nodejs.org

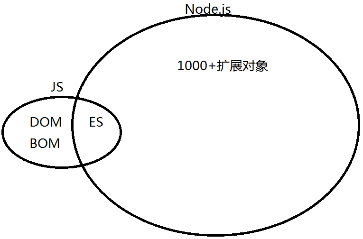
1.Node.js概述

**JavaScript**是一门运行于浏览器(客户端)中的语言，主要对象包括：

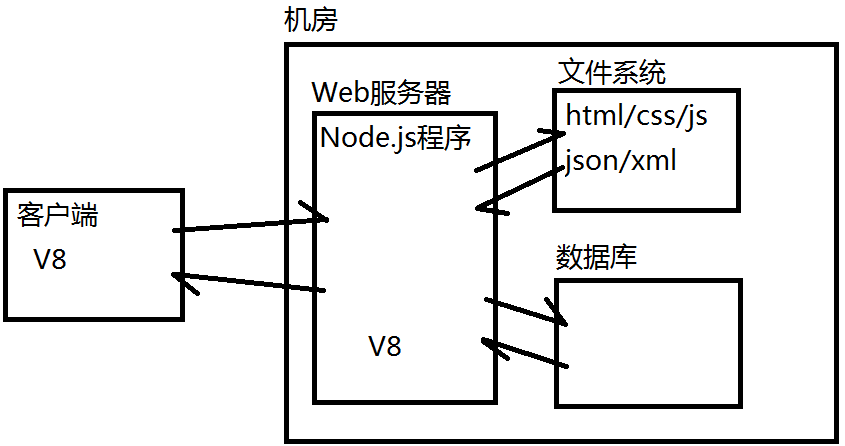
ES原生对象（11个）、BOM对象、DOM对象

**Node.js**是一个运行于服务器端的语言，类似于PHP、JSP、ASP.NET，主要对象包括：

ES原生对象、1000+多个扩展对象

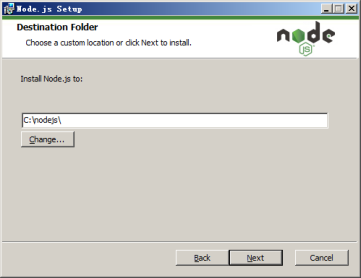
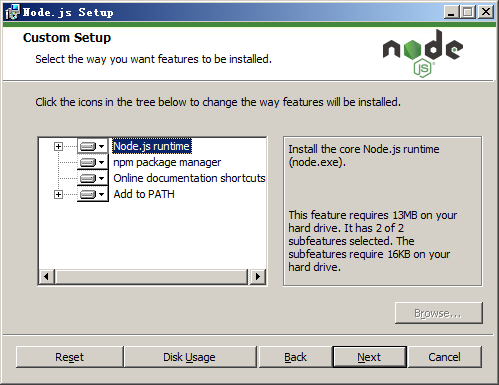


一种语言可以通吃前后端！走在最前端！



下载并按照Node.js解释器： https://nodejs.org

LTS: Long Term Support

2.Node.js的两种运行模式

**(1)交互模式：REPL模式，Read Evaluate Print Loop**

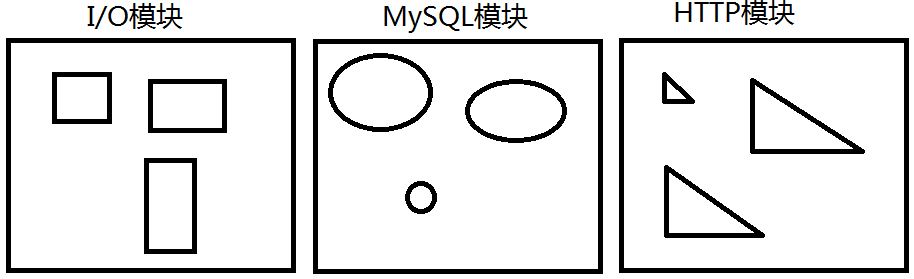
e:/mynode> node

一般用于测试，需要输入一行执行一行。

**(2)脚本模式：把要执行的所有代码置于一个文本文件中，可以.js后缀，也可以没有**

e:/mynode> node x.js

正式项目应用中都使用脚本模式。



Module：模块，用于将不同的功能对象划分到不同的空间下，避免彼此干扰或污染全局对象。

3.Node.js的常用模块——QueryString模块

HTTP 协议规定，请求消息可以在URL后面追加一个?开头的QueryString，用于向服务器提交请求数据。

Node.js提供了一个QueryString模块，用于解析键值对形式的查询字符串，得到一个对象。

使用方法：

var qs = require('querystring');

var obj = qs.parse('k1=v1&k2=v2');

4.Node.js的常用模块——URL模块

Node.js提供的URL模块用于解析一个完整的URL地址，得到其中的各个部分。

使用方法：

var url = require('url');

var obj = url.parse('http://tedu.cn:80/s.do?k1=v1&k2=v2');

5.Node.js的常用模块——FileSystem模块

FileSystem模块用于操作文件系统——读写目录、读写文件——Node.js底层使用C程序来实现，这些功能是客户端JS所不具备的。

使用方法：

var fs = require('fs');

var buf = fs.readFileSync('源文件名'); //读取文件内容到缓冲区

fs.writeFileSync('目标文件名', '要写出的内容'); //向文件中写出内容

fs.appendFileSync('目标文件名', '要追加的内容'); //向文件中追加内容

练习：使用Node.js把已经存在的1.html拷贝为11.html

提示：先读取指定文件内容，再把这些内容写出到另一个文件

6.Node.js的常用模块——HTTP模块

HTTP协议的内容：

**(1)请求消息**

GET /web/index\_new.html?tedu HTTP/1.1

Host: tmooc.cn

Connection: keep-alive

**(2)响应消息**

HTTP/1.1 200 OK

Server: Apache-Coyote/1.1

Content-Type: text/html;charset=UTF-8

Node.js的HTTP模块可以用于接收客户端的请求消息，读取其中的请求数据，向客户端返回响应消息——即实现Web服务器的功能。

7.Node.js的常用第三方模块——MySQL模块

Node.js官方提供了二三十个模块，但仍然不够用。如缺少访问MySQL数据库的模块，到第三方网站下载，如全世界最大的node.js第三方模块集散地： https://www.npmjs.com/

提示：下载到的第三方模块要想使用，必须保存在nodejs的同级目录下的名为node\_modules的目录下。

第三方模块的目录下有一个README的文件，就是该模块的使用说明文档。

使用步骤：

var mysql = require('mysql');

var conn = mysql.createConnection({

host : '127.0.0.1',

user : 'root',

password : '',

database : 'tedu'

})

conn.query('INSERT/SELECT...', function(err, result){

})

conn.end();

8. 使用Node.js创建一个静态Web服务器

1)创建一个HTTP Server

2)为Server指定处理请求消息的过程

2.1)解析请求URL中的资源名称， 如 /login.html

2.2)读取指定文件中的内容，如 htdocs/login.html

2.3)构建响应消息，把读取到的文件内容输出客户端

3)让Server开始监听特定端口

提示：上述程序需要用到http、url、fs模块

9. 使用Node.js创建一个动态Web服务器——难点

动态Web服务器：可以接收客户端提交的内容，访问数据库，动态的生成响应内容。

提示：上述程序需要用到http、url、fs、mysql模块

运行过程：

1)创建一个HTTP Server

2)为Server指定处理请求消息的过程

2.1)解析请求URL中的资源名称， 如 /login.html或/login.do

2.2)若请求资源名称以.html结尾，直接读取指定文件中的内容，如 htdocs/login.html，作为响应消息数据； **若请求资源名称以.do结尾，解析请求数据，访问数据库，把执行结果作为响应消息数据**。

2.3)构建响应消息，把读取到的文件内容输出客户端客户端

3)让Server开始监听特定端口