

目录	
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏 “一片空白” 最近阅读	
18 如何应对首屏 “一片空白”（下）	
19 不容小觑的 DOM 性能优化	

← 慕课专栏	≡ 你不知道的前端性能优化技巧 / 17 如何应对首屏 “一片空白” （上）
目录	提升了用户体验。
第1章 优化的概念	实现方案
1 开篇词：你的前端性能还能再抢救一下	关于骨架屏的实现方案其实非常多，对应到具体的React、Vue框架中也有相应的解决方案，下面我们就来介绍几种常用的骨架屏实现方案。
2 解读雅虎35条军规（上）	绘制静态骨架屏
3 解读雅虎35条军规（下）	最简单的方案就是就是根据首页的大致轮廓，然后自己用 Photoshop 绘制静态的骨架屏图片。前面我们讲到base64 格式的图片可以直接插入到 HTML 文档中，而且可以节省一次 HTTP 请求，所以这里我们把骨架屏的静态图片直接转换成 base64 格式然后插入到 HTML 代码中，使骨架屏能够更快地显示。具体代码如下：
4 你要不要看这些优化指标？	<pre><!DOCTYPE html> <html> <head> <title>骨架屏</title> <meta charset="utf-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width, initial-scale=1"> </head> <body> <section id="root"> </section> <script> // 相关JavaScript代码 </script> </body> </html></pre>
第2章 性能工具介绍	如上，我们先把骨架屏静态图片插入到了 HTML 当中，然后当资源加载完毕，我们再用真正的内容替换掉骨架屏，这里替换时机就是当所有资源都加载完毕，对应的事件就是 onLoad。我们只要监听这个事件，当它完成进行替换即可。如果是 Vue 的项目，我们则可以选择 mounted 钩子当中进行替换；如果是 React 项目，我们则可以选择 componentDidMount 钩子当中进行替换。
5 性能优化百宝箱（上）	当然这个方法是比较笨的方法，而且一旦我们的首屏发生变化，我们就需要重新绘制，灵活度很差，所以这种方法我们一般不推荐使用。
6 性能优化百宝箱（下）	代码方式绘制骨架屏
第3章 网络部分	用代码方式绘制骨架屏也不难，而且可定制化程度高，我们可以根据自己的需要，自行选择自己需要的样式。首先我们需要创建与首屏内容轮廓大致类似的 HTML 结构，然后使用 CSS 为这些轮廓填充背景色。如下：
7 聊聊 DNS Prefetch	<pre><!DOCTYPE html> <html> <head> <title>骨架屏</title> <style type="text/css"></pre>
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏 “一片空白” 最近阅读	
18 如何应对首屏 “一片空白” （下）	
19 不容小觑的 DOM 性能优化	

<div><div>← 慕课专栏</div><div>☰ 你不知道的前端性能优化技巧 / 17 如何应对首屏 “一片空白” （上）</div></div>	
目录	
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标?	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏 “一片空白” 最近阅读	<pre>.framework .framework-head, .framework .framework-title, .framework .framework-content { background: rgb(194, 207, 214); } .framework-head { width: 100px; height: 100px; float: left; } .framework-body { margin-left: 110px; } .framework-title { width: 500px; height: 60px; } .framework-content { width: 260px; height: 30px; margin-top: 10px; } </style> </head> <body> <div class="framework"> <div class="framework-head"></div> <div class="framework-body"> <div class="framework-title"></div> <div class="framework-content"></div> </div> </div> </body> </html></pre>
18 如何应对首屏 “一片空白” （下）	
19 不容小觑的 DOM 性能优化	

上面就是一个简单的骨架屏效果，如下图：



上面的这个是静态效果，我们还可以通过改动 CSS，把它变成动态的效果，原有 HTML 结构不变，这里我们把 CSS 代码替换如下：

```
.framework {
padding: 10px;
}

.framework .framework-head,
.framework .framework-title,
.framework .framework-content {
background: rgb(194, 207, 214);
background-image: linear-gradient(90deg, rgba(255, 255, 255, 0.15) 25%
```

<div>← 慕课专栏</div> <div>☰ 你不知道的前端性能优化技巧 / 17 如何应对首屏 “一片空白” （上）</div>	
<div>目录</div>	
<div>第1章 优化的概念</div>	
<div>1 开篇词：你的前端性能还能再抢救一下</div>	
<div>2 解读雅虎35条军规（上）</div>	<pre>.framework-head { width: 100px; height: 100px; float: left; }</pre>
<div>3 解读雅虎35条军规（下）</div>	<pre>.framework-body { margin-left: 110px; }</pre>
<div>4 你要不要看这些优化指标?</div>	<pre>.framework-title { width: 500px; height: 60px; }</pre>
<div>第2章 性能工具介绍</div>	
<div>5 性能优化百宝箱（上）</div>	<pre>.framework-content { width: 260px; height: 30px; margin-top: 10px; }</pre>
<div>6 性能优化百宝箱（下）</div>	<pre>@keyframes framework-stripes { from { background-position: 0 0 ; } to { background-position: 20rem 0; } }</pre>
<div>第3章 网络部分</div>	
<div>7 聊聊 DNS Prefetch</div>	
<div>8 Webpack 性能优化两三事</div>	
<div>9 图片加载优化（上）</div>	
<div>10 图片加载优化（下）</div>	
<div>第4章 缓存部分</div>	
<div>11 十八般缓存</div>	
<div>12 CDN 缓存</div>	
<div>13 本地缓存（Web Storage）</div>	
<div>14 浏览器缓存（上）</div>	
<div>15 浏览器缓存（下）</div>	
<div>第5章 渲染部分</div>	
<div>16 渲染原理与性能优化</div>	
<div>17 如何应对首屏 “一片空白” 最近阅读</div>	
<div>18 如何应对首屏 “一片空白” （下）</div>	
<div>19 不容小觑的 DOM 性能优化</div>	

那么具体的动态效果如下：



如上，我们通过修改 CSS，可以自定义骨架屏的动画效果，所以这种方式明显比绘制静态骨架屏图片更加灵活。因此在实际开发当中，我们多选择这种方法，具体的替换时机与上面讲的绘制静态骨架屏方案相同。

自动化解决方案

上面介绍的都是给一个页面添加骨架屏，如果我们需要多个页面都使用骨架屏，显然上面的两种方法就无法很好地满足我们的需要。这个时候我们更需要自动根据我们的页面生成对应的骨架屏，社区中已经有很多比较成熟的解决方案，目前使用比较广泛的是 [page-skeleton-webpack-plugin](#) 这个插件，它是一个 Webpack 插件，可以根据具体的页面生成对应的骨架屏。具体的[配置方案](#)这里写得非常详细，这里不再详细罗列，而且它支持动态调试，对于多个页面的使用非常方便。

<div>← 慕课专栏</div>	<div>≡ 你不知道的前端性能优化技巧 / 17 如何应对首屏 “一片空白” （上）</div>
<div>目录</div>	
<div>第1章 优化的概念</div>	
<div>1 开篇词：你的前端性能还能再抢救一下</div>	
<div>2 解读雅虎35条军规（上）</div>	
<div>3 解读雅虎35条军规（下）</div>	
<div>4 你要不要看这些优化指标？</div>	<div>使用成熟框架</div> <div>最后一种方法是使用一些成熟的框架，这些框架当中已经自带有骨架屏的组件，我们只要引入即可。这里我们以Ant-design 为例看下使用方法，如下：</div>
<div>第2章 性能工具介绍</div>	
<div>5 性能优化百宝箱（上）</div>	<div><pre>import React from 'react'; import ReactDOM from 'react-dom'; import 'antd/dist/antd.css'; import './index.css'; import { Skeleton, Button } from 'antd'; class Demo extends React.Component { state = { loading: false, }; showSkeleton = () => { this.setState({ loading: true }); setTimeout(() => { this.setState({ loading: false }); }, 3000); }; render() { return (<div className="article"> <Skeleton loading={this.state.loading}> <div> <h4>Ant Design, a design language</h4> <p> We supply a series of design principles, practical patterns and high resources (Sketch and Axure), to help people create their product beautifully and efficiently. </p> </div> </Skeleton> <Button onClick={this.showSkeleton} disabled={this.state.loading}> Show Skeleton </Button> </div>); } }</pre></div>
<div>6 性能优化百宝箱（下）</div>	
<div>第3章 网络部分</div>	
<div>7 聊聊 DNS Prefetch</div>	
<div>8 Webpack 性能优化两三事</div>	
<div>9 图片加载优化（上）</div>	
<div>10 图片加载优化（下）</div>	
<div>第4章 缓存部分</div>	
<div>11 十八般缓存</div>	
<div>12 CDN 缓存</div>	
<div>13 本地缓存（Web Storage）</div>	
<div>14 浏览器缓存（上）</div>	
<div>15 浏览器缓存（下）</div>	
<div>第5章 渲染部分</div>	
<div>16 渲染原理与性能优化</div>	
<div>17 如何应对首屏 “一片空白” 最近阅读</div>	<div><pre>ReactDOM.render(<Demo />, document.getElementById('container'));</pre></div>
<div>18 如何应对首屏 “一片空白”（下）</div>	<div>如上，我们在使用的时候只需要把我们的页面内容用 Skeleton 组件包起来即可，具体效果如下：</div>
<div>19 不容小觑的 DOM 性能优化</div>	

<div><div>← 慕课专栏</div><div>≡ 你不知道的前端性能优化技巧 / 17 如何应对首屏 “一片空白” （上）</div></div>	
目录	
第1章 优化的概念	
1 开篇词：你的前端性能还能再抢救一下	
2 解读雅虎35条军规（上）	
3 解读雅虎35条军规（下）	
4 你要不要看这些优化指标？	
第2章 性能工具介绍	
5 性能优化百宝箱（上）	
6 性能优化百宝箱（下）	
第3章 网络部分	
7 聊聊 DNS Prefetch	
8 Webpack 性能优化两三事	
9 图片加载优化（上）	
10 图片加载优化（下）	
第4章 缓存部分	
11 十八般缓存	
12 CDN 缓存	
13 本地缓存（Web Storage）	
14 浏览器缓存（上）	
15 浏览器缓存（下）	
第5章 渲染部分	
16 渲染原理与性能优化	
17 如何应对首屏 “一片空白” <div>最近阅读</div>	
18 如何应对首屏 “一片空白” （下）	
19 不容小觑的 DOM 性能优化	