在浏览器中缓存分为 **强缓存** 和 **协商缓存** 两种：

**强缓存**：  
强缓存的意思就是资源的过期时间（Expires）或者是缓存时间（Cache-Control中设置的时间）是在当前时间之后，也就是说文件或资源没有过期，那么就直接从缓存中获取了。强缓存的HTTP状态码为200。  
**协商缓存**：  
协商缓存就是指已经超出缓存时间了，然后向服务器发送一个HTTP请求，服务器说还可以使用原来的缓存，然后就又从缓存中获取文件或资源了。协商缓存的HTTP状态码为304。

当强缓存与协商缓存都没有命中的时候，那么就要从服务器重新获取资源了。  
首先来一张图片，这张图片是访问www.jd.com的时候其主页的一个文件的头信息：



**强缓存**  
在http1.0协议中提出了一个名为**Expires**的头属性，这个属性的值代表着资源过期时间，它是一个GMT格式的绝对时间，由服务器返回，在上面那个请求中可以看到在Response Headers中有这么一句：

*http header*

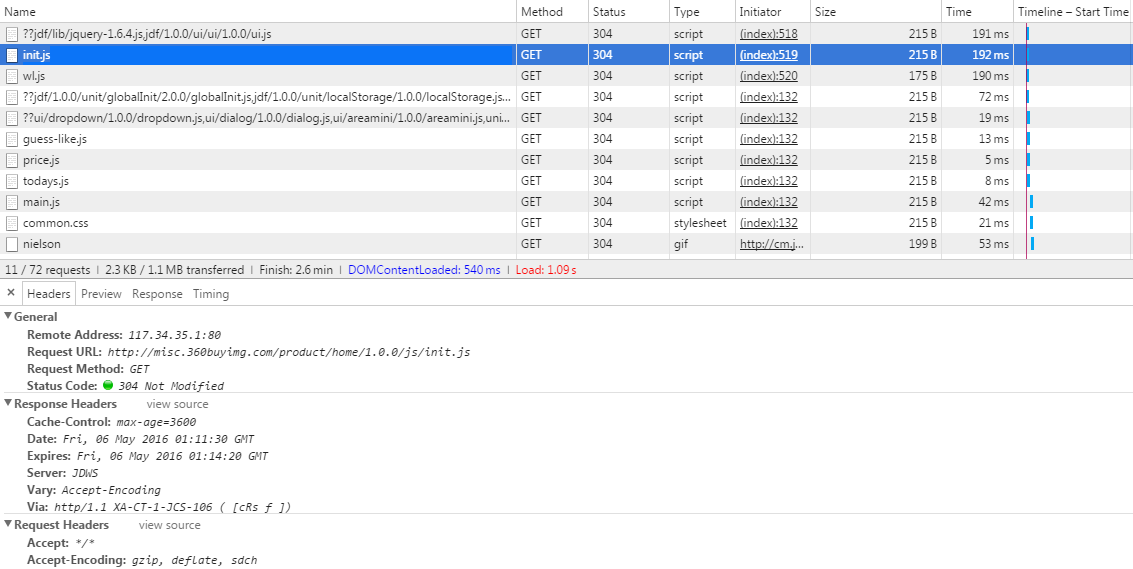
Expires:Mon,  20  Apr  2026  18:11:31  GMT

这就是过期时间，如果当前时间在这个时间之前，那么就会直接从缓存中获取。  
Expires是较老的强缓存管理属性，由于它是服务器返回的一个绝对时间，在服务器时间与客户端时间相差较大时，缓存管理容易出现问题，比如随意修改下客户端时间，就能影响缓存命中的结果。所以在http1.1的时候，提出了一个新的header，就是Cache-Control。  
**Cache-Control**是一个相对时间，以秒为单位，指从缓存开始时间过后多长时间，比如说上面的请求中的:

*http header*

Cache-Control:max-age=315360000

上面说的这两个属性：Expires和Cache-Control可以使用一个，也可以都使用，当都使用的时候Cache-Control的优先级要高于Expires。  
**协商缓存**  
当浏览器对某个资源的请求没有命中强缓存，就会发一个请求到服务器，验证协商缓存是否命中，如果协商缓存命中，请求响应返回的http状态为304并且会显示一个Not Modified的字符串。



可以看到上面这张图片中的status为304，也就是命中了协商缓存。  
协商缓存利用两对头属性来管理缓存：

① **Last-Modified** 和 **If-Modified-Since**  
② **ETag** 和 **If-None-Match**

**一、使用Last-Modified 和 If-Modified-Since 的原理**：  
① 当浏览器第一次向服务器请求某一个资源的时候，服务器会在返回的头信息中添加一个Last-Modified属性，这个属性的值就是当前资源在服务器的最后修改时间。  
② 当浏览器再一次向服务器请求这个资源的时候，会在请求头中添加一个If-Modified-Since属性，这个属性的值就是上一次服务器返回的Last-Modified的值。  
③ 服务器根据传递过来的If-Modified-Since值和服务器上面资源的最后修改时间来进行判断，如果没有修改，那么就返回304状态码，如果最后修改时间不一样，那么就返回新的资源和新的Last-Modified值。  
④ 如果浏览器收到了304响应，那么就直接使用缓存中的资源。  
**二、使用ETag 和 If-None-Match 的原理**：  
ETag就是服务器对资源生成的一个唯一标识，只要资源有变化，这个值就会不一样。  
和上面的过程类似，当浏览器第一次请求资源的时候，服务器会返回一个ETag，当浏览器再次请求的时候，就会在头部添加一个If-None-Match属性，这个属性的值就是ETag的值，然后服务器再次对资源生成ETag，与传过来的If-None-Match值进行比较，如果相同代码资源没有变化，那么返回304状态码，否则就会返回资源。  
和使用Last-Modified不同的是，ETag无论资源有没有修改，在服务器的响应报文中都会有这个属性，而Last-Modified在资源最后修改时间没有变化的时候服务器的响应报文中不会返回这个属性。

分布式系统里多台机器间文件的Last-Modified必须保持一致，以免负载均衡到不同机器导致比对失败；  
分布式系统尽量关闭掉ETag(每台机器生成的ETag都会不一样）；

协商缓存需要配合强缓存使用，否则协商缓存没有任何意义。