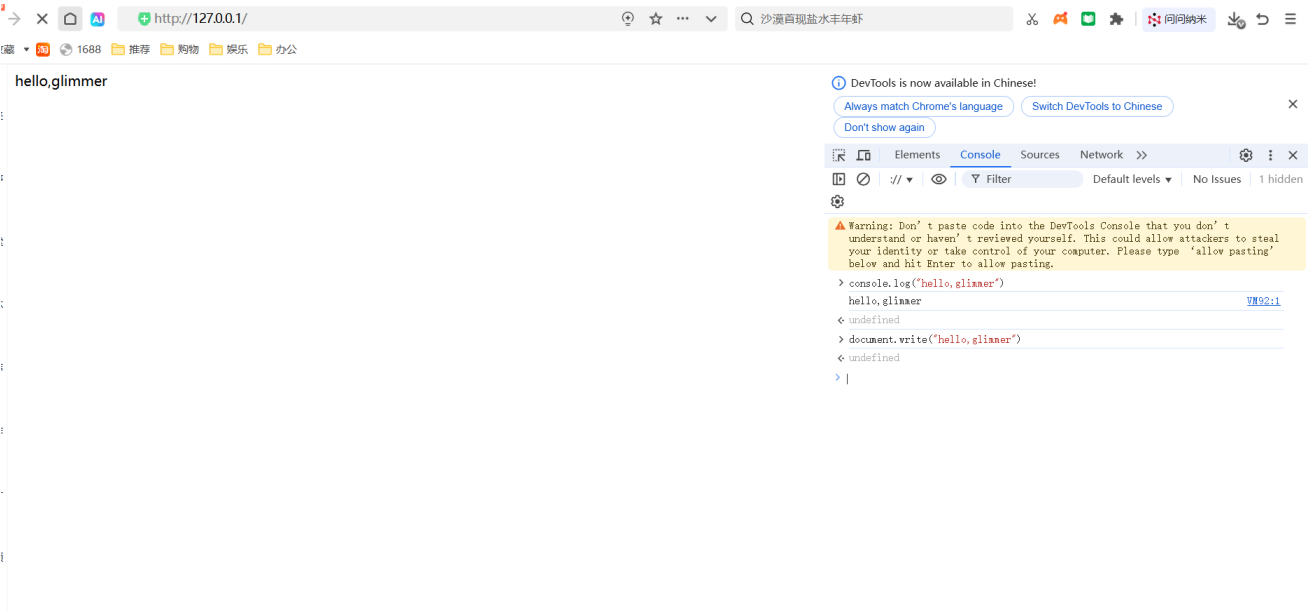
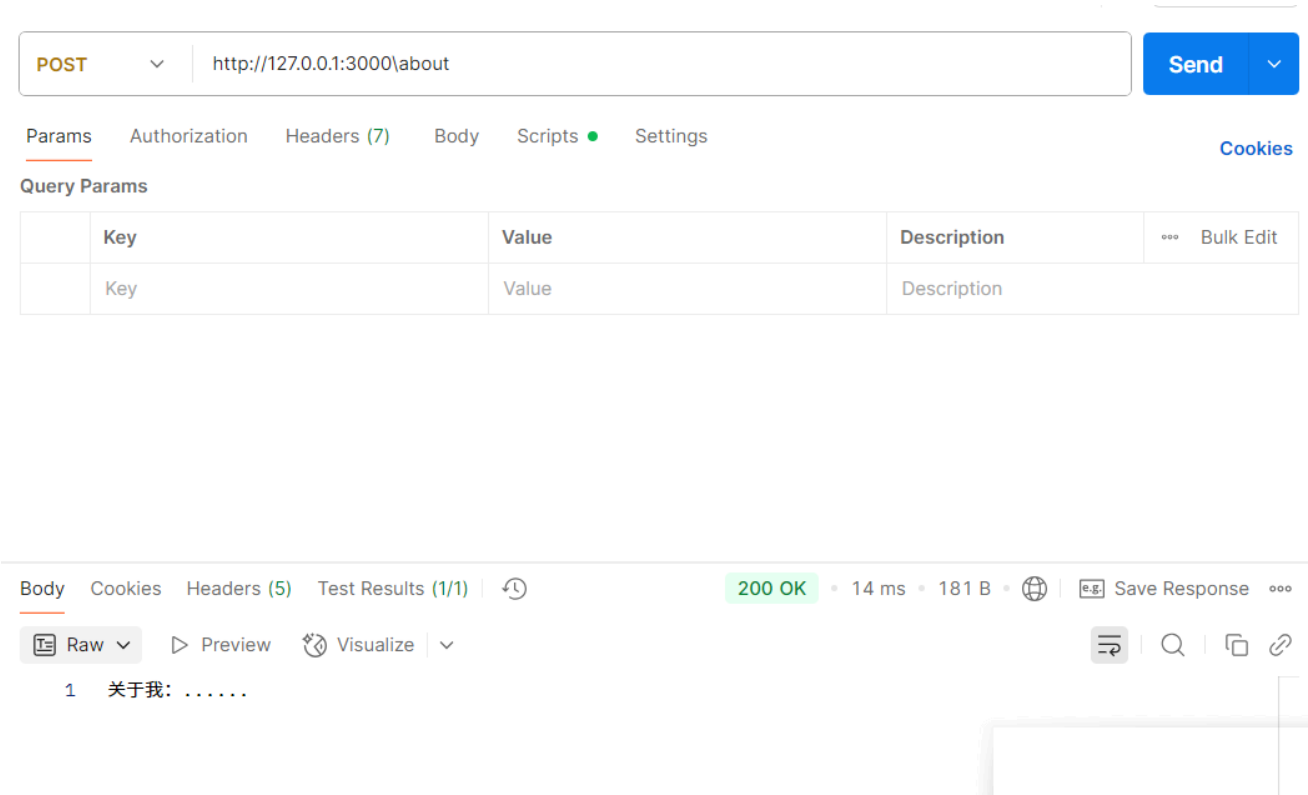


第七题

1. 在命令框和浏览器上显示



2. 期望输出



![75829346296](C:\Users\gguod\AppData\Local\Temp\1758293462965.png)

POST

▼

http://127.0.0.1:3000

Send

▼

ParamsAuthorizationHeaders (7)BodyScripts●Settings

Cookies

Query Params

	Key	Value	Description	⋮ Bulk Edit
	Key	Value	Description	

BodyCookiesHeaders (5)Test Results (1/1)

↺

200 OK

• 4 ms • 181 B •

🌐

📄 Save Response

⋮

📄 Raw ▼▶ Preview🔍 Visualize ▼

🔍 | 🔍 | 📄 | 🔗

1 这是一个首页

POST

http://127.0.0.1:3000/date

Send

ParamsAuthorizationHeaders (7)BodyScripts ●Settings

Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (5)Test Results (1/1)

200 OK • 5 ms • 205 B • Save Response

RawPreviewVisualize

1 现在是2025-10-05, 北京时间14:30:22

POST

http://127.0.0.1:3000/hhh

Send

ParamsAuthorizationHeaders (7)BodyScripts ●Settings

Cool

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookiesHeaders (5)Test Results (1/1)

200 OK • 5 ms • 181 B • Save Response

RawPreviewVisualize

1 页面不存在QwQ

3. 代码

```
C: > Typora > JS node.js > server.on('request') callback
1  const http = require('http')
2
3  const server=http.createServer()
4
5  server.on('request',(req,res)=>{
6
7      const url=req.url
8
9      let content='页面不存在QwQ'
10
11     if(url=== '/'){
12
13         content='这是一个首页'
14     }
15
16     else if(url=== '/about'){
17
18         content='关于我: .....{
34
35     console.log('server running at http://localhost:3000')
36
37 })
```

4. 心得

4.1 显示url和方法类型

C: > Typora > JS node.js > ...

```
1  const http = require('http')
2  const server=http.createServer()
3
4  server.on('request',req=>{
5      const url=req.url
6      const method=req.method
7      const str=`url:${url},method:${method}`
8      console.log(str)
9  })
10
11
12
13  server.listen(3000,()=>{
14      console.log(`server running at http://localhost:3000`)
15  })
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

gguod@DESKTOP-OKDI8QB MINGW64 /c/1111/.vscode

\$ cd c:\Typora

gguod@DESKTOP-OKDI8QB MINGW64 /c/Typora

\$ node node.js

server running at http://localhost:3000

url:/,method:GET

url:/,method:POST

url:/,method:POST

4.2 笔记

1.新建一个js文档

打开终端 cmd

先输入cd 空格路径/在文件夹空白处shift+右键 选在powershell中打开

输入node空格要执行的js文件路径

2.快捷键 上箭头 返回上一次命令

比如上一次输入node 1.js修改之后再执行这个文件

文件名称只打第一个 然后tab可以补全

esc清空已有

3.读取和写入

先引入fs const fs=require("fs")

读取: fs.readFile(文件路径, 以什么编码格式读取文件, 之后的回调函数function(错误err, 成功))

err.message表示输出错误的相关信息写入文件内容

导入fs const fs=require('fs')

调用 fs.writeFile(路径, 要写入的内容, utf8, 回调函数function(err, datastr){

if(err){

return console.log('文件写入失败')}

else

console.log('文件写入成功'))}

在终端中node 1.js查看

4.路径错误: 提供完整路径--右键文件 复制路径 把单/改成//

__dirname+'相对路径'--代表目录+相对路径

5.path路径模块

导入const path=require('path')

path.join拼接一个完整的路径

fs.readFile(path.join(__dirname, 'a', 'b', '..', 'd'), 'utf8', function(err, datastr){console.log(err)})

path.basename

获取文件名 const fpath=路径

const fullName=path.basename(fpath)

不拿到文件扩展名

const name=path.basename(fpath, 'html')

只拿到文件扩展名 path.extname

const html=path.extname(fpath)

6.案例

导入fs path 模块

匹配标签的正则 const regStyle=/

