requirejs

RequireJS 是一个JavaScript模块加载器。它非常适合在浏览器中使用，但它也可以用在其他脚本环境，就像 Rhino and Node。使用RequireJS加载模块化脚本将提高代码的加载速度和质量。

浏览器的兼容情况如下:

IE 6+ .......... 兼容 ✔ Firefox 2+ ..... 兼容 ✔ Safari 3.2+ .... 兼容 ✔

Chrome 3+ ...... 兼容 ✔ Opera 10+ ...... 兼容 ✔

最新版本: 2.1.11

**中文名**

requireJS

**外文名**

requireJS

目录

1 [获取 REQUIREJS 1](https://baike.baidu.com/item/requirejs/9527151?fr=aladdin#1)

2 [添加 REQUIREJS](https://baike.baidu.com/item/requirejs/9527151?fr=aladdin#2)

3 [优化 3](https://baike.baidu.com/item/requirejs/9527151?fr=aladdin#3)

开始使用 REQUIREJS

* 获取 RequireJS
* 添加 RequireJS
* [优化](https://baike.baidu.com/item/%E4%BC%98%E5%8C%96)

注意: 如果你使用 jQuery, 这里是针对 jQuery 的教程

获取 REQUIREJS 1

去[1]  下载页面下载文件。

添加 REQUIREJS

注意: 关于 jQuery 集成的意见, 请看jQuery 集成页面

假定你的项目中 JavaScript 都放在一个 "scripts" 目录。 例如, 你的项目中有一个 project.html 页面和一些 scripts, 目录布局如下：

* 项目目录/
  + util.js
  + main.js
  + helper/
  + project.html
  + scripts/

添加 require.js 到 scripts 目录, 如下:

* 项目目录/
  + util.js
  + main.js
  + require.js
  + helper/
  + project.html
  + scripts/

为了充分利用优化工具，建议您将所有的scripts放到HTML外面, 然后只引用 require.js 来请求加载你其它的scripts:

<!DOCTYPE html>  
<html>  
<head><title>My Sample Project</title>  
 <!-- data-main attribute tells require.js to load scripts/main.js after require.js loads. -->  
 <script data-main="scripts/main" src="scripts/require.js"></script>  
</head>  
<body><h1>My Sample Project</h1></body>  
</html>

在 main.js, 你可以使用 require() 来加载所有你需要运行的scripts. 这可以确保你所有的scripts都是在这里加载的, 你可以指定 data-main script 使用异步加载.

require(["helper/util"], **function** (util) {  
 //This function is called when scripts/helper/util.js is loaded.  
 // If util.js calls define(), then this function is not fired until  
 // util's dependencies have loaded, and the util argument will hold  
 // the module value for "helper/util".  
});

加载 helper/util.js 脚本. 想要充分利用 RequireJS, 请看API 文档去了解更多相关定义和模块的使用

优化 3

如果你最终决定在你的代码中使用, 可以使用[优化](https://baike.baidu.com/item/%E4%BC%98%E5%8C%96)结合 JavaScript 文件来减少加载时间。 在上面的例子中, 你可以结合 main.js 和 helper/util.js 加到一个文件中。