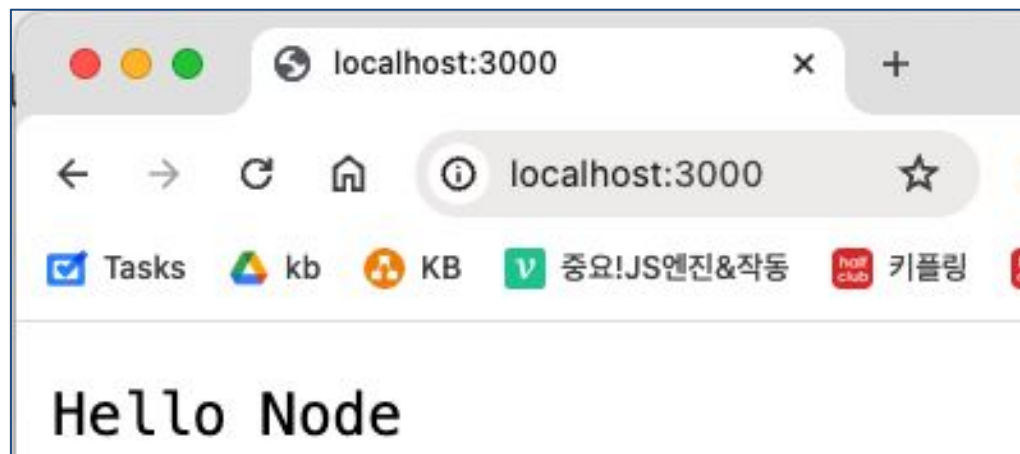


2024년 상반기 K-디지털 트레이닝

NodeJS 활용

[KB] IT's Your Life

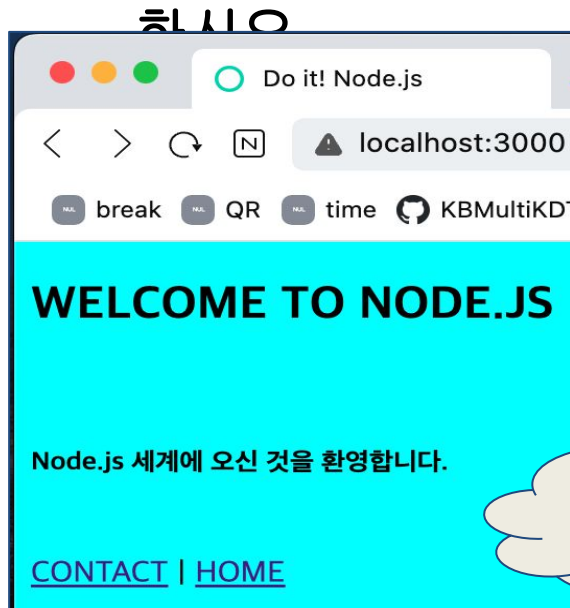
- http 모듈로 서버를 만들어 get요청시 다음과 같이 출력되도록 하시오.
- 브라우저의 네트워크탭에 요청된 파일의 상태 코드를 확인하시오.



이름	상태	유형	시작점	크기	시간	폭포	
localhost	200	doc...	기타	177 B	8밀...		
hook-exec.js	200	scri...	<u>hook.js:1</u>	6.9 ...	28...		
detector-exec.js	200	scri...	<u>detector.js:</u>	1.2 ...	2밀...		
favicon.ico	200	text...	기타	177 B	2밀...		

Q2 - http라우팅

- 각 요청 주소에 맞는 html파일을 서빙하여 요청시 브라우저에 출력되도록 프로그래밍

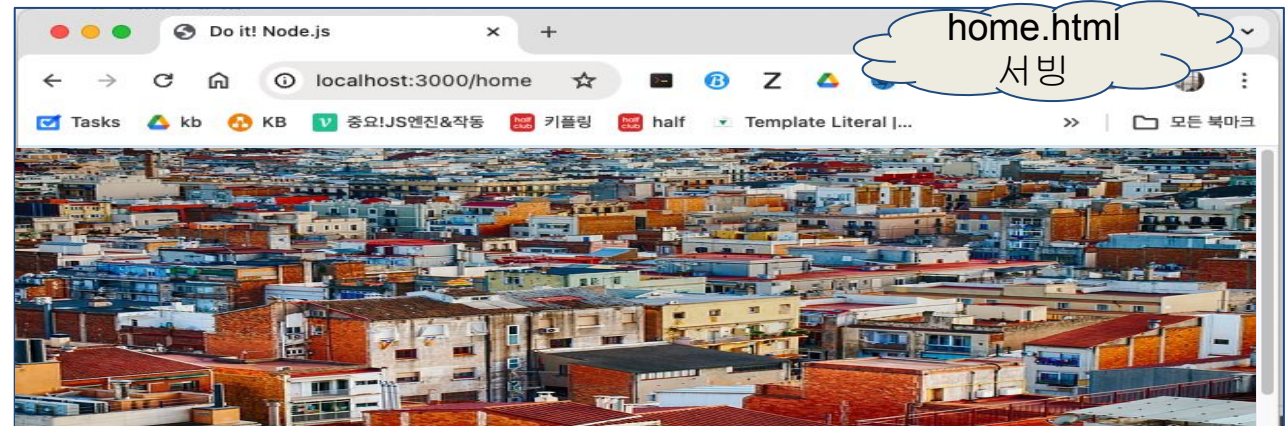


```
<a href="contact">CONTACT</a> |  
<a href="home">HOME</a>
```

index.html을
서빙하여
출력



image파일은
이미지 주소
복사를 사용!



Q3 - 동기와 비동기

- 다음과 같이 처리되는 이유는 무엇인가?
- 코드를 작성하여 실행해보고, 블로킹IO/논블로킹IO 관점에서 설명하시오.

```
const fs = require("fs");  
  
let data = fs.readFileSync("./example.txt", "utf-8");  
console.log(data);  
  
console.log("코드 끝!");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● alicia@Aliciaui-MacBookAir results % node blocking-1

t 엔진으로 빌드된 JavaScript 런타임입니다.Node.js는 크롬 V8
avaScript 런타임입니다.Node.js는 크롬 V8 Javascript 엔진으로
ipt 런타임입니다.Node.js는 크롬 V8 Javascript 엔진으로 빌드
ode.js는 크롬 V8 Javascript 엔진으로 빌드된 JavaScript 런타
8 Javascript 엔진으로 빌드된 JavaScript 런타임입니다.Node.j
엔진으로 빌드된 JavaScript 런타임입니다.

코드 끝!

Q4 - 동기와 비동기

- 다음과 같이 처리되는 이유는 무엇인가?
- 코드를 작성하여 실행해보고, 이유를 블로킹IO/논블로킹IO 관점에서 설명하시오.

```

3  const { log } = require("console");
4  const fs = require("fs");
5
6  // const data = fs.readFileSync("example.txt"); // 블로킹
7  // log(data.toString()); // 파일 읽기가 끝날 때까지 대기
8  // log("코드 끝!"); // 파일을 읽고 내용을 표시할 때까지 대기
9
10 const data2 = fs.readFile("example.txt", (error, readData) => {
11     if(error){
12         log("error");
13     }
14     log(readData.toString());
15 })
16 log("이걸 보고 싶은 거였어!! =====>> 코드 끝!");

```

example.txt	
388784	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388785	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388786	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388787	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388788	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388789	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388790	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388791	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388792	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388793	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388794	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388795	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388796	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388797	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388798	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388799	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388800	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388801	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.Node.js is
388802	Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다. ----

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

❗ alicia@Aliciaui-MacBookAir results % node blocking-1
이걸 보고 싶은 거였어!! =====>> 코드 끝 !
^C

```


- 다음과 같이 `readdir`을 모두 완료하기 전에 아래 코드가 실행되었다. `promise`를 사용하여 `readdir`후에 아래 코드가 실행되도록 수정하시오.

```
1 // 콜백 함수를 사용해 비동기 처리하기
2
3 const fs = require("fs");
4
5 let files = fs.readdir("./", (err, files) => {
6   if (err) {
7     console.error(err);
8   }
9   console.log(files);
10 });
11
12 console.log("Code is done.");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● alicia@Aliciaui-MacBookAir results % node async-3
Code is done.
[
  'async-1.js',
  'async-2.js',
  'async-3.js',
  'await.js',
  'blocking-1.js',
  'blocking-2.js',
  'blocking-3.js',
  'example copy.txt',
  'example.txt',
  'node-sync.js',
  'non-blocking.js',
  'promise.js',
  'sync.js'
]
```

Q5 - 동기와 비동기(이어서)

- 다음과 같이 `readdir`을 모두 완료하기 전에 아래 코드가 실행되었다. `async/await`를 사용하여 `readdir`후에 아래 코드가 실행되도록 수정하십시오.

```
1 // 콜백 함수를 사용해 비동기 처리하기
2
3 const fs = require("fs");
4
5 let files = fs.readdir("./", (err, files) => {
6   if (err) {
7     console.error(err);
8   }
9   console.log(files);
10 });
11
12 console.log("Code is done.");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● alicia@Aliciaui-MacBookAir results % node async-3
Code is done.
[
  'async-1.js',
  'async-2.js',
  'async-3.js',
  'await.js',
  'blocking-1.js',
  'blocking-2.js',
  'blocking-3.js',
  'example copy.txt',
  'example.txt',
  'node-sync.js',
  'non-blocking.js',
  'promise.js',
  'sync.js'
]
```

- 다음 사이트를 완성하시오.

** image는 url링크를 사용하시오.

홈으로 | 로그인 | 회사정보 | 회사위치

다이어트 전문 기업



홈으로 | 로그인 | 회사정보 | 회사위치

아이디 :
패스워드



홈으로 | 로그인 | 회사정보 | 회사위치



-다이어트보조식
-집중다이어트식
-단백질보조식

홈으로 | 로그인 | 회사정보 | 회사위치



대표번호	02-144
제품문의	02-155
위치안내	02-166
야간안내	02-177

