# Promise

Promise对象用于表示一个异步操作的最终状态（完成或者失败），以及其返回的值。

语法： new Promise ((resolve,reject) => { /\* 异步操作（executor）\*/ })

通过resolve和reject改变promise的状态（fulfilled/rejected），executor在promise构造函数返回新对象前被调用。

Promise接收两个参数，resolve和reject,他们是两个函数，由JavaScript引擎提供。在异步操作成功时，resolve将异步操作的结果作为参数传递出去，并设置promise的状态为fullfill（resolved）。异步操作失败时，reject将异步操作报出的错误作为参数传递出去，并设置promise的状态为rejected。

# 客户端Web API

Api:基于编程语言构建的结构，使开发人员容易地创建复杂的功能。它们抽象了复杂的代码，并提供一些简单的接口规则直接使用。

客户端JavaScript api通常分为两类：

1. 浏览器API，如Geolocation API
2. 第三方API，如Twitter API

## 1常见的浏览器API

### 1.1操作DOM的API



图1.1速览DOM接口

#### window

#### navigator

浏览器存在于web上的状态和标识，可以获取如：用户地理信息、用户偏爱语言、多媒体流等信息。

#### document

载入窗口的实际页面，在浏览器中用DOM表示，在JavaScript用document表示，可以用这个对象来返回和操作文档中的HTML和CSS上的信息。例如获取DOM中一个元素的引用，修改其文本内容，并应用新的样式，创建新的元素并添加为当前元素的子元素，甚至删除一个元素。

获取元素：

1. var link = document.querySelector(‘a’)
2. Var element1 = document.getElementById(‘myId’)

创建并且放置新的节点：

var sect = document.querySelector(‘section’) // 引用元素

var para = document.createElement(‘p’) // 创建元素

para.textContent = ‘We hope you enjoyed the ride.’

sect.appendChild(para) //追加元素

### 1.2从服务器获取数据的API

#### 1.2.1XMLHttpRequest

#### 1.2.2Fetch API

### 1.3用于绘制和操作图像的API

#### 1.3.1canvas

用JavaScript来绘制图形。

#### 1.3.2webgl

Web图像库，通过引入一个与opengl se 2.0紧密结合的API，可以在HTML5 canvas元素中使用。

### 1.4音频和视频API

### 1.5设备API

### 1.6客户端存储API

#### 1.6.1web storage

两种机制：

1. sessionStorage：仅回话期间可用。
2. localStorage：关闭浏览器，重新打开后数据仍然存在。

#### 1.6.2indexeddb

存储大量的结构化数据。

### 1.7API工作方式

属性和方法的访问链接在一起

仅在操作完成时调用函数-确保一个操作已经完成，然后在另一个操作中尝试使用该操作返回的数据（“异步”操作）。假设这样写：不能确保position数据立即返回，因此这样写行不通。Position数据还不可用。



## 2常见第三方API