创建第一个应用

1. 第一个Node.js程序：Hello World!

1.1 脚本模式:

```
首先我们的创建一个Node.js程序文件 ： 代码如下
console.log("Hello World");

运行实例 »
保存该文件，文件名为 helloworld.js， 并通过 node命令来执行：
$ node helloworld.js

程序执行后，正常的话，就会在终端输出：
Hello World。
```

1.2 交互模式:

```
打开终端，键入node进入命令交互模式，可以输入一条代码语句后立即执行并显示结果，例如：
$ node
> console.log('Hello World!');
Hello World!
```

2. 创建第一个应用

> 如果我们使用PHP来编写后端的代码时，需要Apache 或者 Nginx 的HTTP 服务器，并配上 mod_php5 模块和php-cgi。不过对 Node.js 来说，概念完全不一样了。使用 Node.js 时，我们不仅仅 在实现一个应用，同时还实现了整个 HTTP 服务器。事实上，我们的 Web 应用以及对应的 Web 服务器基本上是一样的。

```
Node.js 应用是由哪几部分组成的：
1. 引入 required 模块：我们可以使用 require 指令来载入 Node.js 模块。
2. 创建服务器：服务器可以监听客户端的请求，类似于 Apache 、Nginx 等 HTTP 服务器。
3. 接收请求与响应请求 服务器很容易创建，客户端可以使用浏览器或终端发送 HTTP 请求，服务器接收请求后返回响应数据。
```

1. 新建一个 server.js 文件

```
// 1. 加载 http “工具”
const http = require('http');

// 2. 使用 http.createServer() 方法创建一个 Web 服务器
//    返回一个 Server 实例
```

```
const server = http.createServer();

// 3. 注册 request 请求事件
//    当客户端请求过来，就会自动触发服务器的 request 请求事件，然后执行第二个参数：回调处理函数
server.on('request', function (req, res) {
  res.end('Hello Node.js!');
});

// 4. 绑定端口号，启动服务器
server.listen(8888, function () {
  console.log('服务器启动成功，请求访问 http://127.0.0.1:8888/');
});
```

2. 使用 node 命令执行这个脚本代码

   $ node server.js Server running at http://127.0.0.1:8888/

3. 打开浏览器访问： http://127.0.0.1:8888/，你会看到结果。

3. REPL命令(交互式解释器)

Node 自带了交互式解释器，可以执行以下任务：

- 读取 - 读取用户输入，解析输入了Javascript 数据结构并存储在内存中。
- 执行 - 执行输入的数据结构
- 打印 - 输出结果
- 循环 - 循环操作以上步骤直到用户两次按下 ctrl-c 按钮退出。

Node 的交互式解释器可以很好的调试 Javascript 代码。

```
ctrl + c - 退出当前终端。
ctrl + c 按下两次 - 退出 Node REPL。
ctrl + d - 退出 Node REPL.
向上/向下 键 - 查看输入的历史命令
tab 键 - 列出当前命令
.help - 列出使用命令
.break - 退出多行表达式
.clear - 退出多行表达式
.save filename - 保存当前的 Node REPL 会话到指定文件
.load filename - 载入当前 Node REPL 会话的文件内容。
```