

# node.js first day

---

## 介绍

- node是什么 \* Node.js是基于Chrome的V8 JavaScript引擎构建的JavaScript运行时环境 \* node.js不是库也不是语言 \* 可以解析和执行javascript代码 \* node里的js可以脱离浏览器来运行 \* node.js里的js没有BOM、DOM，在执行环境中为js提供了一些服务器级别的操作API \* 文件读写 \* http操作 \* 网络服务构建 \* 网络通信
  - node.js能做什么？
    - web服务器后台
    - 游戏，接口等的服务器
    - 命令行工具
  - 需要掌握
    - B/S编程模型
    - node 常用api
    - 模块化编程
    - 异步编程
    - express 开发框架

## helloworld

---

- node操作
  - 编写js脚本文件
  - 打开终端 定位到指定目录
  - 通过node 文件名进行编译
- 文件操作
  - 读取文件

```
var fs=require('fs')
```

- fs有两个参数，data 和 error
    - 获取文件成功 data就是获取到底数据 error是null
    - 获取文件失败 data null error是错误对象
- http模块
  - 功能：构建web服务器，用来创建编写服务器
  - 使用
    - 加载http模块

```
var http=require('http')
```

- 使用http.createServer()创建服务器

```
var server=http.createServer()
```

- 监听request请求事件

```
Var server server.on('request',function(){  
    Console.log('收到')  
})
```

- 绑定端口号, 启动服务器

```
Server.listen(3000,function(){  
    Console.log('服务器启动成功')  
})  
//浏览器访问: http://127.0.0.1:3000/
```

- 在监听request请求时会有两个参数

- Request 请求对象 请求对象可以用来获取客户端的请求信息
- Response 响应对象 响应对象可以用来给客户端发送响应消息

```
var http=require('http')  
var server= http.createServer()  
server.on('request',function(request,response){  
    if(request.url=='/login'){  
        console.log('接收到请求'+request.url)  
        response.end('index')  
        // response.write('index')  
        // response.end()  
    }else if(request.url=='/h'){  
        console.log('接收到请求'+request.url)  
        response.end('hahahahahah')  
        // response.write('hahahahahah')  
        // response.end()  
    }else{  
        console.log('接收到请求'+request.url)  
        response.end('node')  
        // response.write('node')  
        // response.end()  
    }  
})  
server.listen(3000,function(){  
    console.log('服务器启动成功')  
})
```

注意：只能响应二进制或字符串，其他类型需要使用JSON转换

```
// 方法一
response.write('node')
response.end()
//方法二
response.end('node')
```