

# Node.js 발표 자료

이 문서는 대학생을 위한 Node.js에 대한 발표 자료를 제공합니다. Node.js는 서버 사이드 애플리케이션을 구축하기 위한 JavaScript 런타임 환경으로, 비동기 이벤트 기반 아키텍처를 특징으로 합니다. 이 자료는 Node.js의 기본 개념, 특징, 설치 방법, 주요 모듈 및 활용 사례를 포함하고 있습니다.

## 1. Node.js란?

- 정의: Node.js는 Chrome의 V8 JavaScript 엔진을 기반으로 한 서버 사이드 JavaScript 런타임입니다.
- 특징:
  - 비동기 I/O 처리
  - 이벤트 기반 아키텍처
  - 단일 스레드 모델
  - 높은 성능과 확장성

## 2. Node.js의 장점

- 빠른 성능: V8 엔진 덕분에 JavaScript 코드가 빠르게 실행됩니다.
- 비동기 처리: I/O 작업을 비동기로 처리하여 높은 처리량을 자랑합니다.
- 모듈화: npm(Node Package Manager)을 통해 다양한 패키지를 쉽게 설치하고 관리할 수 있습니다.
- 크로스 플랫폼: Windows, macOS, Linux 등 다양한 운영체제에서 실행 가능합니다.

## 3. Node.js 설치 방법

- Node.js** 다운로드: [Node.js 공식 웹사이트](https://nodejs.org)에서 최신 버전을 다운로드합니다.
- 설치: 다운로드한 설치 파일을 실행하여 설치합니다.
- 확인: 터미널(명령 프롬프트)에서 **node -v** 명령어를 입력하여 설치된 버전을 확인합니다.

## 4. 기본 코드 예제

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello, World!\n');
});

const PORT = 3000;
server.listen(PORT, () => {
  console.log(`서버가 http://localhost:${PORT}/ 에서 실행 중입니다.`);
});
```

## 5. 주요 모듈

- http**: HTTP 서버 및 클라이언트를 생성하는 모듈
- fs**: 파일 시스템과 상호작용하는 모듈
- path**: 파일 및 디렉토리 경로를 처리하는 모듈
- express**: 웹 애플리케이션 프레임워크

## 6. 활용 사례

- 웹 서버: RESTful API 서버 구축
- 실시간 애플리케이션: 채팅 애플리케이션, 게임 서버
- 마이크로서비스: 분산 시스템 구축

## 7. 결론

Node.js는 비동기 처리와 높은 성능 덕분에 현대 웹 애플리케이션 개발에 적합한 플랫폼입니다. 다양한 모듈과 커뮤니티 지원으로 인해 개발자들이 쉽게 사용할 수 있는 환경을 제공합니다. 이 발표 자료를 통해 Node.js의 기본 개념과 활용 방법을 이해하는 데 도움이 되었기를 바랍니다.