

path.dirname()获取路径名

path.dirname () 获取文件的后缀

```
1  const { dirname } = require('path');
2  const path=require('path')
3  //获取文件名
4  path.dirname('C:\Users\万股有\Desktop\word game\pkgse\node,js\path\pashname.js')
5  // 获取文件后缀
6  console.log( path.extname('C:\Users\万股有\Desktop\word
game\pkgse\node,js\path\pashname.js'));
7
8  path.join()//拼接地址
9  path.relative()//拼接地址
10
11
12  // __dirname文件当前的目录
13  console.log(path.join(__dirname, 'public'));
14  console.log(path.resolve(__dirname, 'public'));
```

__dirname () 当前文件的目录

```
1  console.log(__dirname);
2
```

filename () 获取当前文件的文件名

```
1  console.log(__filename);
```

path.join()拼接地址

```
1  console.log(path.join(__dirname, 'public'))
```

path.resolve()拼接地址

```
1  console.log(path.resolve(__dirname, 'public'));
```

join和relonse在拼接地址的时候一般情况效果是一样的但是如果当前目录参照变了效果不一样

判断请求的路径和方法，通过不一样的路径和方法响应给客户端

```
1  const http = require('http')
2  const fs = require('fs')
3  const path = require('path')
4  const { log } = require('console')
5  //创建的服务器
6  const servse = http.createServer()
7  //监听请求事件
8  servse.on('request', (req, res) => {
9      //req请求shuju request 请求对象
10     //res 响应数据 response响应对象
11     // res.setHeader('Content-Type', 'text/html;charset=UTF-8') //设置请求头不然会乱码
12     // res.end('王德广逛窑子') //响应的数据
13     console.log(req.url, req.method);
14     console.log(__dirname);
15     if (req.url === '/index.html' && req.method === 'GET') {
16         fs.readFile(path.resolve(__dirname, 'jack', 'index.html'), 'utf-8', (error,
17         data) => {
18
19             if (error) throw error
20             res.setHeader('Content-Type', 'text/html;charset=utf-8')
21             res.end(data)
22         })
23     }
24     else if (req.url === '/index.css' && req.method === 'GET') {
25         fs.readFile(path.resolve(__dirname, 'jack', 'index.css'), 'utf-8', (error, data)
26         => {
27             if (error) throw error
28             res.setHeader('Content-Type', 'text/html;charset=utf-8')
29             res.end(data)
30         })
31     }
```

```
31
32     }
33
34
35 })
36 //设置服务的端口
37 servse.listen(8080, () => {
38     console.log('服务已启动端口是8080');//给命令行使用
39 })
```