

模块化引入：

## 1、服务器的基本搭建

http服务模块的引入：`const http=require('http');`

`var server = http.createServer(function(req,res){`

`res.write('9999999999999999');`

`res.end();`

`}).listen(8080);`

服务器的创建

响应给客户端的内容

响应结束

服务器监听服务端口

```
const http=require('http'); // http服务模块的引入
// 服务器的创建 有两个参数请求、响应
var server = http.createServer(function(req,res){
// 响应给客户端
res.write(str);
// 响应结束
res.end();
console.log('有人来了');
// 监听及监听端口
}).listen(8080);
```

## 2、文件操作模块：fs——File System

`readFile(文件名, function (err, data){});`读取文件

`writeFile(文件名, 内容, function (err){});`写文件

```

const fs=require('fs'); //引入文件操作模块
//readFile(文件名, 回调函数)
fs.readFile('./liu/aaa.txt', function (err, data){
  if(err){
    console.log('读取失败');
  }else{
    console.log(data.toString());
  }
});
// writeFile(文件名, 内容, 回调)
fs.writeFile("./liu/bbb.txt", "我是新写的文件看看写了啥东西", function (err){
  console.log(err);
});

```

3、通过引入文件操作模块完成用户的访问请求

```
const http = require('http'); //引入HTTP模块
```

```
const fs = require('fs'); //引入文件操作模块
```

// 通过用户请求的地址来读取相应的文件

```
var server = http.createServer(function(req,res){
```

```
  var file_name = './liu'+req.url; //获取请求的地址 组装成文件名
```

```
  fs.readFile(file_name,function(err,data){ //读取文件
```

```
    if(err){ //判断 如果用户访问的地址不存在则返回404
```

```
      res.write('404');
```

```
    }else{
```

```
      res.write(data); //否则返回 访问的内容
```

```
    }
```

```
    res.end(); //读取结束
```

```
  });
```

```
});
```

```
server.listen('8080');
```

//请求的地址: http://localhost:8080/3.html or <http://localhost:8080/2.html>

```
const http = require('http'); // 引入HTTP模块
const fs = require('fs');      // 引入文件操作模块
// 通过用户请求的地址来读取相应的文件
var server = http.createServer(function(req,res){
  var file_name = './liu'+req.url; // 获取请求的地址 组装成文件名
  fs.readFile(file_name,function(err,data){ // 读取文件
    if(err){ // 判断 如果用户访问的地址不存在则返回404
      res.write('404');
    }else{
      res.write(data); // 否则返回 访问的内容
    }
    res.end(); // 读取结束
  });
});
server.listen('8080');
// http://localhost:8080/3.html or http://localhost:8080/2.html
```