Question1: What will be outputs in ES module and CommonJS module?

```
import fs from "fs"; // or const fs = require('fs');

process.nextTick(() => console.log("nextTick 1"));
Promise.resolve().then(() => console.log("promise 1"));

setImmediate(() => {
  console.log("setImmediate 1");
});

setTimeout(() => console.log("setTimeout 1"), 0);

fs.readFile("./files/input.txt", "utf-8", (err, data) => {
  if (err) console.log("there is an error. can not read from file");
  else {
    console.log(data);
  }
});
```

CommonJS module(reading file has error)

```
nextTick 1
promise 1
setTimeout 1
there is an error. can not read from file
setImmediate 1
```

CommonJS module(reading file has no error)

```
nextTick 1
promise 1
setTimeout 1
setImmediate 1
Read from input.txt
```

ES module

```
promise 1
nextTick 1
setImmediate 1
setTimeout 1
there is an error. can not read from file
const http = require("http");
const fs = require("fs");
```

Question 2: Create a web server using http module:

```
const path = require("path");

const server = http.createServer((req, res) => {
  const { url } = req;

  if (url === "/image") {
    const imagePath = path.join(__dirname, "..", "imgs", "happy_cat.png"); // replace with
your actual image path
    fs.readFile(imagePath, (err, data) => {
      if (err) {
        res.writeHead(500, { "Content-Type": "text/plain" });
        res.end("Error loading image");
      } else {
        res.writeHead(200, { "Content-Type": "image/jpeg" });
        res.end(data);
      }
    });
  } else if (url === "/pdf") {
    const pdfPath = path.join(__dirname, "..", "imgs", "document.pdf");
    fs.readFile(pdfPath, (err, data) => {
      if (err) {
        res.writeHead(500, { "Content-Type": "text/plain" });
        res.end("Error loading PDF");
      } else {
        res.writeHead(200, { "Content-Type": "application/pdf" });
        res.end(data);
      }
    });
  } else if (url === "/about") {
    const aboutPath = path.join(__dirname, "..", "imgs", "about.txt");
    fs.readFile(aboutPath, "utf8", (err, data) => {
      if (err) {
        res.writeHead(500, { "Content-Type": "text/plain" });
        res.end("Error loading text file");
      } else {
        res.writeHead(200, { "Content-Type": "text/plain" });
        res.end(data);
      }
    });
  } else if (url === "/" || url === "/home") {
    res.writeHead(200, { "Content-Type": "text/plain" });
```

```
    res.end("Welcome to my website");
  } else {
    res.writeHead(404, { "Content-Type": "text/plain" });
    res.end("404 Not Found");
  }
});

const PORT = 3000;
server.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
import fs from 'fs';
import { readFile } from 'fs/promises';
```

Question3: practice the fs.readFileSync(), fs.readFile(), fspromises.readFile(), and fs.createReadStream() methods. What are the differences?

```
// 1. Synchronous
const syncContent = fs.readFileSync('input.txt', 'utf8');
console.log('1. Sync:', syncContent);

// 2. Asynchronous (Callback)
fs.readFile('input.txt', 'utf8', (err, data) => {
 if (err) throw err;
 console.log('2. Async (callback):', data);
});

// 3. Asynchronous (Promise/async-await)
(async () => {
 const promiseContent = await readFile('input.txt', 'utf8');
 console.log('3. Async (promise):', promiseContent);
})();

// 4. Streaming
const stream = fs.createReadStream('input.txt', { encoding: 'utf8' });
stream.on('data', chunk => {
 console.log('4. Stream chunk:', chunk);
});
stream.on('end', () => {
 console.log('4. Stream finished.');
});
```