



链滴

# HTML 图片顺序加载

作者: [Sysecho](#)

原文链接: <https://ld246.com/article/1498742940041>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## JavaScript 中图片顺序加载，防止页面阻塞

- 浏览器一般只能同时存在4-8个连接 (chrome6个),为了使得图片和页面渲染同时进行，需要保证图加载只有一个连接使用，即顺序加载：

```
$(function(){
    //获取页面所有图片
    var images = document.images;
    //图片异步加载
    preload(images,0);
});

//图片异步加载
function preload(images, index) {
    index = index || 0;
    if (images && images.length > index) {
        var img = images[index];
        //图片加载完成后回到函数中开始加载下一张图片
        img.onload = function() {
            preload(images, index + 1);
        };
        var src = "download?id="+images[index].id+"&isThumball=false";
        img.src = src;
    }
}
```

## HTML 代码

```
<img src="" id="{drawingImg.id}" alt="{drawingImg.imageName}"></div>
```

## 图片下载方法DEMO

```
@RequestMapping("download")
public void downFile(@RequestParam("id") String id, HttpServletResponse response) throws
xception {
    List<ImageVO> list = imageService.getImageInfoBySrctypeAndDataId(SrcTypeEnum.MATE
IAL.getValue(), id);
    OutputStream out = new ByteArrayOutputStream();
    String imageld = "";
    String fileName = String.valueOf(new Date().getTime()) + ".jpg";
    if (list.size() > 0) {
        ImageVO image = list.get(0);
        imageld = image.getImageld();
        image.getName();
        fileName = image.getName();
    }
    out = imageService.downloadImage(imageld, CompressLevel.STANDARD, true);
    response.setContentType("application/octet-stream");
```

```
// 设置下载文件名
response.addHeader("Content-Disposition", "attachment; filename=\"" + fileName + "\"");
OutputStream outstrm = response.getOutputStream();
outstrm.write(((ByteArrayOutputStream) out).toByteArray());
outstrm.flush();
out.close();
outstrm.close();
}
```