

1 下列说法错误的有哪些（ ）

- A. Node.js本质是从Chrome浏览器中独立出来的V8引擎，用于解释执行JS。
- B. Node.js与HTML页面中使用的JS最大的区别是运行于服务器端。
- C. Node.js兼容所有在HTML页面中可以使用的JS对象，并添加了更多的对象。
- D. Node.js在解释执行脚本时采用的单线程执行，相较于其他采用并发多线程模型的开发语言，有一定的不足。
- E. Node.js中的全局对象不是window，而是global。

参考答案,

本题正确答案为CD。

C选项说法错误。网页中可以使用的JS对象包括EcmaScript原生对象、BOM和DOM对象；Node.js中支持所有的ES对象，但服务器端脚本不需要使用BOM和DOM对象，故所有的window、document等等对象是不支持的。

D选项说法错误。Node.js的最大特点是单线程异步回调，主线程不会被IO、网络请求等耗时的操作阻塞，相较于其它采用多线程但同步执行的服务器，在很多场景下不但处理速度更加卓越，还能处理更大的并发量。

2 简述setTimeout(fn,0)、setImmediate(fn)和process.nextTick(fn)的异同。

这三个方法都可以实现立即尽快异步执行指定的方法。setTimeout(fn,0)和setImmediate(fn)将指定事件处理函数追加到下次事件循环队列的头部，下次事件循环一吗开始立即执行；process.nextTick(fn)把指定事件处理函数追加到当前事件循环队列的尾部，所以会先于前两者执行。

3 根据要求编写代码。

创建一个Node.js文件模块，名为serverResponse，包含两个方法response404和response500，分别返回404的响应页面字符串和500的响应页面字符串；主模块中引入该文件模块，分别调用上述两个方法，输出它们的返回值。

参考答案

实现此案例可按照如下步骤进行。

步骤一：新建serverResponse.js文件

新建serverResponse.js文件，在该文件中创建response404和response500两个成员方法，并导出。代码如下所示：

```

02.      * 返回404响应页面必需的字符串数据
03.      */
04.      function response404() {
05.          var buf = new Buffer(`
06.              <!DOCTYPE html>
07.              <html>
08.              <head lang="en">
09.                  <meta charset="UTF-8">
10.              </head>
11.              <body>
12.                  <h1>404 - Page Not Found</h1>
13.                  <p>The page you requested was not found.</p>
14.              </body>
15.              </html>
16.          `);
17.          return buf.toString();
18.      }
19.      /**
20.      * 返回500响应页面必需的字符串数据
21.      */
22.      function response500() {
23.          var buf = new Buffer(`
24.              <!DOCTYPE html>
25.              <html>
26.              <head lang="en">
27.                  <meta charset="UTF-8">
28.              </head>
29.              <body>
30.                  <h1>500 - Server Error</h1>
31.                  <p>The server encountered runtime error(s), please try again later.</p>
32.              </body>
33.              </html>
34.          `);
35.          return buf.toString();
36.      }
37.
38.      //对外导出成员方法
39.      module.exports.response404 = response404;
40.      module.exports.response500 = response500;

```

在同一级目录下，创建index.js，引入serverResponse.js，调用其导出的response404和response500两个成员方法。代码如下所示：

```
01.    const serverResponse = require('./serverResponse');
02.
03.    //调用404响应处理过程
04.    var data404 = serverResponse.response404();
05.    console.log('404响应消息内容：')
06.    console.log(data404);
07.
08.    //调用500响应处理过程
09.    var data500 = serverResponse.response500();
10.    console.log('404响应消息内容：')
11.    console.log(data500);
```