# JS/异步

1. 回调, 适应于异步处理较少的情况

* get(url, callback);

1. Promise, 链式方式写异步处理

* 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.

* 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)

# FUNDAMENTAL/基础

# 本地缓存

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