# 实验1：使用node.js

实验背景

利用已掌握的知识，成功安装node.js交互环境，并完成入门实例。如图所示：

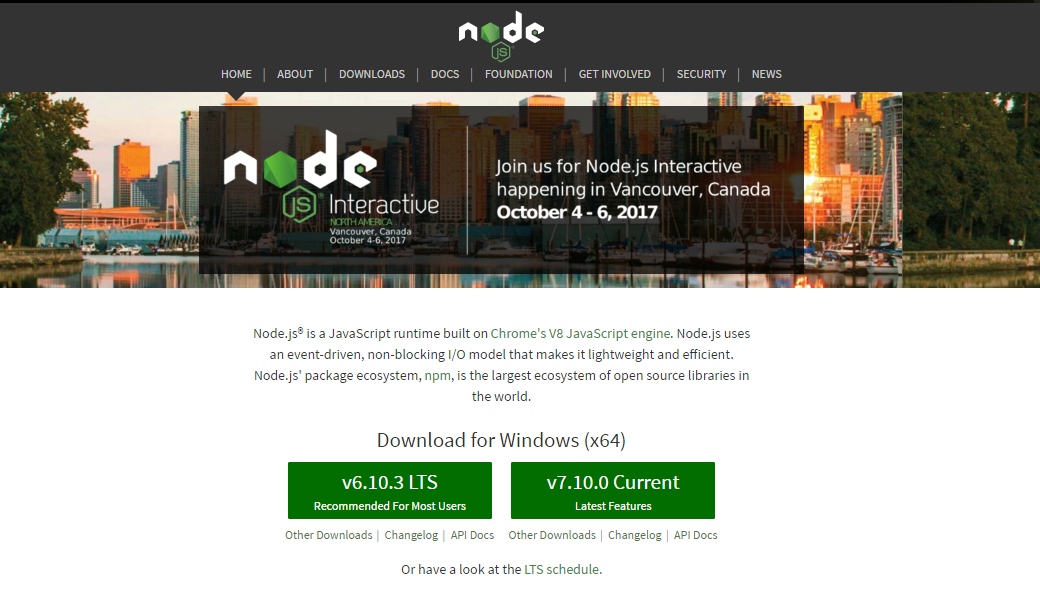


图1-1 创建简单的http请求

实验目的

1．了解node.js的特点

2．掌握node.js的下载

3．掌握repl终端的使用

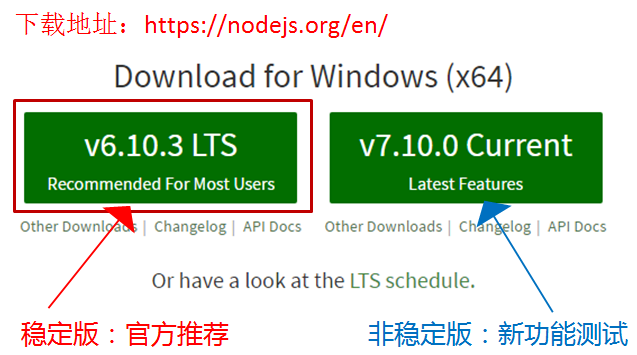
4．掌握入门小实例

实验分析

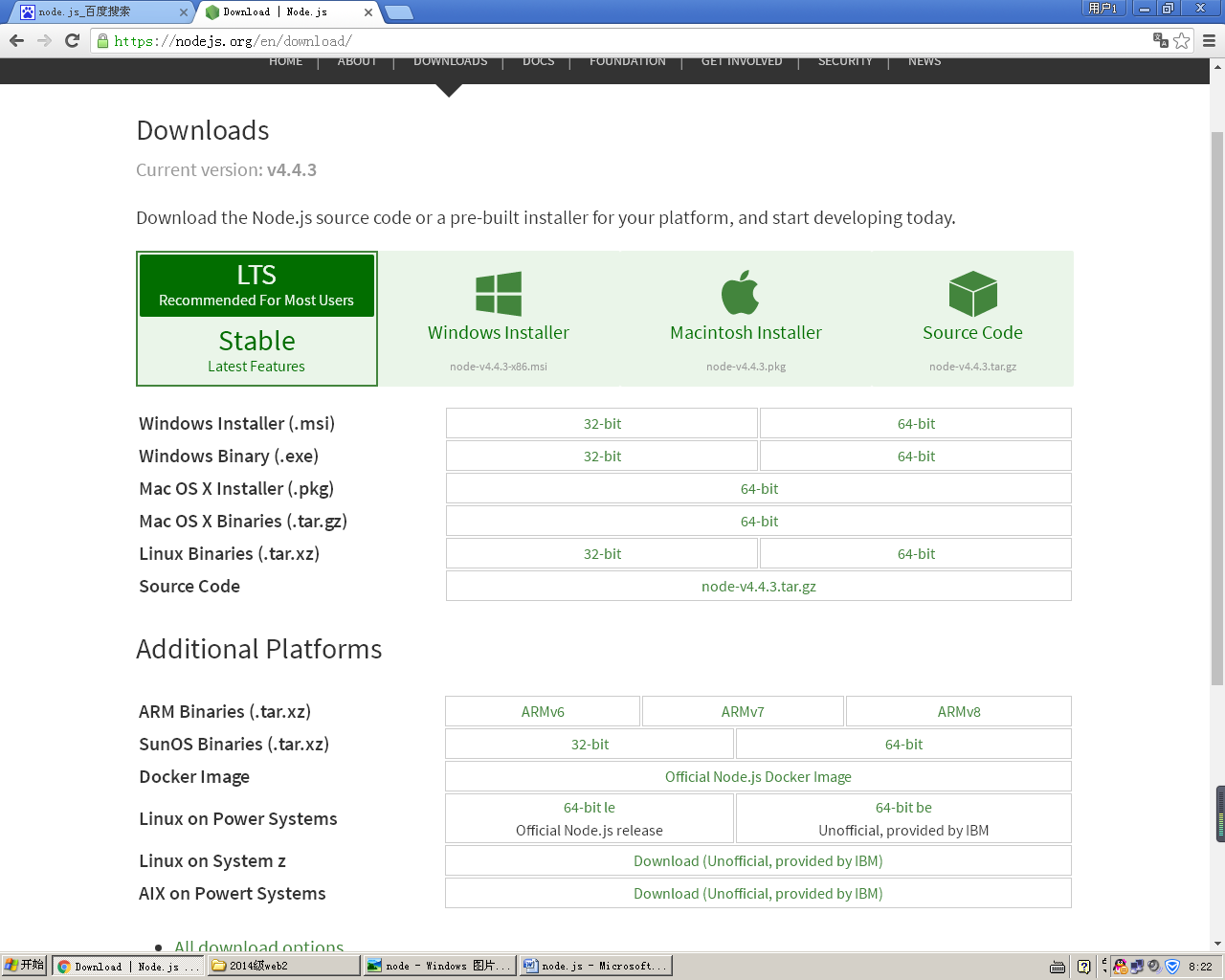
本次实验要完成的有三个方面的知识。第一：正确下载和安装node.js？第二：使用repl，第三：完成入门小实例。

实验步骤

1. 下载node.js



选择合适的版本：



Node.js 历史版本下载地址：https://nodejs.org/dist/

注意：Linux上安装Node.js需要安装Python 2.6 或 2.7 ，不建议安装Python 3.0以上版本。

目前 Node.js 推出了长期支持版本和新功能稳定版，你可以根据需求下载，v4.0.0 是 Node.js 和 io.js 合并之后的首个稳定版本，支持 ES 6的大部分特性。

**➀Windows 上安装Node.js**

Windows 安装包(.msi) :

**安装步骤：**

1 : 双击下载后的安装包 node-v6.10.3-x64.msi，如下所示：

2 : 点击以上的Run(运行)，将出现如下界面：

3 : 勾选接受协议选项，点击 next（下一步） 按钮 :

4 : Node.js默认安装目录为 "C:\Program Files\nodejs\" , 你**可以修改目录**，并点击 next（下一步）：

5 : 点击树形图标来选择你需要的安装模式 , 然后点击下一步 next（下一步）

默认安装：

* Node.js运行环境，必装。
* npm包管理器，用来安装第三方的Package，Express。
* Online document shortcuts就是在安装中，会在开始菜单的Node.js项下产生一个在线文档的链接。
* Add to PATH就是在安装完后，将Node.js的安装目录添加到环境变量Path中，那么，在命令行中的任何路径下运行node命令。

6 :点击 Install（安装） 开始安装Node.js。你也可以点击 Back（返回）来修改先前的配置。 然后并点击 next（下一步）：

7:点击 Finish（完成）按钮退出安装向导。

**检测PATH环境变量是否配置了Node.js**

点击开始=》运行=》输入"cmd" => 输入命令"path"，输出如下结果：

PATH=C:\oraclexe\app\oracle\product\10.2.0\server\bin;C:\Windows\system32;

C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;

c:\python32\python;C:\MinGW\bin;C:\Program Files\GTK2-Runtime\lib;

C:\Program Files\MySQL\MySQL Server 5.5\bin;C:\Program Files\nodejs\;

C:\Users\rg\AppData\Roaming\npm

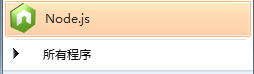
我们可以看到环境变量中已经包含了C:\Program Files\nodejs\

检查Node.js版本

node --version

**启动node.js**

在“开始”-》“程序”找到

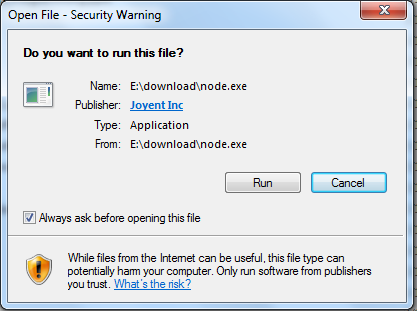


直接双击node.js

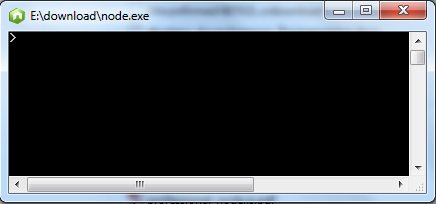
**➁、Windows 二进制文件 (.exe)安装 :**

安装步骤

1 : 双击下载的安装包 Node.exe



点击 Run（运行）按钮将出现命令行窗口



版本测试：进入 node.exe 所在的目录，

node-version

如果你获得以上输出结果，说明你已经成功安装了Node.js。

1. 使用REPL终端

简单表达式

在Node.js的REPL命令提示符下完成简单的数学计算：

$ node

> 1 + 3

4

> 1 + ( 2 \* 3 ) - 4

3

>

使用变量

可以使用变量之后存储值和打印，就像传统的脚本。 如果不使用var关键字接着值存储在变量和打印。而如果是使用var关键字则值存储不打印。可以使用console.log打印变量()。

$ node

> x = 10

10

> var y = 10

undefined

> x + y

20

> console.log("Hello World")

Hello Workd

undefined

多行表达

Node REPL支持类似于JavaScript的多行表达。让我们来看看以下do-while循环的操作：

$ node

> var x = 0

undefined

> do {

... x++;

... console.log("x: " + x);

... } while ( x < 5 );

x: 1

x: 2

x: 3

x: 4

x: 5

undefined

>

... 自动打开自带括号后，当按下进入。Node自动检查表达式的连续性。

下划线变量

您可以使用下划线\_得到最后的结果：

$ node

> var x = 10

undefined

> var y = 20

undefined

> x + y

30

> var sum = \_

undefined

> console.log(sum)

30

undefined

>

1. 入门小实例

**➀、引入required模块**

我们使用 **require 指令**来载入http模块，并将实例化的HTTP赋值给变量http，实例如下:

var http = require("http");

**➁、创建服务器**

接下来我们使用 **http.createServer()方法**创建服务器，并使用 **listen方法绑定 8888 端口**。 **函数通过 request, response 参数来接收和响应数据**。

实例如下，在你项目的根目录下创建一个叫 server.js 的文件，并写入以下代码：

var http = require('http');

http.createServer(function (request, response) {

// 发送 HTTP 头部

// HTTP 状态值: 200 : OK

// 内容类型: text/plain

response.writeHead(200, {'Content-Type': 'text/plain'});

// 发送响应数据 "Hello World"

response.end('Hello World\n');

}).listen(8888);

// 终端打印如下信息

console.log('Server running at http://127.0.0.1:8888/');

以上代码我们完成了一个可以工作的HTTP服务器。

**➂使用node命令执行**

node server.js

Server running at http://127.0.0.1:8888/

接下来，打开浏览器访问 http://127.0.0.1:8888/，你会看到一个写着 "Hello World"的网页。

扩展练习

1.思考并实现，在linux上安装node.js

参考学习网址：http://www.runoob.com/nodejs/nodejs-tutorial.html

实验思考

1.思考并实现，使用repl输出\*号组成的正方形。