## 一、RequireJS要解决的问题：

1. 实现js文件的异步加载，避免网页失去响应；

使用RequireJS解决了我们的页面同步加载的问题，避免了页面长时间阻塞，甚至无响应。

1. 管理模块之间的依赖性，便于代码的编写和维护。

再也不用一个页面有很多个<script src=”xxx.js”></script>,而且不用担心分不清他们之间的依赖关系。有了require.js我们只需要在<head>标签里加一句script就行了。极大的便于开发人员编写和维护代码。

## 二、如何使用

#### （一）、加载require.js

1、使用require.js的第一步，是先去官方网站[下载](http://requirejs.org/docs/download.html)最新版本。

假定把require.js放在根目录下的js目录里。我们可以将如下代码放在<head>标签里：

<script src="js/require.js" **defer async="true"** ></script> //（01）

2、接下来加载自己的代码：

假定自己的代码文件叫main.js，也在js目录里。我们只需要将代码 01 作如下修改即可：

<script src="js/require.js" **defer async="true" data-main="js/main"** ></script> //（02）

**main.js就是我们的主模块。请将所有的子模块都放在main.js里面。**

#### （二）、主模块的写法：

提出依赖问题，比如我们自己写的main.js要使用了jquery插件，说明我们的main.js要依赖jquery,那么我们就向如下添加依赖关系。假如我们将jquery库放在与main.js同目录下的lib文件夹里面。

1、首先我们**配置主模块**：

//main.js

require.config({

//baseUrl: "js/lib", //(03)

paths: {

"jquery": "lib/jquery-1.9.1.min" //(04) 黄色部分是路径。

}

});

（03）:不建议使用基路径(baseUrl)，因为我们不可能把所的的js文件都放在一个文件夹里面，我们会分文件管理js文件。

2、配置完成后我们开始**编写主模块**：

require([“jquery”] , function($) { //(05)

//我们在这里编写主模块代码，这里面的代码依赖于jquery.如：

$(function() {

$(window).on(“click”,function() {

console.log(“jquery test success!”);

});

});

});

(05)处的jquery 和(04)处的jquery是关联的。

当然也可以依赖多个多个模块：

假定主模块依赖jquery、underscore和backbone这三个模块，main.js就可以这样写：

　　require(['jquery', 'underscore', 'backbone'], function ($, \_, Backbone){

　　　　// some code here

　　});

require.js会先加载jQuery、underscore和backbone，然后再运行回调函数。主模块的代码就写在回调函数中。

#### （三）、自定义AMD模块的写法

require.js加载的模块，采用AMD规范。也就是说，模块必须按照AMD的规定来写。

具体来说，就是模块必须采用特定的define()函数来定义。如果一个模块不依赖其他模块，那么可以直接定义在define()函数之中。

假定现在有一个math.js文件，它定义了一个math模块。那么，math.js就要这样写：

　　// math.js **如果有依赖关系，记得先配置。**

　　define(function (){

　　　　var add = function (x,y){

　　　　　　return x+y;

　　　　};

　　　　return {

　　　　　　add: add

　　　　};

　　});

加载方法如下：

　　// main.js

　　require(['math'], function (math){

　　　　alert(math.add(1,1));

　　});

如果这个模块还依赖其他模块，那么define()函数的第一个参数，必须是一个数组，指明该模块的依赖性。

　　define(['myLib'], function(myLib){

　　　　function foo(){

　　　　　　myLib.doSomething();

　　　　}

　　　　return {

　　　　　　foo : foo

　　　　};

　　});

当require()函数加载上面这个模块的时候，就会先加载myLib.js文件。

#### （四）其它使用方法详见：

<http://www.ruanyifeng.com/blog/2012/11/require_js.html>

#### （五）可参考demo使用