



从前端小工到中级工程师的必备技能

加载相关前端性能优化

1、lazyload

延迟加载

lazyload

1 图片 lazyload

2 组件 lazyload

1.1、图片 lazyload —— showcase7

图片一般是页面最大的资源，所以非首屏延迟加载很重要

实例解析防抖动（Debouncing）和节流阀（Throttling）

<http://www.css88.com/archives/7010>

顺便介绍一下图片压缩工具

<https://github.com/miniflycn/optimg>

1.2、组件 lazyload

既然图片可以异步加载，别的资源当然也可以同样异步

AMD异步

通过控制异步 require 时机来让组件异步化

```
1  define('my-mod', function () {  
2      |      require(['async-mod'], function (mod) {  
3          |          mod.init()  
4      |      })  
5  })
```


webpack 异步加载——showcase8

主要指 require.ensure 机制

```
1  require(['./async/index'], function (require) {  
2    require('./async/index').init()  
3  })
```

异步标签

有兴趣可以玩玩，将组件都标签化后，直接对标签异步

```
1 | <my-component async/>
```

2、小图 -> 大图

网速很差的时候，可以先用小图让用户觉得没有白屏

小图 -> 大图

- 1 小图是大图的缩略图，然后放到大图的大小
- 2 大图保存成图片交错模式
- 3 小图 onload 之后再加载大图，加载完成后直接替换小图

3、预加载技术

预先把资源准备好，等用户操作的时候再使用

预加载技术

1 下一页预加载





































2 页面数据预加载

3.1、下一页数据预加载

很容易理解，就是空闲时候加载下一页数据

3.2、页面数据预加载

尽早发出数据请求

 diy.html	200	docu...	Other	3.4 KB	15...			
 main.css	200	styles...	diy.html	783 B	15...			
 require.min.js	200	script	diy.html	(from memory...	0 ms			
 main.css	200	styles...	diy.html	979 B	18...			
 main.css	200	styles...	diy.html	4.3 KB	42...			
 logo.png	200	png	diy.html	4.8 KB	10...			
 main.js	200	script	require.min.j...	876 B	15...			
 main.js	200	script	require.min.j...	510 B	27...			
 factory.js	200	script	require.min.j...	818 B	38...			
 Q.js	200	script	require.min.j...	942 B	28...			
 db.diy.js	200	script	require.min.j...	427 B	23...			
 third.js	200	script	require.min.j...	973 B	17...			
 Q.js	200	script	require.min.j...	50.3 KB	14...			
 db.js	200	script	require.min.j...	3.3 KB	28...			
 jquery.min.js	200	script	require.min.j...	90.6 KB	23...			
 db.core.js	200	script	require.min.j...	6.3 KB	7 ms			
 highlight.min.js	200	script	main.js:15	(from disk cac...	7 ms			
 index_json	200	xhr	db.core.js:61	12.8 KB	87...			

diy.html	200	docu...	Other	3.8 KB	17...	
main.css	200	styles...	diy.html	783 B	16...	
require.min.js	200	script	diy.html	(from memory...	0 ms	
main.css	200	styles...	diy.html	979 B	17...	
main.css	200	styles...	diy.html	4.3 KB	36...	
index_json?callback=__PRELOAD__CB__0	200	script	diy.html	12.8 KB	15...	
logo.png	200	png	diy.html	4.8 KB	9 ms	
main.js	200	script	require.min.j...	876 B	22...	
main.js	200	script	require.min.j...	510 B	29...	
factory.js	200	script	require.min.j...	818 B	45...	
Q.js	200	script	require.min.j...	942 B	29...	
db.diy.js	200	script	require.min.j...	450 B	30...	
third.js	200	script	require.min.j...	973 B	22...	
Q.js	200	script	require.min.j...	50.3 KB	17...	
db.js	200	script	require.min.j...	3.3 KB	31...	
jquery.min.js	200	script	require.min.j...	90.6 KB	21...	
db.core.js	200	script	require.min.j...	6.3 KB	8 ms	
highlight.min.js	200	script	main.js:15	(from disk cac...	8 ms	

4、Inline

将小的资源合并到页面，例如 Base64 图



THANKS!



扫码了解更多