网站优化：图片异步加载

随着网站的文章越来越多，网站的图片也不知不觉的多了起来，图片多起来带来的问题就是访问的人多的时候会导致页面加载速度越来越慢，这严重影响了网站的用户体验，所以网站图片异步加载势在必行。

图片异步加载就是图片只有在视野范围内才加载，没出现在范围内的图片就暂不加载，等用户滑动滚动条时再逐步加载。当然第一想到的就是去网站上寻找相应的插件，我找到一个名为lazyload.js的插件，它是一款jQuery的插件，全称是jquery.lazyload.js，当时把网站的所有图片按照lazyload.js插件要求设置好，但是我发现了有个bug，就是我博客右侧的高度如果小于左侧太多，会导致右侧的部分图片一直无法加载，不管你怎么移动浏览器滑动条，就是不加载！于是只能换别的插件了，我在网上找到一款名为asyncLoad.js的异步加载插件，这个插件是个人写的，试了一些能够解决lazyload.js插件出现的问题，但是有一点我不满意的是asyncLoad.js插件在图片还没加载的时候没有图片占位符，看起来像下面一样：

对于我这种"强迫症"的人来说肯定不行，那咋办呢？于是只能自己弄了，经过研究，最终还是弄好了。我在asyncLoad.js里面加了一个名为placeholder的参数，这个参数可以设置默认的图片占位符，也就是图片在加载之前会显示这个占位符。完整的代码如下：

```
(function(a) {
    a.fn.scrollLoading = function(b) {
        var c = {
```

```
        attr: "data-url",
        placeholder: "
EAAAICTAEAOw==",
        container: a(window),
        callback: a.noop
    };
    var d = a.extend({},
    c, b || {});
    d.cache = [];
    a(this).each(function() {
        var h = this.nodeName.toLowerCase(),
        g = a(this).attr(d.attr);
        if (a(this).attr("src") === undefined || a(this).attr("src") === false) {
            if (a(this).is("img")) {
                a(this).attr("src", d.placeholder);
            }
        }
        var i = {
            obj: a(this),
            tag: h,
            url: g
        };
        d.cache.push(i)
    });
    var f = function(g) {
        if (a.isFunction(d.callback)) {
            d.callback.call(g.get(0))
        }
    };
    var e = function() {
        var g = d.container.height();
        if (d.container.get(0) === window) {
            contop = a(window).scrollTop()
        } else {
            contop = d.container.offset().top
        }
        a.each(d.cache,
        function(m, n) {
            var p = n.obj,
            j = n.tag,
            k = n.url,
            l, h;
            if (p) {
                l = p.offset().top - contop,
                h = l + p.height();
                if ((l >= 0 && l < g) || (h > 0 && h <= g)) {
```

```
                if (k) {
                    if (j === "img") {
                        f(p.attr("src", k))
                    } else {
                        p.load(k, {},
                        function() {
                            f(p)
                        })
                    }
                } else {
                    f(p)
                }
                n.obj = null
            }
        }
    })
};
e();
d.container.bind("scroll", e)
}
})(jQuery);
```

占位符图片我使用的是base64加密的图片，因为这个可以帮助我们减少图片请求。各个参数的含义介绍：

参数	默认	释义
attr	data-url	获取元素加载地址的属性名
placeholder	见代码	占位图片
container	$(window)	滚动的容器。默认为$(window)，也就是默认的网页滚动。
callback	$.noop	回调。元素动态加载完毕后执行的回调函数。其中回调函数的上下文this就是当前DOM元素。注意：如果无法获取元素加载地址，则不执行动态加载，但是会触发回调。在某些需求下，您可以缺省url值，仅仅触发回调。

实例

异步加载全部图片

所有需要异步加载的图片URL位置全部放在data-url属性里面，如下：

```
<img data-url="/pic/asynLoad.png" />
```

然后我们在网站加入以下js代码：

```
$("img").scrollLoading();
```

这样就可以异步加载图片了。

异步加载某特定图片

如果你需要对某一个class（或者是id）的图片异步加载，而不是全部的img标签，可以如下写：

```
<img class="asyn" data-url="/pic/asynLoad.png" />
<img id="asynId" data-url="/pic/asynLoad.png" />
```

js代码如下：

```
$(".asyn").scrollLoading();
$("#asynId").scrollLoading();
```

设置占位图片

默认的占位是一张base64加密的图片，如果你觉得这个比较难看，你可以设置成一幅动态加载的图片，如下：

```
$("img").scrollLoading({
 placeholder: "/wp-content/themes/yusi2.0/img/loading.gif"
});
```

container使用

```
$("img").scrollLoading({
 container: $("#iteblog")
});
```

callback使用

```
$("img").scrollLoading({
 callback: function() {
  this.style.border = "1px solid #ff6600";
 }
});
```

设置图片加载效果

本插件默认是没有图片加载特效的，不过我们可以通过jQuery实现，如下：

```
$("img").load(function () {
 //图片默认隐藏
 $(this).hide();
 //使用fadeIn特效
 $(this).stop().fadeIn("5000");
});
```

这样我们就为图片的异步加载使用fadeIn特效显示。好了，很晚了，该洗洗睡了，如果使用有啥问题请留言。

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。
本文链接: 【】（ ）