**实验二 异步编程与文件操作**

一、实验目的：

1. 了解同步和异步的概念
2. 掌握使用回调函数处理异步函数的异常
3. 了解fs模块的作用。
4. 了解文件写入的同步方法
5. 掌握文件写入的异步方法
6. 掌握文件读取的方法

二、实验内容：

1. 异步编程程序执行方式

编写下列代码，并执行查看结果

**function abc(){**

**console.log("我是ABC函数的执行输出")**

**}**

**console.log("开始")**

**setTimeout(function(){**

**console.log("异步执行")**

**},0)**

**console.log("中间")**

**setTimeout(function(){**

**abc()**

**},0)**

**console.log("结束")**

回答下列问题：

1. 上述代码有哪几个地方是异步执行的

**setTimeout就是异步执行的**

1. 要让某功能异步执行，方式是什么

**将该功能放在异步函数内**

1. 回调函数的定义及使用

编写下列代码，并执行查看结果

**function fa(a){**

**return a\*a**

**}**

**function fb(x,y, callback){**

**var z=x+y**

**var t= callback(x)**

**return z+t**

**}**

**c=fb(5,6,fa)**

**console.log(c)**

回答下列问题：

1. 什么是回调函数

**当一个函数作为参数传递，那么这个函数就叫做回调函数**

1. 如何定义回调函数，什么时候会调用回调函数

**回调函数在外部定义，主函数调用回调函数时，回调函数即会执行**

1. 异步执行回调函数

function main(info,cb){

    console.log('孩子，快回家吧');

    setTimeout(function(){

        cb(info) ;//调用回调函数

    },0)

}

function sendMsg(msg){  //我是回调函数

    for(var i=0;i<10;i++){

       console.log("我在工作 ")

    }

    console.log(msg);

}

main('我到家了！',sendMsg);//执行主函数

console.log("妈妈去做其他事")

回答下列问题：

简要解释本例子输出结果，并给出一些心得

**Main与console.log("妈妈去做其他事")同步执行，优先执行main函数，在main函数中同步任务console.log('孩子，快回家吧')优先执行，执行完后异步任务等待，同步任务console.log("妈妈去做其他事")执行，再执行setTimeout异步任务，调用回调函数，回调函数中都是同步任务，根据调用顺序执行。**

1. 同步和异步写入文件

var fs=require("fs")

var data="hello world你好"

fs.writeFile(“demo1.txt”,data,function(error ){

    console.log("异步写入完成")

})

fs.writeFileSync("demo2.txt",data,"utf8")

console.log("同步写入完成")

回答下列问题：

异步写入文件的方法名叫什么，方法的参数有何意义

**方法名：writeFile 参数一：文件路径 参数二：写入的内容 参数三：回调函数**

1. 异步读出文件

var fs = require("fs")

fs.readFile("demo.txt", function (error, data) {

    if (error) {

        console.log("文件写入有误")

    }

    else{

         console.log(data.toString())

    }

})

console.log("aaaa")

回答下列问题：

异步读出文件的方法名叫什么，方法的参数有何意义，读出的数据在哪里

**方法名：readFile 参数一：文件路径 参数二：读取文件采用的编码格式 参数三：回调函数**

1. 异步读写JSON文件
2. 编写一个person.json文件，内容如下

[

    {

        "name": "张三",

        "ages": 28

    },

    {

        "name": "李四",

        "ages": 25

    },

    {

        "name": "王五",

        "ages": 30

    }

 ]

1. 编写读出JSON文件的js文件如下

const fs = require('fs');

fs.readFile('person.json',function(err,data){

    if(err) throw err;

   // console.log(data.toString())

    var person =JSON.parse(data.toString());//将字符串转换为JSON对象

    //对JSON数据进行处理

    for(var i = 0; i < person.length;i++){

       console.log('姓名：'+ person[i].name +'     年龄：'+ person[i].ages );

    }

});

1. 编写读写JSON文件的js文件如下

const fs = require('fs');

var newobj = { "name":"刘强", "ages":21 }

fs.readFile('person.json',function(err,data){

    if(err) throw err;

    var person =JSON.parse(data.toString());//将字符串转换为JSON对象

    person.push(newobj);//将新的对象加到数组对象中

    var str = JSON.stringify(person); //将JSON对象转换成字符串重新写入JSON文件中

    fs.writeFile('person.json',str,  function(err) {

       if(err) throw err;

       console.log("数据写入成功！");

    });

});

1. 编写一个程序读出person.json文件，将姓名为“李四“的数据删除，”将姓名为“刘强”的年龄改为100，最后写回到person.json中去

**const fs = require('fs');**

**fs.readFile('person.json', function (err, data) {**

**if (err) throw err;**

**// console.log(data.toString())**

**var person = JSON.parse(data.toString());//将字符串转换为JSON对象**

**//对JSON数据进行处理**

**for (var i = 0; i < person.length; i++) {**

**if (person[i].name === '刘强') {**

**person[i].ages = 100**

**}**

**}**

**for (var i = 0; i < person.length; i++) {**

**if (person[i].name === '李四') {**

**person.splice(i, 1)**

**}**

**}**

**var str = JSON.stringify(person); //将JSON对象转换成字符串重新写入JSON文件中**

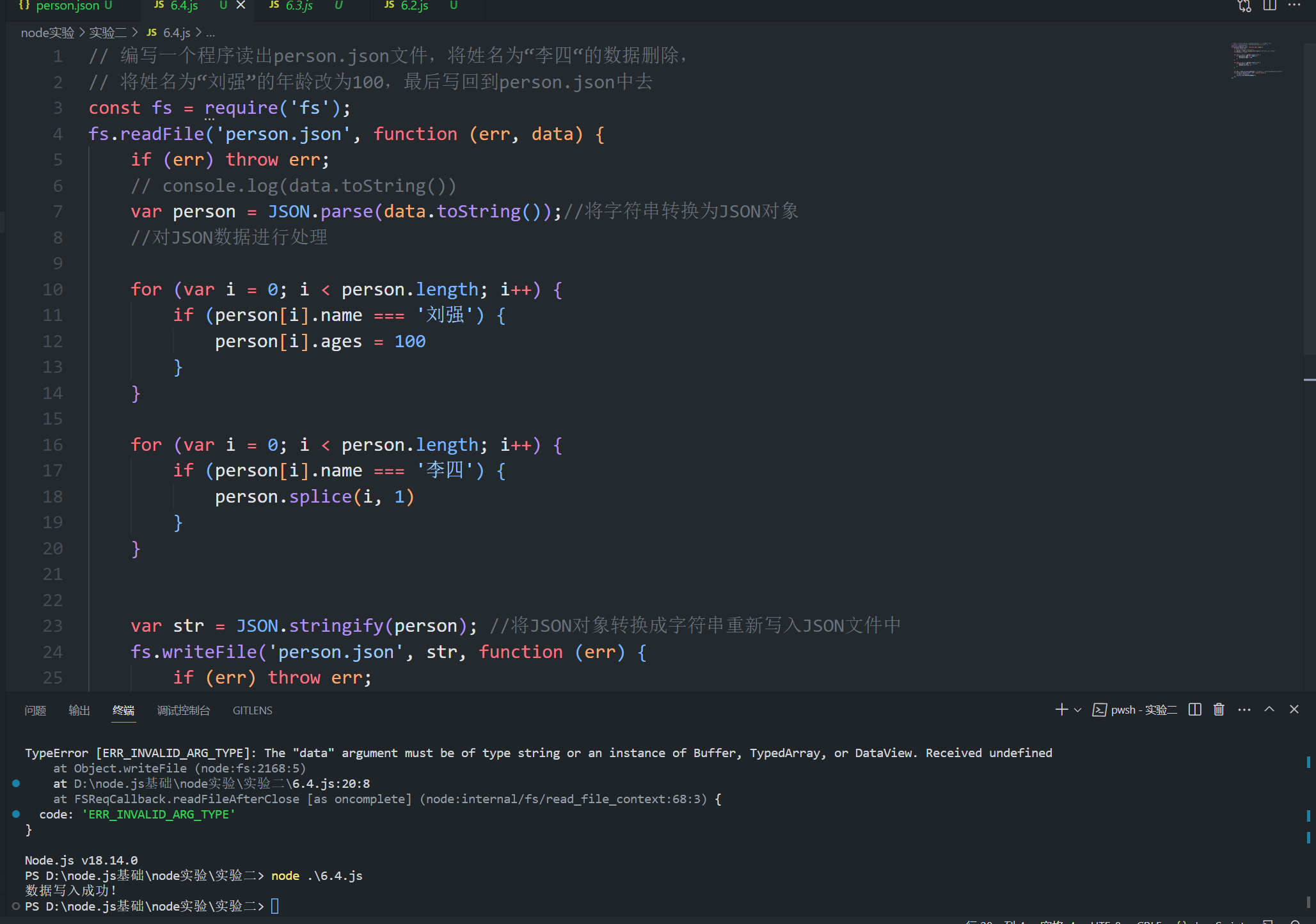
**fs.writeFile('person.json', str, function (err) {**

**if (err) throw err;**

**console.log("数据写入成功！");**

**});**

**});**

****

****

1. 准备两个文件person1.json、person2.json，内容和person,json类似，编程将这两个文件内容读出，并合并写入person3.json

**const fs = require('fs');**

**fs.readFile('person1.json', function (err, data) {**

**if (err) throw err;**

**var person1 = JSON.parse(data.toString());//将字符串转换为JSON对象**

**var str1 = JSON.stringify(person1); //将JSON对象转换成字符串重新写入JSON文件中**

**fs.readFile('person2.json', function (err, data) {**

**if (err) throw err;**

**var person2 = JSON.parse(data.toString());//将字符串转换为JSON对象**

**var str2 = JSON.stringify(person2); //将JSON对象转换成字符串重新写入JSON文件中**

**var str = []**

**for (i = 0; i < person1.length; i++) {**

**str.push(person1[i])**

**}**

**for (j = 0; j < person2.length; j++) {**

**str.push(person2[j])**

**}**

**var str3 = JSON.stringify(str)**

**fs.writeFile('person3.json', str3, function (err) {**

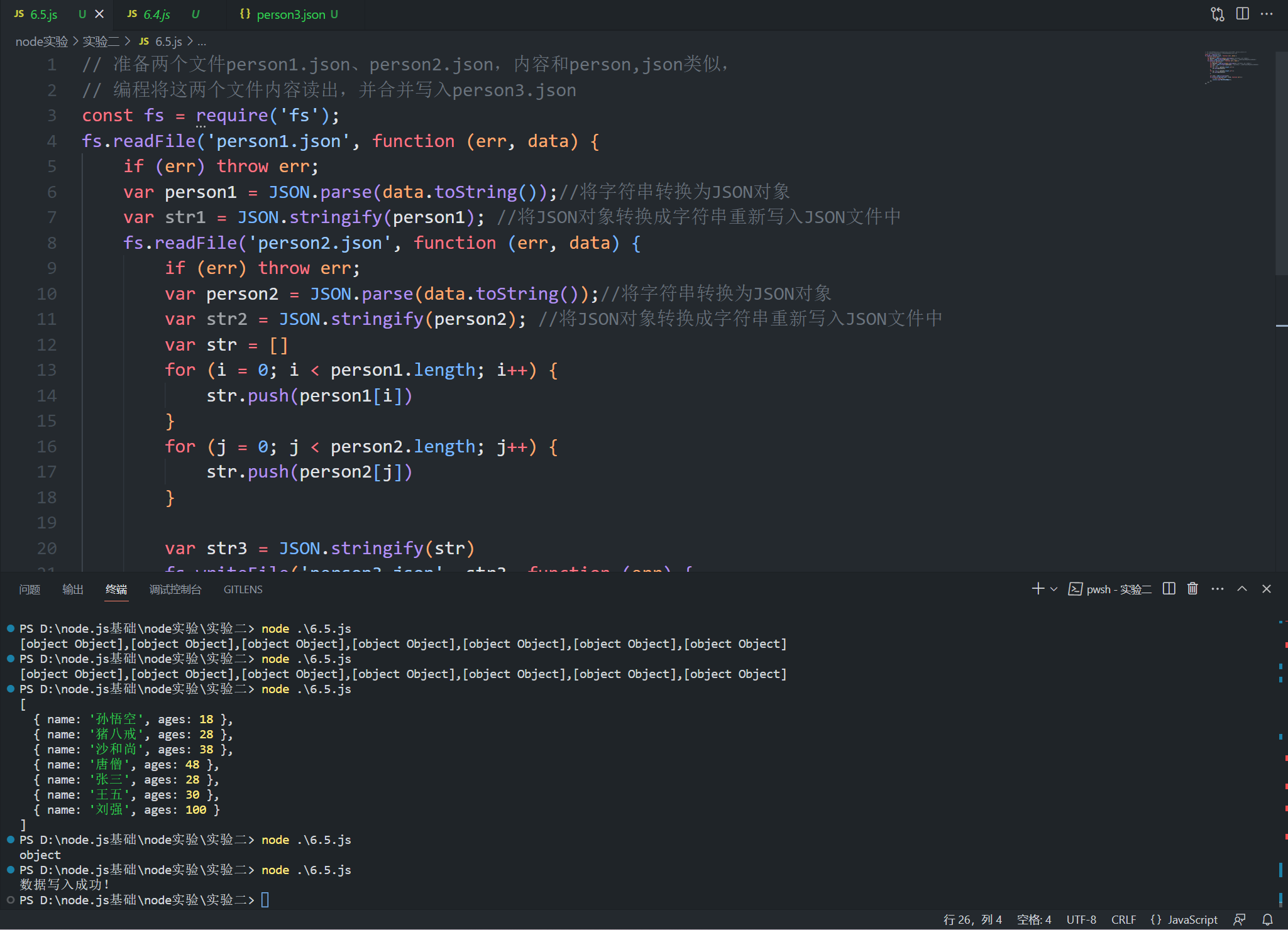
**if (err) throw err;**

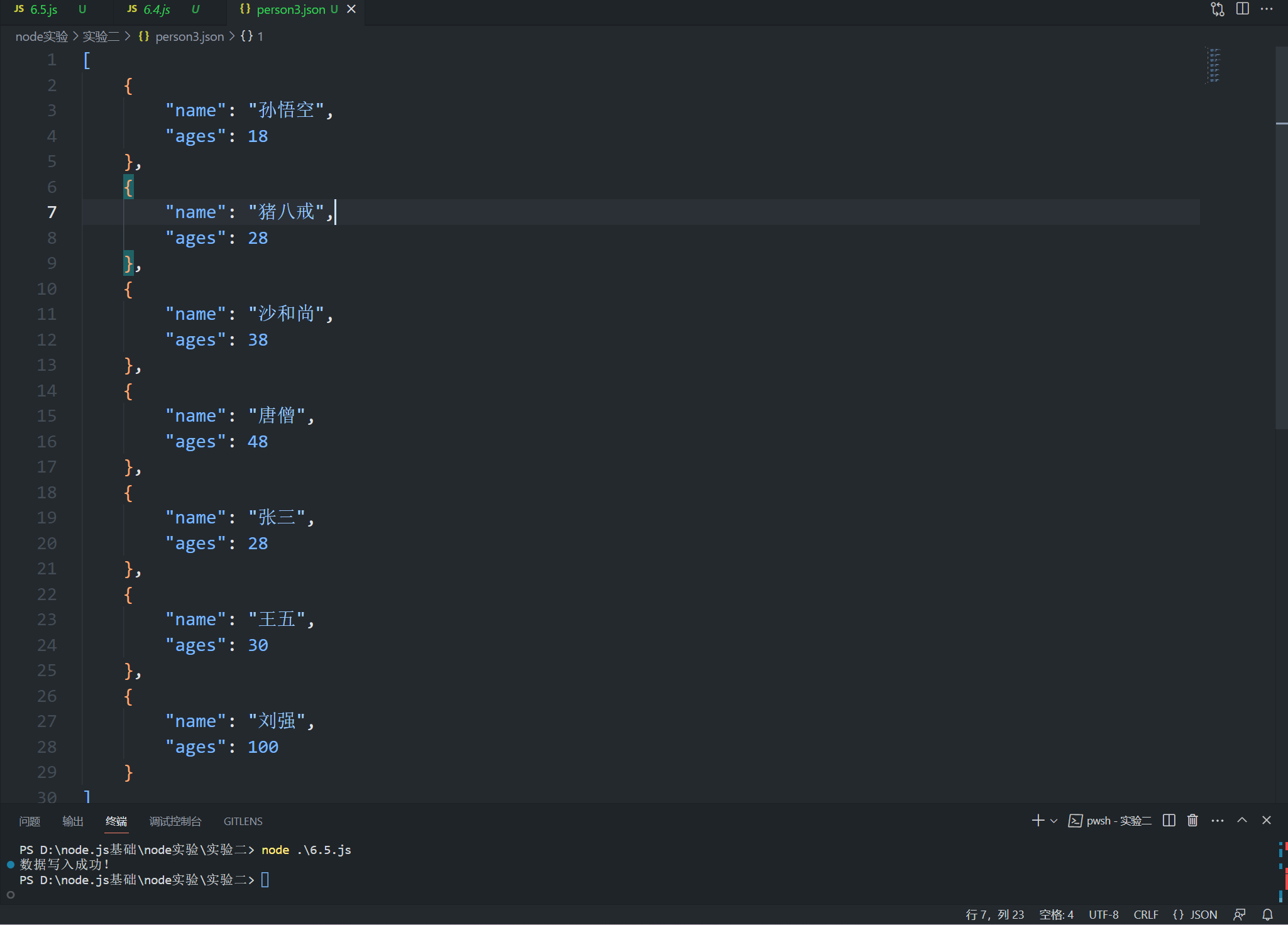
**console.log("数据写入成功！");**

**});**

**});**

**});**

****

****

三．实验小结

**本次实验让我们知道了什么是异步编程，什么是回调函数，知道了fs模块，了解了对文件的一些操作，这是我们学习的第一个node内置的模块，有助于培养我们模块化编程的思想，为今后的学习打下基础。**