

# JavaScript进阶

---JS异步与网络数据交互



河北师范大学软件学院  
Software College of Hebei Normal University

# 内容提纲

---

- **JS异步相关概念**
- **JS异步的几种形式**
- **JS异步与数据交互**



# JS异步相关概念

## • 单线程与多线程（优缺点）

- 线程是程序执行流的最基本单元（类比单个实体的人），是进程中的一个实体
- 一个线程可与同属一个进程的其它线程共享进程所拥有的系统资源
- 标准的线程由线程ID，当前指令指针(PC)，寄存器集合和堆栈组成
- JS是单线程的，指的是JS引擎解释和执行代码是单线程的，完成所有任务还要配合其他线程

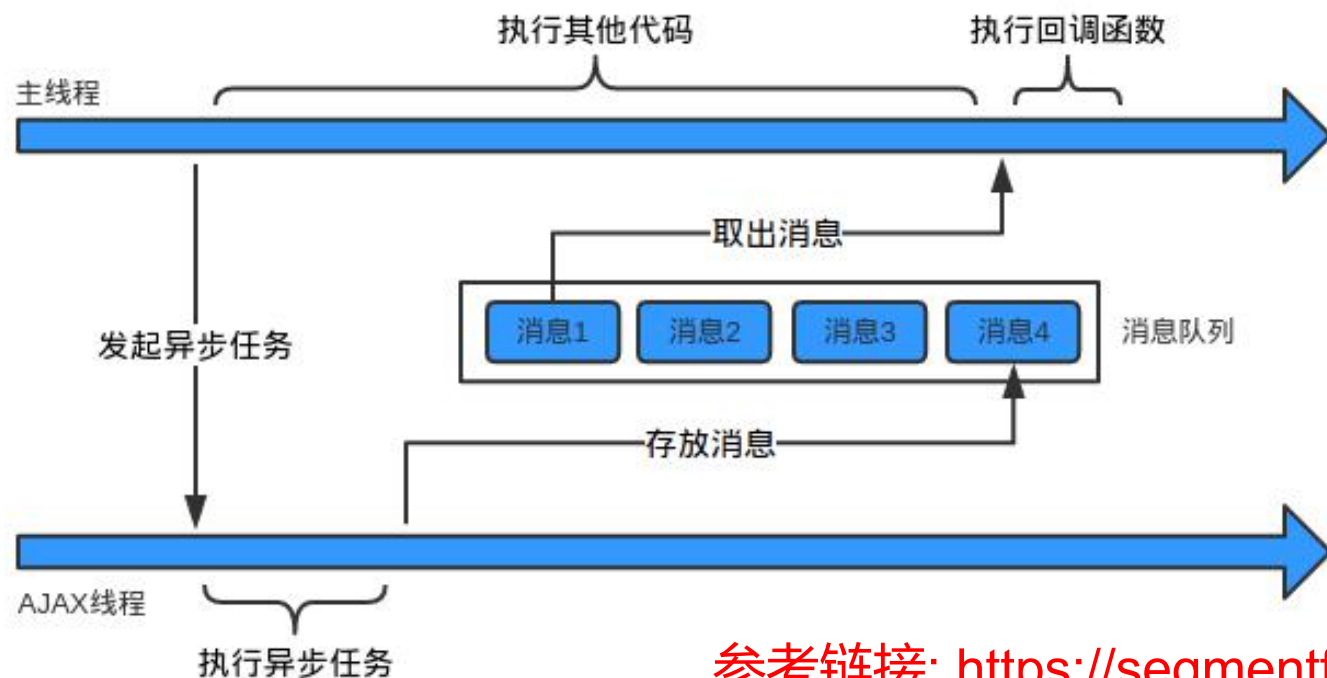
## • 阻塞与非阻塞（优缺点）

- 阻塞是指调用结果返回之前，当前线程会被挂起。调用线程只有在得到结果之后才会返回
- 非阻塞调用指在不能立刻得到结果之前，该调用不会阻塞当前线程

# JS异步相关概念

## • 同步与异步（JS中的任务主要分为两种）

- 同步任务：在主线程上排队执行的任务，只有前一个任务执行完毕，才能执行后一个任务
- 异步任务：不进入主线程、而进入"任务队列"（task queue）的任务，只有等主线程任务执行完毕，"任务队列"开始通知主线程，请求执行任务，该任务才会进入主线程执行



参考链接: <https://segmentfault.com/a/1190000004322358>

# 内容提纲

---

- JS异步相关概念
- JS异步的几种形式
- JS异步与数据交互案例



# JS异步的几种形式

---



## • 回调函数

- 优点：简单、容易理解和部署
- 缺点：不利于代码的阅读和维护，各部分之间高度耦合（Coupling），流程会很混乱，而且每个任务只能指定一个回调函数

## • 事件监听机制

- 任务的执行不取决于代码的顺序执行，而取决于某个事件是否发生
- 优点：容易理解，可以绑定多个事件，每个事件可以指定多个回调函数，可以在多个不同模块中传递事件和数据
- 缺点：整个程序都要变成事件驱动型，运行流程会变的不清晰

# JS异步的几种形式

## •发布订阅（观察者模式）

- 发布---订阅模式又叫观察者模式，它定义了对象间的一种一对多的关系，让多个观察者对象同时监听某一个主题对象，当一个对象发生改变时，所有依赖于它的对象都将得到通知
- 发布---订阅模式的性质与“事件监听”类似，但是我们可以通过常“消息中心”，了解存在多少信号、每个信号有多少订阅者，从而监控程序的运行

## •Promise

- 每一个异步任务返回一个Promise对象，可链式的指定回调函数
- 优点：回调函数编程里链式写法，程序的流程更接近人们的思维方式，便于理解和代码追踪
- 缺点：要求对异步有更深入的理解

# 内容提纲

---

- JS异步相关概念
- JS异步的几种形式
- JS异步与数据交互案例





# JS异步网络数据交互

---



## • Ajax

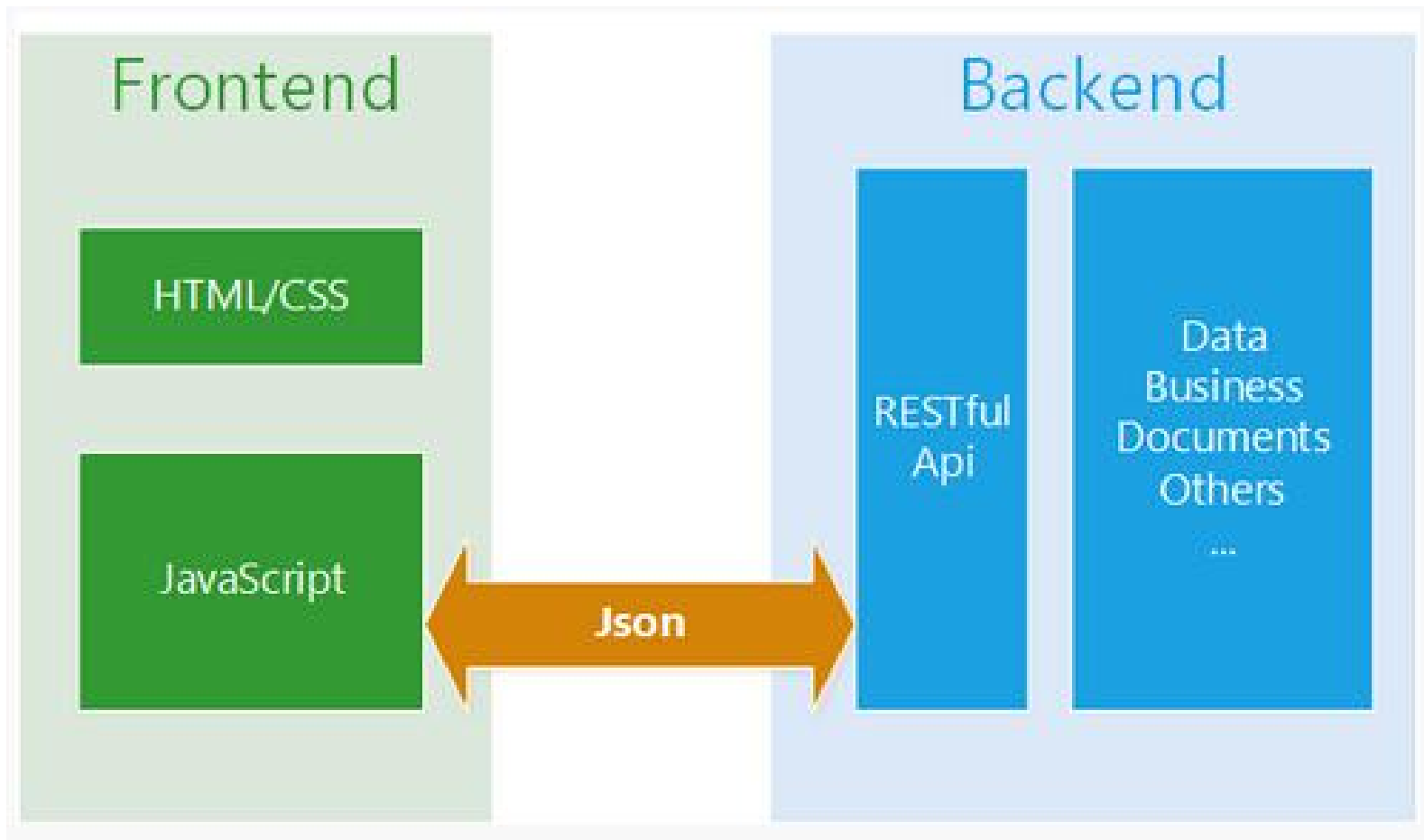
- AJAX = Asynchronous JavaScript and XML ( 异步 JavaScript 和 XML , 也可用JSON )
- AJAX 是一种使用现有标准和技术的综合实现
- AJAX 是与服务器交换数据并更新部分网页的艺术 , 在不重新加载整个页面的情况下
- 实现方式 ( XMLHttpRequest对象、jQuery中的ajax对象等 )

## • XMLHttpRequest

- 通过XMLHttpRequest实例化的对象可用于在后台与服务器交换数据 , 在不重新加载整个网页的情况下 , 对网页的某部分进行更新
- 参考链接 : <http://blog.csdn.net/liujiahan629629/article/details/17126727>
- 参考链接 : [http://blog.csdn.net/Huang\\_Cai\\_Yuan/article/details/54881407](http://blog.csdn.net/Huang_Cai_Yuan/article/details/54881407)

# JS异步网络数据交互

- 前后端分离与前后端交互



查看建站相关  
[www.aliyun.com](http://www.aliyun.com)

[cloud.tencent.com](http://cloud.tencent.com)

[cloud.tencent.com/solution/database](http://cloud.tencent.com/solution/database)

理解网络前后端交互等  
相关知识



# JS异步网络数据交互

## •XMLHttpRequest对象的属性

|                    |   |
|--------------------|---|
| readyState         | 表示XMLHttpRequest对象的状态：0：未初始化。对象已创建，未调用open；<br>1： open方法成功调用，但Send方法未调用；<br>2： send方法已经调用，尚未开始接受数据；<br>3： 正在接受数据。Http响应头信息已经接受，但尚未接收完成；<br>4： 完成，即响应数据接受完成。 |
| Onreadystatechange | 请求状态改变的事件触发器（readyState变化时会调用这个属性上注册的javascript函数）。   |
| responseText       | 服务器响应的文本内容  |
| responseXML        | 服务器响应的XML内容对应的DOM对象   |
| Status             | 服务器返回的http状态码。200表示“成功”，404表示“未找到”，500表示“服务器内部错误”等。   |
| statusText         | 服务器返回状态的文本信息。   |



# JS异步网络数据交互

## •XMLHttpRequest对象的方法 一

|   |  |
|---|--|
| Open(string method,string url,boolean asynch,String username,string password) | 指定和服务端交互的HTTP方法，URL地址，即其他请求信息；<br>Method:表示http请求方法，一般使用"GET","POST".<br>url: 表示请求的服务器的地址；<br>asynch: 表示是否采用异步方法，true为异步，false为同步；<br>后边两个可以不指定，username和password分别表示用户名和密码，提供http认证机制需要的用户名和密码。 |
| Send(content)   | 向服务器发出请求，如果采用异步方式，该方法会立即返回。<br>content可以指定为null表示不发送数据，其内容可以是DOM对象，输入流或字符串。  |
| SetRequestHeader(String header,String Value)                                  | 设置HTTP请求中的指定头部header的值为value.<br>此方法需在open方法以后调用，一般在post方式中使用。   |





# JS异步网络数据交互

## •XMLHttpRequest对象的方法 二

|   |  |
|---|--|
| <code>getAllResponseHeaders()</code>          | 返回包含Http的所有响应头信息，其中相应头包括Content-length,date,uri等内容。<br>返回值是一个字符串，包含所有头信息，其中每个键名和键值用冒号分开，每一组键之间用CR和LF（回车加换行符）来分隔！ |
| <code>getResponseHeader(String header)</code> | 返回HTTP响应头中指定的键名header对应的值  |
| <code>Abort()</code>                          | 停止当前http请求。对应的XMLHttpRequest对象会复位到未初始化的状态。   |

参见NodeAjaxTest文件夹中的 Test01和Test02 （包括html文件和js文件 掌握XMLHttpRequest的应用）

# JS异步网络数据交互

- 使用jQuery中的ajax对象

```
$('#btn').click(function () {  
    $.ajax({  
        url: 'http://localhost:8080?id='+($('#inp').val()||1),  
        dataType: 'JSON',  
        type: 'GET',  
        success: function (data) {  
            console.log(data);  
        },  
        error: function (xhr) {  
            console.log('error');  
        }  
    })  
})
```



Thank You !



河北师范大学软件学院  
Software College of Hebei Normal University

# 作业

---

- 复习本章内容
- 查阅jQuery中ajax请求的实现方法
- 查看慕课网ajax相关内容，链接如下：
- 查看慕课网node基础相关内容（了解前后端交互）

<https://www.imooc.com/learn/250>

<https://www.imooc.com/learn/348>