

Node.js

2022901090 최유진

목차

1. Node.js 소개

- Node.js란 무엇인가?
- 탄생 배경

2. Node.js의 특징

- 비동기 I/O
- 싱글 스레드
- 크로스 플랫폼

3. Node.js의 장점

- 빠른 처리 속도
- 개발 생산성 향상
- 활발한 커뮤니티

4. Node.js의 단점

- CPU 집약적 작업에 부적합
- 오류 관리의 어려움

5. Node.js의 주요 모듈

- http 모듈
- fs 모듈
- path 모듈
- express 모듈

6. Node.js 설치 및 환경 설정

- Node.js 설치 방법
- npm (Node Package Manager)

7. 간단한 예제 코드

- Hello World 서버 코드

8. Node.js의 활용 분야

- 웹 서버
- API 서버
- 실시간 애플리케이션

9. Node.js 커뮤니티 및 자료

- 공식 문서
- Stack Overflow 및 GitHub
- 유명한 프로젝트

1. Node.js 소개

- **Node.js란?**

JavaScript 런타임 환경으로, 브라우저가 아닌 서버에서도 JavaScript 코드 실행 가능.

- **탄생 배경**

JavaScript의 서버 측 사용을 위해 2009년에 개발됨.

2. Node.js 특징

- 비동기 I/O

: 이벤트 기반으로 빠르게 작업 처리.

- 싱글 스레드

: 단일 스레드를 통해 서버 자원 효율 사용.

- 크로스 플랫폼

: 여러 OS에서 실행 가능 (Windows, macOS, Linux).

3. Node.js의 장점

- **빠른 처리 속도**
: Google V8 엔진으로 빠른 JavaScript 실행.
- **개발 생산성 향상**
: 백엔드와 프론트엔드를 하나의 언어로 개발 가능.
- **활발한 커뮤니티**
: 다양한 오픈 소스 모듈과 라이브러리 사용 가능.

4. Node.js의 단점

- CPU 집약적 작업에 부적합
: 싱글 스레드 특성상 CPU 작업에 약함.
- 오류 관리 어려움
: 비동기 코드로 디버깅 복잡.

5. Node.js의 주요 모듈

- **http**
웹 서버 생성 및 관리.
- **fs**
파일 시스템 조작.
- **Path**
파일 경로 조작.
- **Express**
RESTful API와 웹 애플리케이션 구축을 돕는 프레임워크.

6. Node.js 설치 및 환경 설정

- **설치 방법**

공식 웹사이트에서 설치 가능, `node -v`로 확인.

- **npm**

패키지 관리 도구, 다양한 모듈을 설치하고 관리 가능.

7. 간단한 예제 코드

- 간단한 웹 서버로 "Hello World" 응답 제공.

javascript

```
const http = require('http');
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});
server.listen(3000, () => {
  console.log('서버가 http://localhost:3000에서 실행 중입니다.');
```

8. Node.js의 활용 분야

- 웹 서버
빠른 응답 속도로 다중 사용자 처리.
- API 서버
RESTful API 구축에 유용.
- 실시간 애플리케이션
채팅, 온라인 게임 등 실시간 데이터 처리에 적합.

9. Node.js 커뮤니티 및 자료

공식 문서

API와 기능에 대한 자세한 설명 제공.

Stack Overflow & GitHub

Node.js 관련 자료와 오픈 소스 모듈.

유명한 프로젝트

npm, Express, Socket.io 등.