

前端复习

防抖与节流

segmentfault.com

https://segmentfault.com/a/1190000018445196

函数防抖的应用场景

连续的事件，只需触发一次回调的场景有：

- 搜索框搜索输入。只需用户最后一次输入完，再发送请求
- 手机号、邮箱验证输入检测
- 窗口大小 Resize。只需窗口调整完成后，计算窗口大小。防止重复渲染。

函数节流的应用场景

间隔一段时间执行一次回调的场景有：

- 滚动加载，加载更多或滚到底部监听
- 谷歌搜索框，搜索联想功能
- 高频点击提交，表单重复提交

浏览器缓存

一文读懂前端缓存 - 掘金

大家都知道缓存的英文叫做 cache。但我发现一个有趣的现象：这个单词在不同人的口中有不同的读音。为了全面了解缓存，我们得先从读音开始，这样能够在和其他同事(例如 PM)交(zhuāng)流(bì)时体现自己的修(bì)养(gé)。友情提示：文章有些长，您可能需要分批次读完...

 juejin.cn

1. 强制缓存

Expires 存储缓存到期的时间，(当前时间 + 缓存时间) 缓存到期才请求服务器，否则取出缓存

```
Expires: Thu, 10 Nov 2017 08:45:11 GMT
```

JavaScript

Cache-control 表示资源缓存的最大有效期

```
Cache-control: max-age=2592000
```

JavaScript

前者绝对时间，后者相对时间

2. 协商缓存

Last-Modified & If-Modified-Since

1. 服务器通过 `Last-Modified` 字段告知客户端，资源最后一次被修改的时间，例如

```
Last-Modified: Mon, 10 Nov 2018 09:10:11 GMT
```

复制代码

2. 浏览器将这个值和内容一起记录在缓存数据库中。
3. 下一次请求相同资源时时，浏览器从自己的缓存中找出“不确定是否过期的”缓存。因此在请求头中将上次的 `Last-Modified` 的值写入到请求头的 `If-Modified-Since` 字段
4. 服务器会将 `If-Modified-Since` 的值与 `Last-Modified` 字段进行对比。如果相等，则表示未修改，响应 304；反之，则表示修改了，响应 200 状态码，并返回数据。

但是他还是有一定缺陷的：

- 如果资源更新的速度是秒以下单位，那么该缓存是不能被使用的，因为它的时间单位最低是秒。
- 如果文件是通过服务器动态生成的，那么该方法的更新时间永远是生成的时间，尽管文件可能没有变化，所以起不到缓存的作用。

Etag & If-None-Match

为了解决上述问题，出现了一组新的字段 `Etag` 和 `If-None-Match`

`Etag` 存储的是文件的特殊标识(一般都是 hash 生成的)，服务器存储着文件的 `Etag` 字段。之后的流程和 `Last-Modified` 一致，只是 `Last-Modified` 字段和它所表示的更新时间改变成了 `Etag` 字段和它所表示的文件 hash，把 `If-Modified-Since` 变成了 `If-None-Match`。服务器同样进行比较，命中返回 304，不命中返回新资源和 200。

Etag 的优先级高于 Last-Modified