

requireJS

传统代码的缺陷

传统的用script标签导入的JS文件，有可能会阻塞页面渲染，而且正常我们开发的时候会导入引入JS文件，都用script标签导入，很容易造成混乱，不容易维护。

什么是模块化？

随着网页越来越复杂，网页中引入的JS文件也就越来越多，单纯的用<script>引入js文件，已经满足不了我们的需求。我们需要团队协作，模块分离。

额外扩展：

在Java中有一个重要带概念——package，逻辑上相关的代码组织到同一个包内，包内是一个相对独立的王国，不用担心命名冲突什么的，那么外部如果使用呢？直接import对应的package即可 import java.util.ArrayList; 遗憾的是JavaScript在设计时定位原因，没有提供类似的功能， 开发者需要模拟出类似的功能，来隔离、组织复杂的JavaScript代码，我们称为模块化。

模块化是指在处理某些问题时，按照一种分类或者思想对功能进行模块化管理和使用。

设计一个模块化需要具备的能力

- 1、具有定义封装的能力
- 2、具有定义依赖新模块的能力，可以引入其他模块的能力

function fn () {

}

function fn2 () {

}

```
var MyModule = new Object({
  count : 0,
  fn1 : function (num) {
    return count += num;
  },
  fn2 : function (num) {
    return count -= num;
  }
});
```

1、平时我们把功能写在普通函数中，坏处：容易造成函数命名冲突

2、为了解决命名冲突，可以用对象来封装函数 坏处：外部可以通过Mymodule.count来操作count/属性

模块化的演变

3、匿名函数闭包（这种方式已经是模块化编程的雏形了，能够实现功能的封装，也能够隐藏私有变量）

```
var MyModule = (function () {
  var count = 0;
  var fn1 = function (num) {
    return count += num;
  }
  var fn2 = function (num) {
    return count -= num;
  }
  return {
    fn1 : fn1,
    fn2 : fn2
  }
})();
```

模块化编程规范

AMD

AMD是requireJS在推广中对模块化编程规范的产出。

对于依赖的模块，AMD是提前执行，异步加载

AMD推崇依赖前置

```
define(['a','b'], function (a, b) {
  //依赖必须提前写好，才能使用
  a.doSomething();
  b.doSomething();
  ...
});
```

CMD

CMD是seaJS在推广中对模块化编程规范的产出。

CMD是延迟执行。

CMD推崇依赖就近

```
define(function (require, exports, moduls) {
  //依赖就近书写
  var a = require("a");
  a.doSomething();
  var b = require("b");
  b.doSomething();
});
```

这些规范目的都是为了更合理的去实现模块化编程，特别是在浏览器中，大部分都支持模块化编程

requireJS的基本API

requireJS有三个变量：
requirejs,require,define

其中requirejs 和 require 是一个意思，我们一般使用require就可以，require是加载模块用的，而且加载模块后，还可以执行回调函数

define：是定义模块（一个模块就是功能一个js文件）

使用define定义模块

通过require方法来加载模块

require(['js/sayHello'])

注意：syaHello相当于sayHello.js .js可写可不写

还可以设置回调函数

```
require(['js/sayHello'], function () {
  alert("加载完毕！ ");
});
```

如果该模块有导出内容的话，需要在回调函数中写上形参

```
require(['js/sayHello'], function (say) {
  alert("加载完毕！ " + say);
});
```

如何使用？

通过require.config 来配置路径，通过require.config 不仅能加载本地文件，还可以加载网络文件

```
require.config({
  paths : {
    jquery : "http://libs.baidu.com/jquery/2.0.0/jquery",
    say : './js/say'
  }
});

require(['jquery', 'say'], function ($) {
  $(function () {
    alert("加载完毕！ ");
  });
});
```