，就有外部资源的请求发起了。就是说html都没加载完，css就加载了

Re: 2017-04-05 09:52

完，它就知道里面的外部资源链接并发起请求获取了...咋整啊???网上说的乱七八糟的，没有统一的说法，还有矛盾的

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	Docker	OpenStack	VP	
IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP	jQuery	BI	HTML5	Spring	Apache
HTML	SDK	IIS	Fedora	XML	LBS	Unity	Splashtop	UML	components	Windows Mobile	Rails	
Cassandra	CloudStack	FTC	coremail	OPhone	CouchBase	云计算	iOS6	Rackspace	Web App	S		
Compuware	大数据	aptech	Perl	Tornado	Ruby	Hibernate	ThinkPHP	HBase	Pure	Solr	Android	
Cloud Foundry	Redis	Scala	Django	Bootstrap								

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技术有限公司  
京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 