

详解 HTML5 中 rel 属性的 prefetch 预加载功能使用

在 HTML5 中,有个很有用但常被忽略的特性,就是预先加载(prefetch),它的原理是:

利用浏览器的空闲时间去先下载用户指定需要的内容,然后缓存起来,这样用户下次加载时,就直接从缓存中取出来,效率就快了。

举个例子说明:比如要预先加载某个页面,可以这样:

XML/HTML Code

1. `<link rel="prefetch" href="http://www.example.com/"> <!-- Firefox -->`

但如果是 google 的话,要用另外的一个名称,即:

XML/HTML Code

1. `<link rel="prerender" href="http://www.example.com/"> <!-- Chrome -->`

即使在不支持的浏览器,用了这个特性其实是不会出错的,只不过浏览器解析不到而已,

所以,如果你感觉能有办法预先预测到用户期望点的页面(比如用户看最新的受欢迎的热图,他可能看了第一页后,会继续看下一页,这个时候就可以用预先加载这个特性了).比如

XML/HTML Code

1. `<link rel="prefetch" href="<?php echo get_next_posts_page_link(); ?>">`

而单独取一张图片也是可以的,比如:

XML/HTML Code

1. `<link rel="prefetch" href="/images/test.jpg"/>`

有了浏览器缓存,为何还需要预加载?

1. 用户可能是第一次访问网站,此时还无缓存
2. 用户可能清空了缓存
3. 缓存可能已经过期,资源将重新加载
4. 用户访问的缓存文件可能不是最新的,需要重新加载

5.Chrome 的预加载技术

现在的 chrome 聪明到根据你的浏览记录,预测到你可能访问或搜索哪些网站,在你打开网站之前就加载好了一些资源了。

举个栗子,当你在搜索框输入 "amaz" 时,它猜测到你可能要访问 amazon.com,可能就帮你加载了这个网站的一些资源。

如果这个预测算法精准的话,就能大大地提高用户的浏览体验了。

DNS prefetch

我们知道,当我们访问一个网站如 www.amazon.com 时,需要将这个域名先转化为对应的 IP 地址,这是一个非常耗时的过程。

DNS prefetch 分析这个页面需要的资源所在的域名,浏览器空闲时提前将这些域名转化为 IP 地址,真正请求资源时就避免了上述这个过程的时间。

XML/HTML Code

1. `<meta http-equiv='x-dns-prefetch-control' content='on'>`
2. `<link rel='dns-prefetch' href='http://g-ecx.images-amazon.com'>`
3. `<link rel='dns-prefetch' href='http://z-ecx.images-amazon.com'>`

4. `<link rel='dns-prefetch' href='http://ecx.images-amazon.com'>`
5. `<link rel='dns-prefetch' href='http://completion.amazon.com'>`
6. `<link rel='dns-prefetch' href='http://fls-na.amazon.com'>`

应用场景 1: 我们的资源存在在不同的 CDN 中, 那提前声明好这些资源的域名, 就可以节省请求发生时产生的域名解析的时间。

应用场景 2: 如果我们知道用户接下来的操作一定会发起一起资源的请求, 那就可以将这个资源进行 DNS-Prefetch, 加强用户体验。

Resource prefetch

在 Chrome 下, 我们可以用 link 标签声明特定文件的预加载:

XML/HTML Code

1. `<link rel='subresource' href='critical.js'>`
2. `<link rel='subresource' href='main.css'>`
3. `<link rel='prefetch' href='secondary.js'>`

在 Firefox 中或用 meta 标签声明:

XML/HTML Code

1. `<meta http-equiv="Link" content="<critical.js>; rel=prefetch">`

rel='subresource' 表示当前页面必须加载的资源, 应该放到页面最顶端先加载, 有最高的优先级。

rel='prefetch' 表示当 subresource 所有资源都加载完后, 开始预加载这里指定的资源, 有最低的优先级。

注意: 只有可缓存的资源才进行预加载, 否则浪费资源!

Pre render

前面说到的预解析 DNS、预加载资源已经够强悍了有木有, 可还有更厉害的预渲染 (Pre-rendering) !

预渲染意味着我们提前加载好用户即将访问的下一个页面, 否则进行预渲染这个页面将浪费资源, 慎用!

XML/HTML Code

1. `<link rel='prerender' href='http://www.pagetoprerender.com'>`

rel='prerender' 表示浏览器会帮我们渲染但隐藏指定的页面, 一旦我们访问这个页面, 则秒开了!

在 Firefox 中或用 rel='next' 来声明

XML/HTML Code

1. `<link rel="next" href="http://www.pagetoprerender.com">`

不是所有的资源都可以预加载

当资源为以下列表中的资源时, 将阻止预渲染操作:

1. URL 中包含下载资源
2. 页面中包含音频、视频
3. POST、PUT 和 DELETE 操作的 ajax 请求
4. HTTP 认证(Authentication)
5. HTTPS 页面
6. 含恶意软件的页面
7. 弹窗页面
8. 占用资源很多的页面
9. 打开了 chrome developer tools 开发工具

手动触发预渲染操作

在 head 中强势插入 link[rel='prerender'] 即可:

JavaScript Code

1. `var hint =document.createElement("link")`
2. `hint.setAttribute("rel","prerender")`
3. `hint.setAttribute("href","next-page.html")`
4. `document.getElementsByTagName("head")[0].appendChild(hint)`