

目录	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	最近阅读
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	
20 重绘与回流的相爱相杀	
21 Event Loop 力挽狂澜	
第6章 应用实践	
22 防抖节流背后那些事儿	
23 让加载“懒”一点	
24 服务端渲染	
25 移动端的优化技巧也想让你知道	
26 不可不知的React、Vue 优化技巧	
27 写在最后	

已到底部

16 渲染原理与性能优化

更新时间：2019-08-27 09:38:10



“完成工作的方法，是爱惜每一分钟。”——达尔文

不知道大家在面试的过程中是否遇到过这样一个问题，浏览器的渲染引擎有几种？答案如下：

- IE（Trident）
- Chrome（Blink）
- Firefox（Gecko）
- Opera（Blink）
- Safari（Webkit）
- UC（U3）
- QQ浏览器微信（X5/Blink）

那么这个问题考察的就是渲染原理方面的知识点，上面就是市面上主要浏览器对应的渲染引擎，通过这个问题我们就引出了我们今天要介绍的内容。可以看到目前 Blink 渲染引擎占据了半壁江山，当然 Blink 的前身都是由 Webkit 优化改造而来的。也正是因为有这些不同的渲染引擎，大家才会遇到浏览器的兼容性问题，相信随着规范不断推进，需要我们解决的兼容性问题将会越来越少。

渲染原理解析

关于渲染原理，我们首先来看这张图，如下：

<div><div>← 慕课专栏</div><div>☰ 你不知道的前端性能优化技巧 / 16 渲染原理与性能优化</div></div>	
目录	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	最近阅读
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	
20 重绘与回流的相爱相杀	
21 Event Loop 力挽狂澜	
第6章 应用实践	
22 防抖节流背后那些事儿	
23 让加载“懒”一点	
24 服务端渲染	
25 移动端的优化技巧也想让你知道	
26 不可不知的React、Vue 优化技巧	
27 写在最后	
已到底部	

上图就是整个渲染流程的总体流程图，下面我们来——介绍渲染流程当中的每个步骤。

首先是解析HTML，这个过程主要是把 HTML 文档解析为 DOM 树的过程。如果遇到 `<script>` 标签会停止解析，先执行标签当中 JavaScript；如果是外联方式，也需要等待下载并且执行完对应的 JavaScript 代码，然后才能够继续执行解析 HTML 的工作。HTML 解析完成后触发 DOMContentLoaded 事件，这里我们就可以操作 DOM了。最后我们来看这样一段代码，如下：

这段代码经过 HTML 解析器处理之后，生成对应的 DOM 树结构如下：

Tips：上图中的 DOM 树节点类型如何获取呢，这里我们可以在 Chrome 当中选中对应元素，然后在控制台输入 `$0.constructor`，即可打印出对应的类型，如下：

<div><div>← 慕课专栏</div><div>三 你不知道的前端性能优化技巧 / 16 渲染原理与性能优化</div></div>	
目录	HTML 的处理方法是一样的，这里不在赘述。CSS 解析器将 CSS 解析成 CSSStyleSheet (也被叫做 CSSOM 树)，这里的 CSSOM 树与 DOM 树结构类似。解析对应关系如下：
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化 <div>最近阅读</div>	
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	
20 重绘与回流的相爱相杀	
21 Event Loop 力挽狂澜	
第6章 应用实践	
22 防抖节流背后那些事儿	
23 让加载“懒”一点	
24 服务端渲染	
25 移动端的优化技巧也想让你知道	
26 不可不知的React、Vue 优化技巧	
27 写在最后	
已到底部	

接着由 DOM 树和 CSSOM 树一起附着合成渲染树(Render Tree)，显而易见地我们可以得到这个公式：

DOM Tree + CSSOM Tree= Render Tree。同样这里 Render 树我们也可以简单用图来表示一下，如下：



图中左侧为 DOM 树，右侧是 Render 树，可以看到 Render 树结构与 DOM 树是非常相似的，但不是——对应的。每一个节点是一个 Render Object，包含了我们在样式表中设置的样式，样式的宽高和具体位置是通过Layout (重排)计算得出。

Tips：

- 元素如果被设置为 display:none，在 DOM 树中依然会显示，但是在 Render 树中不会显示；
- 元素如果被设置为 visibility:none，那么 DOM 树和 Render 树中都会显示；
- 我们经常说的脱离文档流，其实就是脱离 Render Tree。

以上就是浏览器整个渲染流程的介绍。了解了流程之后，我们就可以有针对性地对浏览器的渲染过程提出相应的优化建议。

针对渲染原理进行相关优化

在渲染原理当中我们提到，CSS 解析和HTML解析是同步进行的，那么一个 HTML 文档首先解析的肯定是 HTML，然后才是 CSS，这就导致了 HTML 解析完成后，往往需要等待 CSS 解

← 慕课专栏	☰ 你不知道的前端性能优化技巧 / 16 渲染原理与性能优化
目录	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	最近阅读
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	
20 重绘与回流的爱恨情仇	
21 Event Loop 力挽狂澜	
第6章 应用实践	
22 防抖节流背后那些事儿	
23 让加载“懒”一点	
24 服务端渲染	
25 移动端的优化技巧也想让你知道	
26 不可不知的React、Vue 优化技巧	
27 写在最后	

已到底部

精选留言 2

<div><div><div>←</div><div>慕课专栏</div></div><div><div>≡</div><div>你不知道的前端性能优化技巧 / 16 渲染原理与性能优化</div></div></div>	
目录	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	最近阅读
17 如何应对首屏“一片空白”（上）	
18 如何应对首屏“一片空白”（下）	
19 不容小觑的 DOM 性能优化	
20 重绘与回流的相爱相杀	
21 Event Loop 力挽狂澜	
第6章 应用实践	
22 防抖节流背后那些事儿	
23 让加载“懒”一点	
24 服务端渲染	
25 移动端的优化技巧也想让你知道	
26 不可不知的React、Vue 优化技巧	
27 写在最后	
已到底部	

suarezxyw96

老师，我看到网上有文章说defer是在DOMContentLoaded事件之前执行的

👍 0 回复

2019-10-11

黑芝麻_ 回复 suarezxyw96

dom都没有，怎么执行？

回复

2020-06-15 08:51:27

菜鸟x

DomContentLoaded事件是在dom树渲染完后执行，那么这个事件是不是跟window.onload同样作用

👍 0 回复

2019-09-04

竹蜻蜓请带我飞 回复 菜鸟x

DOMContentLoaded,事件触发是在DOM结构加载完成，onload事件触发是在所有资源(图片、视频等)全部加载完成的时候才会触发

回复

2019-09-29 08:49:59

菜鸟x 回复 竹蜻蜓请带我飞

好的。谢谢

回复

2019-09-29 08:51:00

千学不如一看，千看不如一练