RequireJS:

主模块: main.js

require([“moduleA”,”moduleB”,”moduleC”],function(moduleA,module,moduleC){

//some code here

}

require()函数接受两个参数。

第一个参数是一个数组，表示所依赖的模块，上例就是['moduleA', 'moduleB', 'moduleC']，即主模块依赖这三个模块；

第二个参数是一个回调函数，当前面指定的模块都加载成功后，它将被调用。加载的模块会以参数形式传入该函数，从而在回调函数内部就可以使用这些模块。

require()异步加载moduleA，moduleB和moduleC，浏览器不会失去响应；它指定的回调函数，只有前面的模块都加载成功后，才会运行，解决了依赖性的问题。

栗子:

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

　　　　// some code here

});

模块加载：

设置基本目录:BaseUrl

require.config({

　　　　baseUrl: "js/lib",

　　　　paths: {

　　　　　　"jquery": "jquery.min",  
　　　　　　"underscore": "underscore.min",  
　　　　　　"backbone": "backbone.min"

　　　　}

　　});

AMD模块：

1.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。

栗子：模块math.js

define(function(){

var add = function(x,y){

return x + y;

};

return {

add: add

};

})

加载：require([“math”],function(){

alert(math.add(1,1));

})

2.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。

栗子：依赖jquery.js

define([“jquery”],function(mylib){

function foo(){

mylib.doSomething();

}

return {

foo: foo

};

})

加载非规范的模块:

方法：在用require()加载之前，要先用require.config()方法，定义它们的一些特征。

栗子：underscore和backbone这两个库，都没有采用AMD规范编写。如果要加载它们的话，必须先定义它们的特征。

require.config({

shim: {

“underscore”:{

exports: “\_”

},

“backbone”:{

deps: [“underscore”,”jquery”],

exports: “Backbone”

}

}

})

require.config()接受一个配置对象，这个对象除了有前面说过的paths属性之外，还有一个shim属性，专门用来配置不兼容的模块。

具体来说，每个模块要定义:

（1）exports值（输出的变量名），表明这个模块外部调用时的名称；

（2）deps数组，表明该模块的依赖性。

require.js插件:

domready插件，可以让回调函数在页面DOM结构加载完成后再运行。

require(['domready!'], function (doc){

　　　　// called once the DOM is ready

　　});

text和image插件，则是允许require.js加载文本和图片文件。

define([

　　　　'text!review.txt',

　　　　'image!cat.jpg'

　　　　],  
  
　　　　function(review,cat){

　　　　　　console.log(review);

　　　　　　document.body.appendChild(cat);

　　　　}

　　);