# WEB HW4 图片加载

# 服务端运行: node server.js

# 懒加载在二级页面"壁纸"(wallpaper.html)体现

当一个网站的加载图片过多时需要懒加载的协助,页面图片多时,在首次载入时一次性加载会耗费时间长,使用懒加载可以使页面加载速度快、减轻服务器的压力、节省流量。

#### 实现的思路:

给目标元素指定一张占位图(尽量小),将真实的图片链接存储在自定义属性中(data-src), 监听scroll事件,当图片出现在可视区时,将data-src值赋值给src,加载图片,在图片未加载时,用alt 属性,在图像无法显示时的替代文本。

## 相关代码

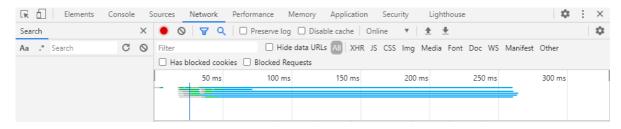
```
window.onload = function() {
        var photos = document.getElementsByClassName("photos");
        for (let i = 0; i < photos.length; <math>i++) {
            photos[i].dataset.isLoaded = "0";
        lazyload();
   window.onscroll = function() {
        lazyload();
   function lazyload() {
        var photos = document.getElementsByClassName("photos");
        var clientHeight = document.documentElement.clientHeight;
        for (let i = 0; i < photos.length; <math>i++) {
            if (photos[i].dataset.isLoaded == "0" && photos[i].getClientRects()
[0].top <= clientHeight + 40) {
                photos[i].dataset.isLoaded = "1";
                let src = photos[i].src;
                let data_src = photos[i].dataset.src;
                photos[i].src = data_src;
                console.log(src);
            }
        }
   }
```

#### 监听scroll事件

isLoaded属性表示图片是否被加载过,维护这个属性是为了防止图片在页面滚动时反复加载。 若图片未加载过且进入了加载区域,就把真实图片路径'data-src'赋值给图片的src,达到加载目的。

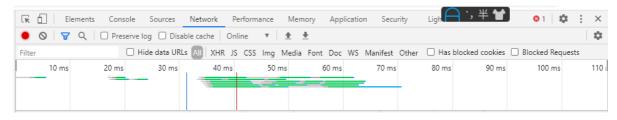
## 实验效果

copy.html是与wallpaper.html布局了相同图片但没有使用懒加载的对比网页:



可以看到,没有懒加载的情况下网页加载耗时近250+ms

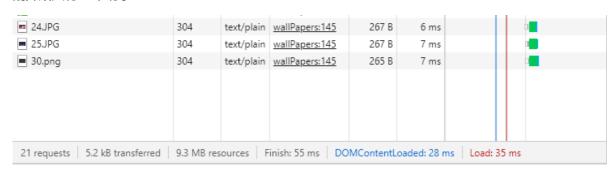
### 使用懒加载后,仅耗时70ms左右:



网页加载速度明显提升。

### 网页向下滚动时, 图片逐渐加载:

### 刚开始只有21个请求



全部加载需要37个请求

■ 24JPG	304	jpeg	wallPapers:145	267 B	6 ms	1
■ 25.JPG	304	jpeg	wallPapers:145	267 B	5 ms	1
■ 30.png	304	png	wallPapers:145	265 B	4 ms	1
	304	jpeg	wallPapers:145	267 B	4 ms	1
15.JPG	304	jpeg	wallPapers:145	267 B	4 ms	1
■ 35.jpg	304	jpeg	wallPapers:145	268 B	5 ms	1
■ 16.JPG	304	jpeg	wallPapers:145	267 B	7 ms	- 1
■ 10.JPG	304	jpeg	wallPapers:145	267 B	6 ms	1
■ 36.jpg	304	ipea	wallPapers:145	268 B	9 ms	1
■ 1JPG	304 N	ot Modifie	allPapers:145	267 B	4 ms	1
<ul> <li>37.jpg</li> </ul>	304	jpeg	wallPapers:145	268 B	7 ms	1
■ 14.JPG	304	jpeg	wallPapers:145	267 B	4 ms	1
■ 12.JPG	304	jpeg	wallPapers:145	267 B	4 ms	1
■ 38.jpg	304	jpeg	wallPapers:145	267 B	7 ms	
3JPG	304	jpeg	wallPapers:145	267 B	3 ms	
■ 39.jpg	304	jpeg	wallPapers:145	267 B	3 ms	
2.JPG	304	jpeg	wallPapers:145	267 B	5 ms	
■ 13.JPG	304	jpeg	wallPapers:145	268 B	6 ms	
■ 40.jpg	304	jpeg	wallPapers:145	268 B	52 ms	