# JS/异步

1. 回调, 适应于异步处理较少的情况

javascript get(url, callback);

1. Promise, 链式方式写异步处理

javascript promise(somthing).then(res => handle)

1. async 和 await, 同步的方式写异步, 可适用于异步处理之间存在依赖的情况.

javascript const promiseFunc = async () => { const result1 = await getSomething(); const result2 = await dependOnResult1(result1); return result2; }; promiseFunc().then(result => { console.log(result); })

## async 和 await

[async](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function), [await](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/await)

### 扩展

1. [理解 JavaScript 的 async/await, 边城, 2016/11/19](https://segmentfault.com/a/1190000007535316)

# 模块化与组件化

内嵌框架图(embedded)

模块化, 强调内聚, 包含完整的业务逻辑, 可以方便业务的复用. 组件化, 强调复用, 重点在于接口的暴露, 和构件的概念类似.

## 实现模块化

业务相关的特殊性都应包含在同一个模块内, 具体到前端, 这些特性包括与业务相关的接口、状态、路由等.

## 实现组件化

关键是如何暴露接口, 方便外部复用.

# Vue.js 权限管理

* [基于Vue实现后台系统权限控制](http://refined-x.com/2017/08/29/%E5%9F%BA%E4%BA%8EVue%E5%AE%9E%E7%8E%B0%E5%90%8E%E5%8F%B0%E7%B3%BB%E7%BB%9F%E6%9D%83%E9%99%90%E6%8E%A7%E5%88%B6/)
* [用addRoutes实现动态路由](http://refined-x.com/2017/09/01/%E7%94%A8addRoutes%E5%AE%9E%E7%8E%B0%E5%8A%A8%E6%80%81%E8%B7%AF%E7%94%B1/)
* [手摸手，带你用vue撸后台 系列二(登录权限篇)](https://juejin.im/post/591aa14f570c35006961acac)
* [Vue 后台管理控制用户权限的解决方案？](https://www.zhihu.com/question/58991978)
* [自定义指令](https://cn.vuejs.org/v2/guide/custom-directive.html)
* <https://codepen.io/diemah77/pen/GZGxPK>

# FUNDAMENTAL/基础

# 参考

* [Markdown Editor](https://segmentfault.com/a/1190000005858817)

# 本地缓存

* [详说 Cookie, LocalStorage 与 SessionStorage](http://jerryzou.com/posts/cookie-and-web-storage/)