

页面

加快首屏渲染

· SSR

1. 服务端渲染对搜索引擎的爬取有着天然的优势（SEO）
2. 直接拿到 html，降低首屏渲染时间

· PWA (service work)

1. 用户可以用手机屏幕启动应用，即使在离线状态或者弱网下，通过事先缓存的资源，也可正常加载运行当前应用，可以完全消除对网络的依赖，从而给用户非常可靠的体验。
2. 因为预先缓存了资源，部分资源无须经过网络，即秒开页面。
3. 和移动设备上的原生应用一样，具有沉浸式的用户体验。

请求 -> SW -> http cache

SSR、PWA 没有任何一种技术的银弹，技术选型都是要综合多种因素（人力，财力，开发成本，开发周期...）决定的，SSR 需要前端有运维能力，PWA 也有兼容性问题。

· 懒加载

指的是长网页中延迟加载特定元素，可以是图片，也可以是 JS/CSS 文件，
好处：可以减少当前屏无效资源的加载。

```
1 ES 新的提案: import('./xxx.js').then()  
2 // 需要的时候才去加载  
3 // 要是没有 import(), 该怎么实现懒加载 js  
4  
5 // 被 webpack 打包完 也是下面这样子的  
6 dom.onClick = function() {  
7   const scriptNode = document.createElement('script');  
8   script.src = './index.js';  
9   document.body.appendChild(script);  
10 }
```

预加载

使用预加载让浏览器来预先加载某些资源（比如图片、JS/CSS/模板），而这些资源是在将来才会被使用到的。简单来说，就是将所需资源提前加载到浏览器本地，这样后面在需要使用的时候就可以直接从浏览器缓存中取了，而不用再重新开始加载。

好处：减少用户后续加载资源等待的时间。

HTML 标签

```
1 
```

使用 Image 对象

```
1 var image= new Image();
2 image.src=https://img10.360buyimg.com/n4/g7/M03/08/0D/rBEHZlBzwZwIAA
3 AAAAI4s0vIiLkAABpMQDf8E4AAjjI749.jpg
```

使用 preload, prefetch 和 preconnect

```
1 <link rel="preload" href="src/style.css" as="style">
2 // 使用 preload 可以对当前页面所需的脚本、样式等资源进行预加载，而无需等到解析到 script 和
   link 标签时才进行加载
3 <link rel="prefetch" href="src/image.png">
4 // 其他页面会用到的资源，可能是下一个页面
5 <link rel="dns-prefetch" href="https://my.com">
6 // DNS 预解析
7 <link rel="preconnect" href="https://my.com" crossorigin>
8 // 预先与服务器握手
9 <link rel="prerender" href="https://my.com">
10 // 后台预渲染指定的 页面
```

接口缓存

一些非实时的接口，请求完数据可以考虑缓存（数据存在 本地缓存里面）