**安装node.js**

去官网https://nodejs.org/en下载安装包.mis或.exe,一直下一步安装完成，将安装路径配置到环境变量path,用cmd或git base输入命令：node –v 和 npm –v看是否成功。

**node.js创建简单服务器**

编写一个server.js文件

文件内容：

var http = require('http');

http.createServer(function (req, res){

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

res.end('Hello node.js\n');

}).listen(1337, '127.0.0.1');

console.log('Server running at <http://127.0.0.1:1337/>');

创建完成后，用cmd或git base进入该文件所在目录，用命令node server.js启动服务器，然后在浏览器中输入http://127.0.0.1:1337就可以看到输出的内容。如果修改了server.js文件的内容，必须在命令行中输入Ctrl+c停止服务，然后重新启动才能看到改变后的内容。

**URL网址解析**

在命令行中敲node后，再敲url.parse('网址')

如：url.parse('http://www.imooc.com/search/course?words=node.js')

就会解析出：

Url {

protocol: 'http:',

slashes: true,

auth: **null**,

host: 'www.imooc.com',

port: **null**,

hostname: 'www.imooc.com',

hash: **null**,

search: '?words=node.js',

query: 'words=node.js',

pathname: '/search/course',

path: '/search/course?words=node.js',

href: 'http://www.imooc.com/search/course?words=node.js' }

如：url.parse('http://www.imooc.com/search/course?words=node.js', true)

解析出：

Url {

protocol: 'http:',

slashes: true,

auth: **null**,

host: 'www.imooc.com',

port: **null**,

hostname: 'www.imooc.com',

hash: **null**,

search: '?words=node.js',

query: { words: 'node.js' }, //发生变化的地方

pathname: '/search/course',

path: '/search/course?words=node.js',

href: 'http://www.imooc.com/search/course?words=node.js' }

如：url.parse('//www.imooc.com/search/course?words=node.js',true,true)

//不知道网址的协议，如果不加最后一个参数true,host和hsotname就会解析不出来

解析出：

Url {

protocol: **null**,

slashes: true,

auth: **null**,

host: 'www.imooc.com',

port: **null**,

hostname: 'www.imooc.com',

hash: **null**,

search: '?words=node.js',

query: { words: 'node.js' },

pathname: '/search/course',

path: '/search/course?words=node.js',

href: '//www.imooc.com/search/course?words=node.js' }

**参数处理（有大量参数传递时最适应）**

在输入node后输入以下命令：

将字符串转换为对象：

如：querystring.parse('username=ypy&password=123')

返回：{ username: 'ypy', password: '123' }

如：querystring.parse('username=ypy,password=123',',') //第二个参数指定分割符号

返回：{ username: 'ypy', password: '123' }

如：querystring.parse('username:ypy,password:123',',',':')

//第三个参数指定参数key和value之间的符号

返回：{ username: 'ypy', password: '123' }

将对象转换为字符串：

如：querystring.stringify({ username: 'ypy', password: '123' })

返回：'username=ypy&password=123'

如：querystring.stringify({ username: 'ypy', password: '123' },',')

返回：'username=ypy,password=123'

如：querystring.stringify({ username: 'ypy', password: '123' },',',':')

返回：'username:ypy,password:123'

转义：

如：querystring.escape('哈哈') //进行转义

返回：%E5%93%88%E5%93%88

如：querystring.unescape('%E5%93%88%E5%93%88') //反转义

返回：'哈哈'