

本课时主要讲解那些“看不见”的 HTML 标签。

提到 HTML 标签，前端工程师会非常熟悉，因为在开发页面时经常使用。但往往关注更多的是页面渲染效果及交互逻辑，也就是对用户可见可操作的部分，比如表单、菜单栏、列表、图文。

其实还有一些非常重要却容易被忽视的标签，这些标签大多数用在页面头部 head 标签内，虽然对用户不可见，但如果在某些场景下，比如交互实现、性能优化、搜索优化，合理利用它们就可以达到事半功倍的效果。

这一课时就和你来聊聊那些“看不见”的 HTML 标签及其使用场景。

## 交互实现

我经常会向我的团队成员提倡一个编码原则：**Less code, less bug**。

在实现一个功能的时候，我们编写的代码越多，不仅开发成本越高，而且代码的健壮性也越差。它和 KISS（Keep it simple, stupid）原则及奥卡姆剃刀原则（如无必要，勿增实体）有相同的意思，都是提倡**编码简约**。

下面介绍几个标签，来看看如何帮助我们更简单地实现一些页面交互效果。

### meta 标签：自动刷新/跳转

假设要实现一个类似 PPT 自动播放的效果，你很可能想到使用 JavaScript 定时器控制页面跳转来实现。但其实有更加简洁的实现方法，比如通过 meta 标签来实现：

```
<meta http-equiv="Refresh" content="5; URL=page2.html">
```

上面的代码会在 5s 之后自动跳转到同域下的 page2.html 页面。我们要实现 PPT 自动播放的功能，只需要在每个页面的 meta 标签内设置好下一个页面的地址即可。

另一种场景，比如每隔一分钟就需要刷新页面的大屏幕监控，也可以通过 meta 标签来实现，只需去掉后面的 URL 即可：

```
<meta http-equiv="Refresh" content="60">
```

细心的你可能会好奇，既然这样做又方便又快捷，为什么这种用法比较少见呢？

一方面是因为不少前端工程师对 meta 标签用法缺乏深入了解，另一方面也是因为在使用它的时候，刷新和跳转操作是不可取消的，所以对刷新时间间隔或者需要手动取消的，还是推荐使用 JavaScript 定时器来实现。但是，如果你只是想实现页面的定时刷新或跳转（比如某些页面缺乏访问权限，在 x 秒后跳回首页这样的场景）建议你可以实践下 meta 标签的用法。

### title 标签与 Hack 手段：消息提醒

作为前端工程师的你对 B/S 架构肯定不陌生，它有很多的优点，比如版本更新方便、跨平台、跨终端，但在处理某些场景，比如即时通信场景时，就会变得比较麻烦。

因为前后端通信深度依赖 HTTP 协议，而 HTTP 协议采用“请求-响应”模式，这就决定了服务端也只能被动地发送数据。一种低效的解决方案是客户端通过轮询机制获取最新消息（HTML5 下可使用 WebSocket 协议）。

消息提醒功能实现则比较困难，HTML5 标准发布之前，浏览器没有开放图标闪烁、弹出系统消息之类的接口，只能借助一些 Hack 的手段，比如修改 title 标签来达到类似的效果（HTML5 下可使用 Web Notifications API 弹出系统消息）。

下面这段代码中，通过定时修改 title 标签内容，模拟了类似消息提醒的闪烁效果：

```
let msgNum = 1 // 消息条数
let cnt = 0 // 计数器
const interval = setInterval(() => {
  cnt = (cnt + 1) % 2
  if(msgNum===0) {
    // 通过DOM修改title
    document.title += `聊天页面`
    clearInterval(interval)
    return
  }
  const prefix = cnt % 2 ? `新消息(${msgNum})` : ''
  document.title = `${prefix}聊天页面`
}, 1000)
```

实现效果如下图所示，可以看到标签名称上有提示文字在闪烁。



通过模拟消息闪烁，可以让用户在浏览其他页面的时候，及时得知服务端返回的消息。

定时修改 title 标签内容，除了用来实现闪烁效果之外，还可以制作其他动画效果，比如文字滚动，但需要注意浏览器会对 title 标签文本进行去空格操作。

动态修改 title 标签的用途不仅在于消息提醒，你还可以将一些关键信息显示到标签上（比如下载时的进度、当前操作步骤），从而提升用户体验。

## 性能优化

性能优化是前端开发中避不开的问题，性能问题无外乎两方面原因：渲染速度慢、请求时间长。性能优化虽然涉及很多复杂的原因和解决方案，但其实只要通过合理地使用标签，就可以在一定程度上提升渲染速度以及减少请求时间。

### script 标签：调整加载顺序提升渲染速度

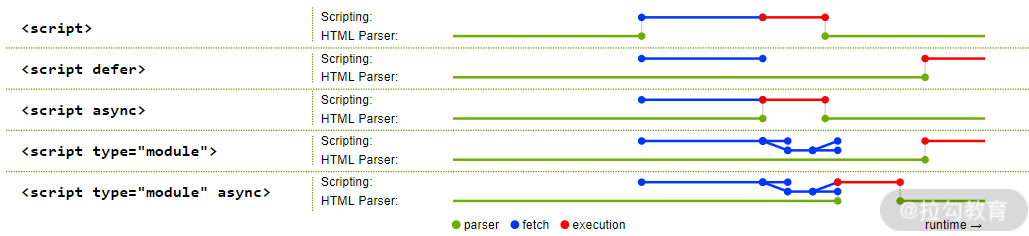
由于浏览器的底层运行机制，渲染引擎在解析 HTML 时，若遇到 script 标签引用文件，则会暂停解析过程，同时通知网络线程加载文件，文件加载后会切换至 JavaScript 引擎来执行对应代码，代码执行完成之后切换至渲染引擎继续渲染页面。

在这一过程中可以看到，页面渲染过程中包含了请求文件以及执行文件的时间，但页面的首次渲染可能并不依赖这些文件，这些请求和执行文件的动作反而延长了用户看到页面的时间，从而降低了用户体验。

为了减少这些时间损耗，可以借助 script 标签的 3 个属性来实现。

- async 属性**。立即请求文件，但不阻塞渲染引擎，而是文件加载完毕后阻塞渲染引擎并立即执行文件内容。
- defer 属性**。立即请求文件，但不阻塞渲染引擎，等到解析完 HTML 之后再执行文件内容。
- HTML5 标准 type 属性**，对应值为“module”。让浏览器按照 ECMA Script 6 标准将文件当作模块进行解析，默认阻塞效果同 defer，也可以配合 async 在请求完成后立即执行。

具体效果可以参看下图：



其中，绿色的线表示执行解析 HTML，蓝色的线表示请求文件，红色的线表示执行文件。

从图中可以得知，采用 3 种属性都能减少请求文件引起的阻塞时间，只有 `defer` 属性以及 `type="module"` 情况下能保证渲染引擎的优先执行，从而减少执行文件内容消耗的时间，让用户更快地看见页面（即使这些页面内容可能并没有完全地显示）。

除此之外还应当注意，当渲染引擎解析 HTML 遇到 `script` 标签引入文件时，会立即进行一次渲染。所以这也就是为什么构建工具会把编译好的引用 JavaScript 代码的 `script` 标签放入到 `body` 标签底部，因为当渲染引擎执行到 `body` 底部时会先将已解析的内容渲染出来，然后再去请求相应的 JavaScript 文件。如果是内联脚本（即不通过 `src` 属性引用外部脚本文件直接在 HTML 编写 JavaScript 代码的形式），渲染引擎则不会渲染。

### link 标签：通过预处理提升渲染速度

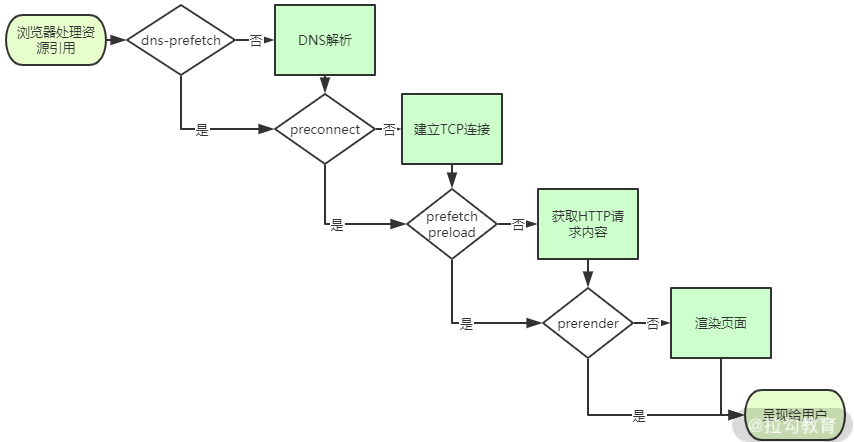
在我们对大型单页应用进行性能优化时，也许会用到按需懒加载的方式，来加载对应的模块，但如果能合理利用 `link` 标签的 `rel` 属性值来进行预加载，就能进一步提升渲染速度。

- **dns-prefetch**。当 `link` 标签的 `rel` 属性值为 `"dns-prefetch"` 时，浏览器会对某个域名预先进行 DNS 解析并缓存。这样，当浏览器在请求同域名资源的时候，能省去从域名查询 IP 的过程，从而减少时间损耗。下图是淘宝网设置的 DNS 预解析。

```
<link rel="dns-prefetch" href="//g.alicdn.com">
<link rel="dns-prefetch" href="//img.alicdn.com">
<link rel="dns-prefetch" href="//tce.alicdn.com">
<link rel="dns-prefetch" href="//gw.mmstat.com">
<link rel="dns-prefetch" href="//tce.taobao.com">
<link rel="dns-prefetch" href="//log.mmstat.com">
<link rel="dns-prefetch" href="//tui.taobao.com">
<link rel="dns-prefetch" href="//ald.taobao.com">
<link rel="dns-prefetch" href="//gw.alicdn.com">
<link rel="dns-prefetch" href="//atamx.alicdn.com">
<link rel="dns-prefetch" href="//dfms.tanx.com">
<link rel="dns-prefetch" href="//ecom.tanx.com">
<link rel="dns-prefetch" href="//res.mmstat.com">
```

- **preconnect**。让浏览器在一个 HTTP 请求正式发给服务器前预先执行一些操作，这包括 DNS 解析、TLS 协商、TCP 握手，通过消除往返延迟来为用户节省时间。
- **prefetch/preload**。两个值都是让浏览器预先下载并缓存某个资源，但不同的是，`prefetch` 可能会在浏览器忙时被忽略，而 `preload` 则是一定会被预先下载。
- **prerender**。浏览器不仅会加载资源，还会解析执行页面，进行预渲染。

这几个属性值恰好反映了浏览器获取资源文件的过程，在这里我绘制了一个流程简图，方便你记忆。



浏览器获取资源文件的流程

### 搜索优化


你所写的前端代码，除了要让浏览器更好执行，有时候也要考虑更方便其他程序（如搜索引擎）理解。合理地使用 `meta` 标签和 `link` 标签，恰好能让搜索引擎更好地理解和收录我们的页面。

#### meta 标签：提取关键信息

通过 `meta` 标签可以设置页面的描述信息，从而让搜索引擎更好地展示搜索结果。

例如，在百度中搜索“拉勾”，就会发现网站的描述信息，这些描述信息就是通过 `meta` 标签专门为搜索引擎设置的，目的是方便用户预览搜索到的结果。

拉勾网-专业的互联网招聘平台 官方




拉勾网是专业的互联网行业招聘平台。致力于提供真实可靠的互联网招聘信息。工资不面议。找工作、要招聘、找人才就来拉勾网。互联网行业找工作就上拉勾网。

拉勾 - 百度快照 - 228条评价

@拉勾教育


为了让搜索引擎更好地识别页面，除了描述信息之外还可以使用关键字，这样即使页面其他地方没有包含搜索内容，也可以被搜索到（当然搜索引擎有自己的权重和算法，如果滥用关键字是会被降权的，比如 Google 引擎就会对堆砌大量相同关键词的网页进行惩罚，降低它被搜索到的权重）。

当我们搜索关键字“垂直互联网招聘”的时候搜索结果会显示拉勾网的信息，虽然显示的搜索内容上并没有看到“垂直互联网招聘”字样，这就是因为拉勾网页面中设置了这个关键字。




2019年12月19日 - [拉勾](#)表示提高反馈效率,做更好的招聘工作,为人才提供更优质的工作机会,为企业提供更加便捷的招聘工具,立足于人才数据,做好[互联网垂直](#)产业的招聘,是[拉勾](#)的使命和不懈努...  
七芊 - 百度快照

靠谱的[互联网招聘网站](#)有哪些? - 知乎



[拉勾](#)是最大的互联网[垂直招聘](#)网站。提供海量互联网求职者信息和[互联网招聘](#)信息。如果想要从事互联网相关工作或是招聘相关人才,这是首选。但其他行业的信息则相对匮乏...  
知乎 - 百度快照

[拉勾网](#)-专业的[互联网招聘平台](#)



[拉勾网](#)是专业的互联网行业招聘平台。致力于提供真实可靠的[互联网招聘](#)信息。工资不面议。找工作、要招聘、搜人才就来[拉勾网](#)。互联网行业找工作就上[拉勾网](#)。  
拉勾 - 百度快照 - 228条评价

[网络招聘渠道对比垂直类](#) - 百度文库

@拉勾教育

对应代码如下:

```
<meta content="拉勾,拉勾网,拉勾招聘,拉钩, 拉钩网 , 互联网招聘,拉勾互联网招聘, 移动互联网招聘, 垂直互联网招聘, 微信招聘, 微博招聘, 拉勾官网, 拉勾百科,跳槽, 高薪职位, 互联网圈子, IT招聘, 职场招聘, 猎头招聘,O2O招聘, L
```

在实际工作中,推荐使用一些关键字工具来挑选,比如 [Google Trends](#)、[站长工具](#)。下图是我使用站长工具搜索“招聘”关键字得到的结果,可以看到得到了相当关键的一些信息,比如全网搜索指数、关键词特点。

序号	关键词	全网指数	长尾词数	竞价数	sem预估价格	搜索量	关键词特点	竞争度
1	招聘	1000093	1243181	0		0	周 夜	困难
2	招聘信息	152629	93063	0		0	潜 周 夜	困难
3	招聘网	119399	105438	0		0	周 夜	困难
4	智联招聘	88134	7928	0		0	潜 移 周 夜	困难
5	58同城招聘	86806	5165	0		0	移 周 夜	容易
6	最新招聘	74018	27181	0		0	周 夜	困难
7	招聘兼职	36678	15897	0		0	周 夜	困难
8	教师招聘	30928	32766	0		0	周 夜	困难
9	最新招聘信息	30467	6181	0		0	周 夜	困难
10	事业单位招聘	18910	7393	0		0	周 夜	困难
11	人才招聘	18296	35577	0		0	周 夜	困难
12	招聘会	17488	48424	0		0	周 夜	困难
13	智联招聘网	16094	1199	0		0	潜 移 周 夜	困难
14	医院招聘	13667	14493	0		0	周 夜	困难
15	司机招聘	12758	23788	0		0	周 夜	困难
16	兼职招聘	12534	51811	0		0	周 夜	困难
17	公司招聘	12512	23877	0		0	周 夜	困难
18	58同城网招聘找工作	11813	10	0		0	周 夜	困难

@拉勾教育

link 标签: 减少重复

有时候为了用户访问方便或者出于历史原因,对于同一个页面会有多个网址,又或者存在某些重定向页面,比如:

- <https://lagou.com/a.html>
- <https://lagou.com/detail?id=abcd>

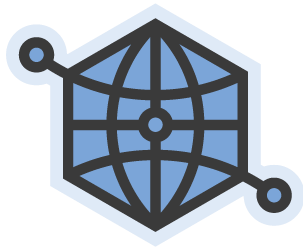
那么在这些页面中可以这样设置:

```
<link href="https://lagou.com/a.html" rel="canonical">
```

这样可以让搜索引擎避免花费时间抓取重复网页。不过需要注意的是，它还有个限制条件，那就是指向的网站不允许跨域。

当然，要合并网址还有其他方式，比如使用站点地图，或者在 HTTP 请求响应头部添加 `rel="canonical"`。这里，我就不展开介绍了，道理都是相通的，你平时可以多探索和实践。

**延伸内容：OGP（开放图表协议）**



好了，前面我们说了 HTML5 标准的一些标签和属性，下面再延伸说一说基于 `meta` 标签扩展属性值实现的第三方协议——OGP（Open Graph Protocol，开放图表协议）。

OGP 是 Facebook 公司在 2010 年提出的，目的是通过增加文档信息来提升社交网页在被分享时的预览效果。你只需要在一些分享页面中添加一些 `meta` 标签及属性，支持 OGP 协议的社交网站就会在解析页面时生成丰富的预览信息，比如站点名称、网页作者、预览图片。具体预览效果会因各个网站而有所变化。

下面是微信文章支持 OGP 协议的代码，可以看到通过 `meta` 标签属性值声明了：网址、预览图片、描述信息、站点名称、网页类型和作者信息。

```
![1583480543843-477274458e5be00b.png] (https://s0.lgstatic.com/i/image/M00/07/0F/CgqCH1647neAc1fJAACYggDXkeE601.png)
```

现在百度已经宣布支持，微信文章的不少页面上也添加了相关标签属性，有兴趣的话你可以查看官方网站：<https://ogp.me/>。

### 总结

本课时，我从交互实现、性能优化、搜索优化场景出发，分别讲解了 `meta` 标签、`title` 标签、`link` 标签，以及 `script` 标签在这些场景中的重要作用，希望这些内容你都能有效地应用到工作场景中，不再只是了解，而是能够熟练运用。

最后布置一道思考题：说一说你还知道哪些“看不见”的标签及用法？

### 社群福利

5月20日前，加入社群获得以下福利：

- ① 讲师交流： 订阅后可加入「 前端进阶交流群 」与讲师直面交流
- ② 独家资料： 进群后领取 「 前端高手进阶课程PPT 」
- ③ 加餐内容： 社群专属直播 「2020 大厂青睐的前端必备技能」

——> [仅限 500 人，戳此加入](#) <——