

웹 프로그래밍 강의노트

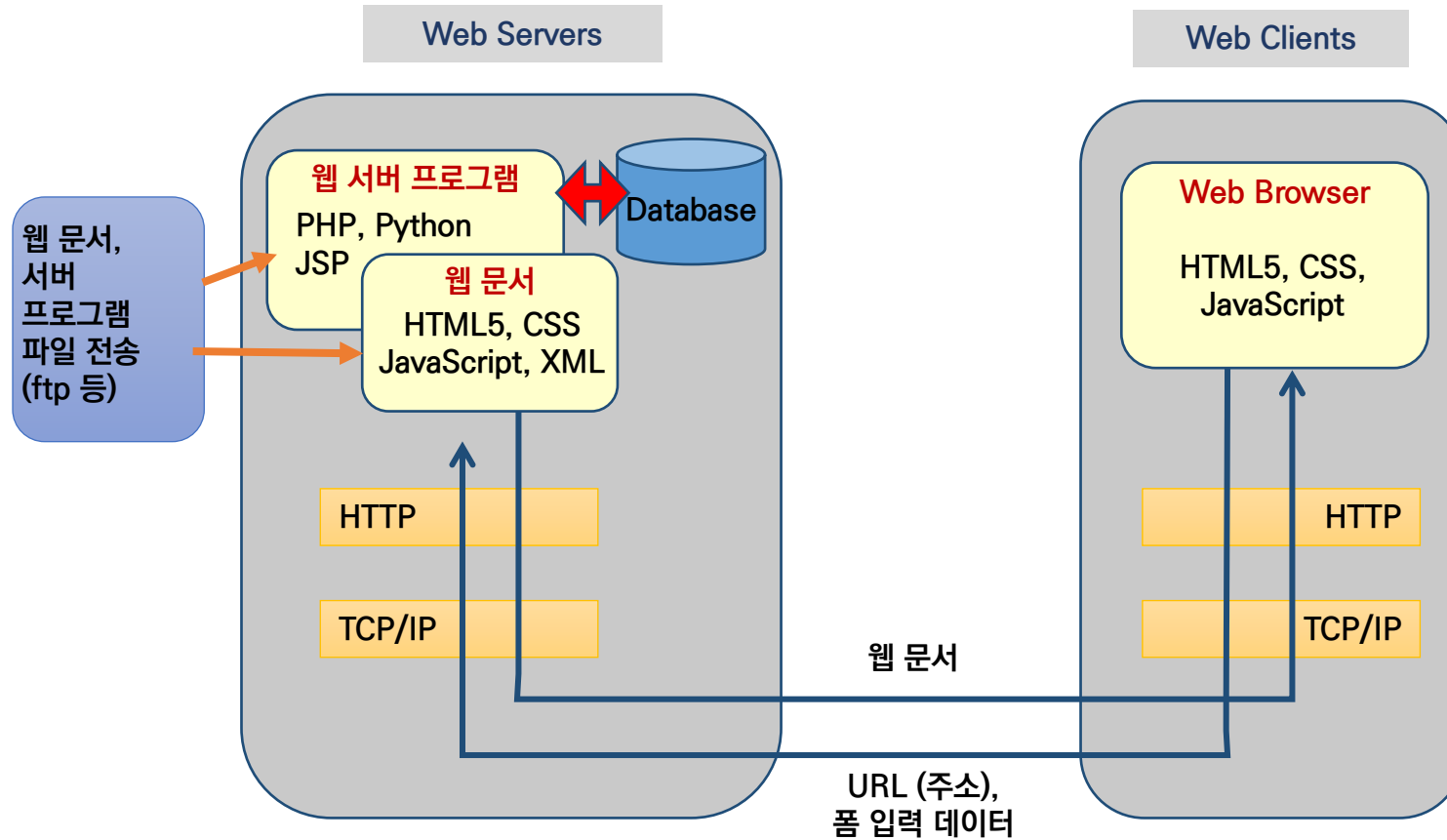
Node.js를 이용한 웹서버

2024년 2학기

웹 서버의 구축

- 웹 서버 컴퓨터 구축
 - 웹 서버 소프트웨어 설치 및 실행
 - 아파치(Apache) 웹 서버가 대표적
- 서버 실행 프로그램
 - HTML5 폼을 전달받아 처리
 - PHP, JSP, Python 등
- 데이터베이스 시스템
 - 서버 측에 데이터를 저장 및 효율적 관리
 - MySQL, SQL, Oracle 등

웹 서버-클라이언트 연결 구조




Node.js

- JavaScript runtime environment
- 원래의 JavaScript
 - 브라우저 내에서만 동작
- Node.js
 - 기계에서 독립적으로 JavaScript 구동하게 만듦

Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

Download Node.js (LTS) 

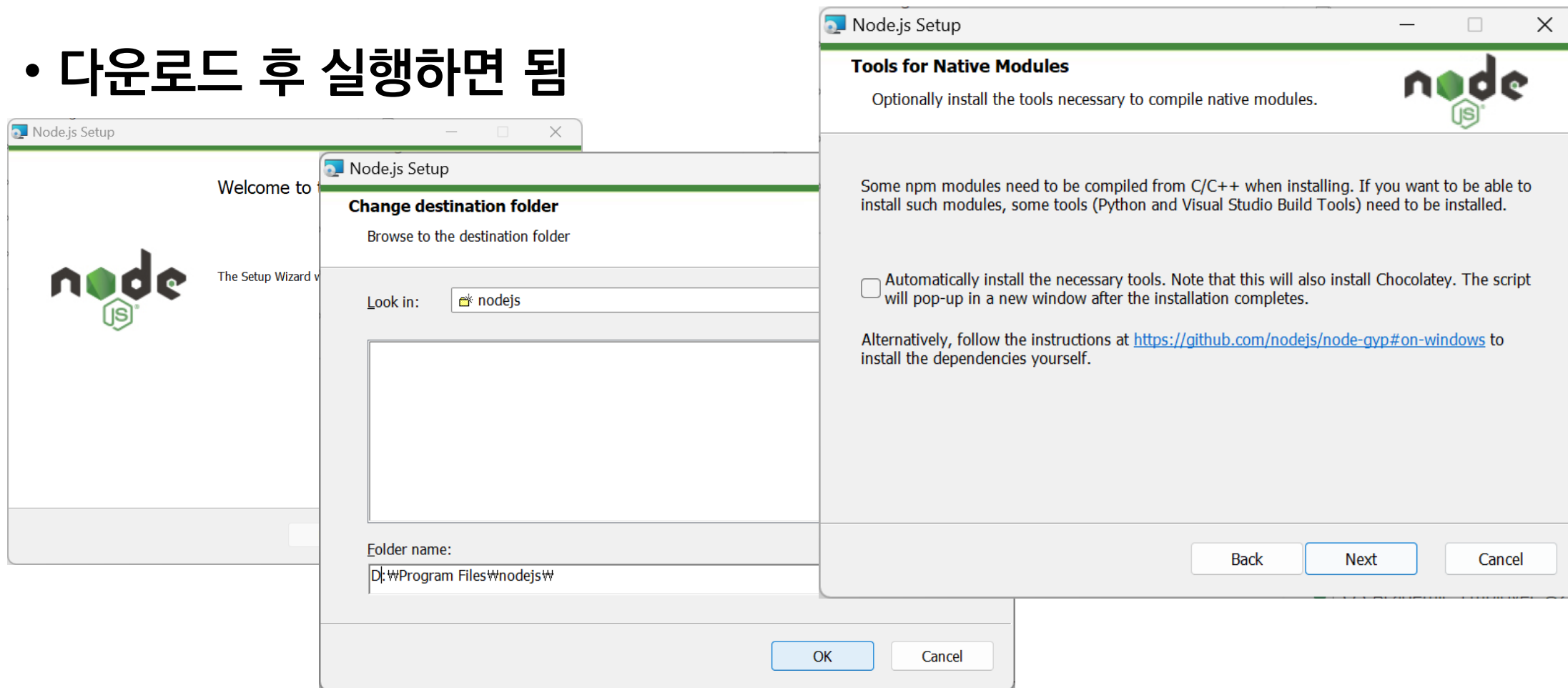
Downloads Node.js **v22.12.0**¹ with long-term support.
Node.js can also be installed via [package managers](#).

Want new features sooner? Get **Node.js v23.3.0**¹ instead.

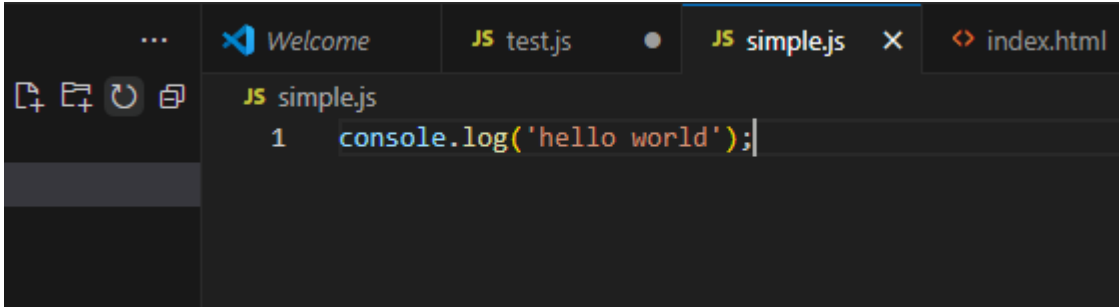
<https://nodejs.org/>

설치

• 다운로드 후 실행하면 됨

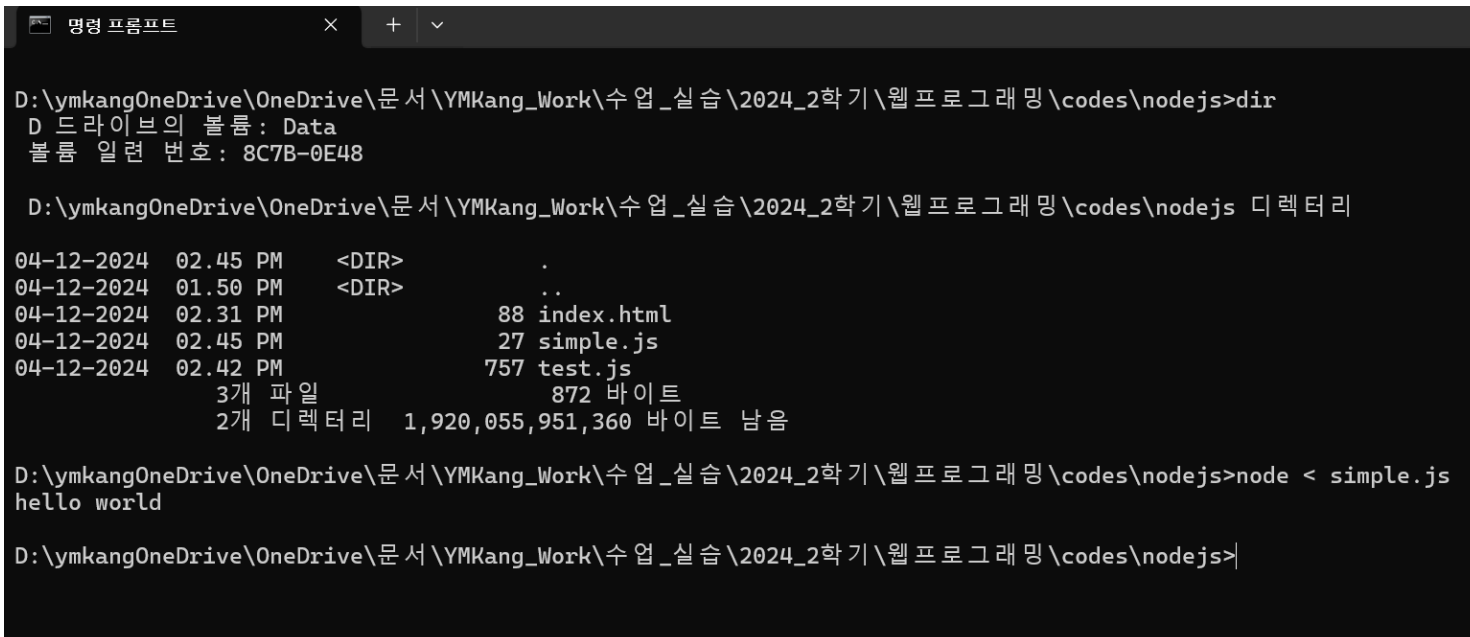


간단한 JavaScript 구현



```
JS simple.js
1 console.log('hello world');
```

simple.js



```
명령 프롬프트
D:\ymkangOneDrive\OneDrive\문서\YMKang_Work\수업_실습\2024_2학기\웹프로그래밍\codes\nodejs>dir
D 드라이브의 볼륨 : Data
볼륨 일련 번호: 8C7B-0E48

D:\ymkangOneDrive\OneDrive\문서\YMKang_Work\수업_실습\2024_2학기\웹프로그래밍\codes\nodejs 디렉터리

04-12-2024  02.45 PM    <DIR>          .
04-12-2024  01.50 PM    <DIR>          ..
04-12-2024  02.31 PM                88 index.html
04-12-2024  02.45 PM                27 simple.js
04-12-2024  02.42 PM               757 test.js
               3개 파일                872 바이트
               2개 디렉터리  1,920,055,951,360 바이트 남음

D:\ymkangOneDrive\OneDrive\문서\YMKang_Work\수업_실습\2024_2학기\웹프로그래밍\codes\nodejs>node < simple.js
hello world

D:\ymkangOneDrive\OneDrive\문서\YMKang_Work\수업_실습\2024_2학기\웹프로그래밍\codes\nodejs>
```

node simple.js

or

node < simple.js

```
const http = require('http'); // http 모듈 импорт
const fs = require('fs');      // file system 모듈 импорт
const path = require('path');  // 경로 관리 모듈 импорт
```

```
// Create the server
```

```
const server = http.createServer((req, res) => {
  // Set the file path
  const filePath = path.join(__dirname, 'index.html');

  // Read and serve the HTML file
  fs.readFile(filePath, 'utf-8', (err, content) => {
    if (err) {
      res.writeHead(500, { 'Content-Type': 'text/plain' });
      res.end('Internal Server Error');
    } else {
      res.writeHead(200, { 'Content-Type': 'text/html' });
      res.end(content);
    }
  });
});
```

```
// Start the server
```

```
const PORT = 3000;
server.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});
```

Web Server

내용 이해

```
const server = http.createServer((req, res) => {  
  // Code to handle the request and send a response  
});
```

- **http.createServer**
 - HTTP 서버 생성 메소드
 - req: request / 서버쪽에 도착하는 요청
 - res: response / 서버의 응답

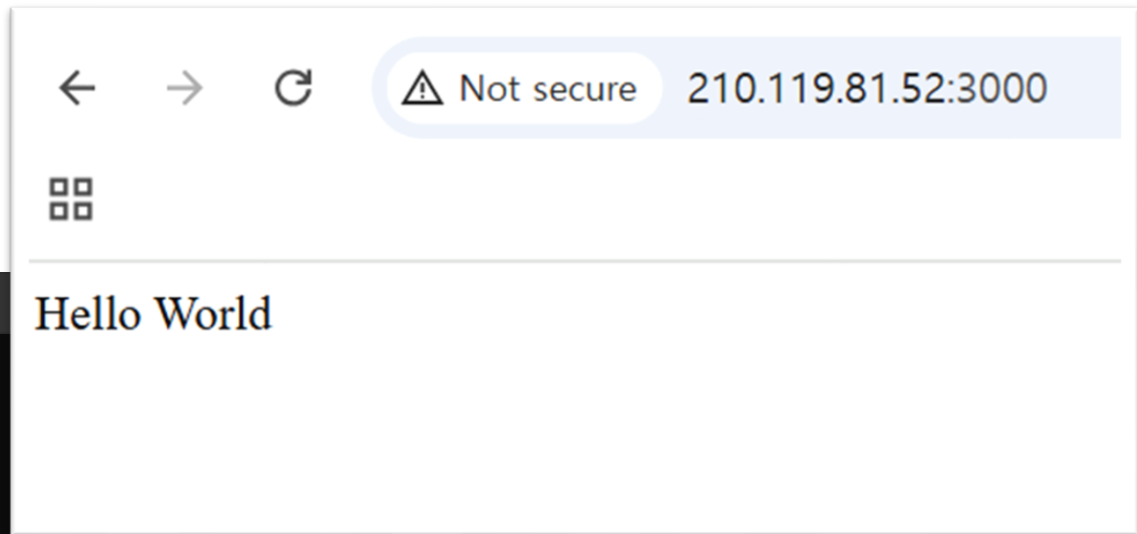
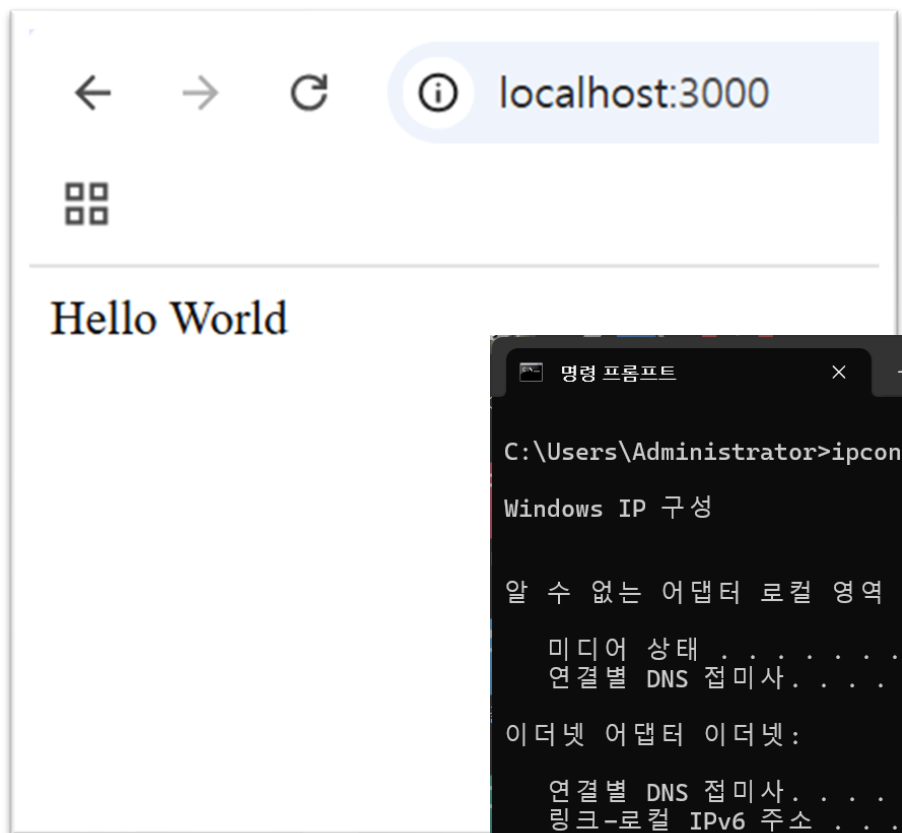
내용 이해

```
const filePath = path.join(__dirname, 'index.html');
```

서버의 디렉토리 내에 있는 index.html을 지정

```
fs.readFile(filePath, 'utf-8', (err, content) => {  
  if (err) { // Handle error }  
  else { // Serve the content }  
});
```

해당 파일을 읽어 들임



```
명령 프롬프트
C:\Users\Administrator>ipconfig

Windows IP 구성

알 수 없는 어댑터 로컬 영역 연결:

    미디어 상태 . . . . . : 미디어 연결 끊김
    연결별 DNS 접미사 . . . . :

이더넷 어댑터 이더넷:

    연결별 DNS 접미사 . . . . :
    링크-로컬 IPv6 주소 . . . : fe80::bb61:7fa3:47cb:322a%15
    IPv4 주소 . . . . . : 210.119.81.52
    서브넷 마스크 . . . . . : 255.255.255.0
    기본 게이트웨이 . . . . . : 210.119.81.1

C:\Users\Administrator>
```

다른 파일도
접근할 수 있는
서버 1/3

```
const http = require('http'); // Import the
HTTP module
const fs = require('fs'); // Import the File
System module
const path = require('path'); // Import the
Path module

// Mapping of file extensions to Content-Type
headers
const mimeTypeypes = {
  '.html': 'text/html',
  '.css': 'text/css',
  '.js': 'application/javascript',
  '.png': 'image/png',
  '.jpg': 'image/jpeg',
  '.jpeg': 'image/jpeg',
  '.gif': 'image/gif',
  '.json': 'application/json'
};
```

```

// Create the server
const server = http.createServer((req, res) => {
  // Parse the URL to get the requested file path
  let filePath = '.' + req.url; // Default to serving the file corresponding to the URL
  if (filePath == './') {
    filePath = './index.html'; // Default to index.html if no file is specified
  }

  // Get the file extension for setting the Content-Type
  const extname = path.extname(filePath);

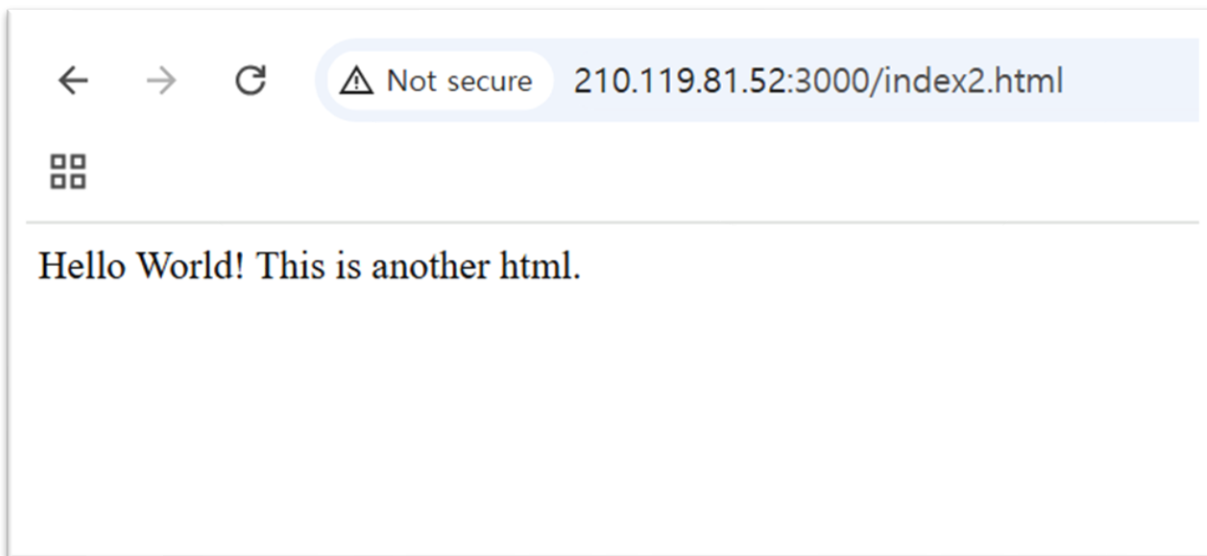
  // Read and serve the requested file
  fs.readFile(filePath, 'utf-8', (err, content) => {
    if (err) {
      // If the file is not found or another error occurs, send a 404 error
      if (err.code == 'ENOENT') {
        res.writeHead(404, { 'Content-Type': 'text/plain' });
        res.end('404 Not Found');
      } else {
        res.writeHead(500, { 'Content-Type': 'text/plain' });
        res.end('Internal Server Error');
      }
    } else {
      // Set the Content-Type based on the file extension
      res.writeHead(200, { 'Content-Type': mimeType[extname] || 'application/octet-stream' });
      res.end(content);
    }
  });
});

```

다른 파일도
접근할 수 있는
서버 2/3

다른 파일도 접근할 수 있는 서버 3/3

```
// Start the server
const PORT = 3000;
server.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});
```



계산기 서비스

