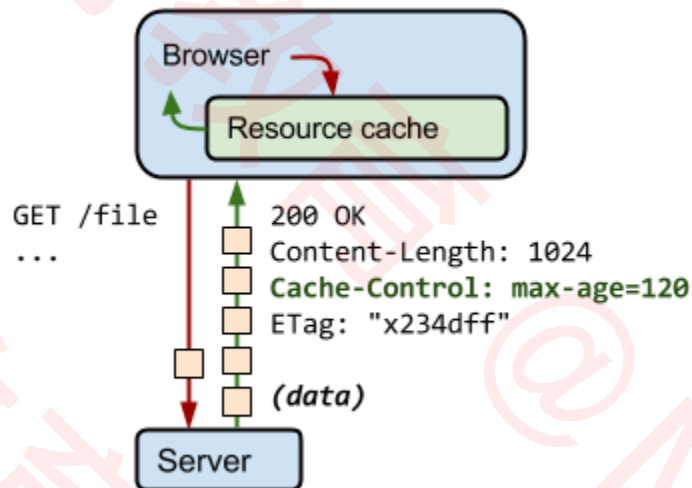


# 缓存 (Cache)

- Cache的发音跟Cash (现金) 一样：美 /kæʃ/



- 实际上，HTTP的缓存机制远远比上图的流程要复杂
- 通常会缓存的情况是：GET请求 + 静态资源（比如HTML、CSS、JS、图片等）
- Ctrl + F5：可以强制刷新缓存

# 缓存 - 响应头

- Pragma: 作用类似于Cache-Control, HTTP/1.0的产物
- Expires: 缓存的过期时间 (GMT格式时间), HTTP/1.0的产物
- Cache-Control: 设置缓存策略
  - no-storage: 不缓存数据到本地
  - public: 允许用户、代理服务器缓存数据到本地
  - private: 只允许用户缓存数据到本地
  - max-age: 缓存的有效时间 (多长时间不过期), 单位秒
  - no-cache: 每次需要发请求给服务器询问缓存是否有变化, 再来决定如何使用缓存
- 优先级: Pragma > Cache-Control > Expires

# 缓存 - 响应头

- Last-Modified: 资源的最后一次修改时间
- ETag: 资源的唯一标识 (根据文件内容计算出来的摘要值)
- 优先级: ETag > Last-Modified

# 缓存 - 请求头

## ■ If-None-Match

- 如果上一次的响应头中**有**ETag，就会将ETag的值作为请求头的值
- 如果服务器发现资源的最新摘要值跟If-None-Match不匹配，就会返回新的资源 (200 OK)
- 否则，就不会返回资源的具体数据 (304 Not Modified)

## ■ If-Modified-Since

- 如果上一次的响应头中**没有**ETag，有Last-Modified，就会将Last-Modified的值作为请求头的值
- 如果服务器发现资源的最后一次修改时间晚于If-Modified-Since，就会返回新的资源 (200 OK)
- 否则，就不会返回资源的具体数据 (304 Not Modified)

# 缓存 – Last-Modified vs ETag

## ■ Last-Modified的缺陷

- 只能精确到秒级别，如果资源在1秒内被修改了，客户端将无法获取最新的资源数据
- 如果某些资源被修改了（最后一次修改时间发生了变化），但是内容并没有任何变化
- ✓ 会导致相同数据重复传输，没有使用到缓存

## ■ ETag可以办到

- 只要资源的内容没有变化，就不会重复传输资源数据
- 只要资源的内容发生了变化，就会返回最新的资源数据给客户端

# 缓存的使用流程

