



Node.js: 웹 개발의 새로운 지평

JavaScript를 기반으로 한 서버 측 개발 플랫폼인 Node.js를 알아보세요.

 작성자: 대완 강

Node.js의 매력적인 특징

비동기 I/O

효율적인 처리, 빠른 응답 시간

단일 스레드 모델

가볍고 효율적인 이벤트 루프 기반

모듈 시스템

다양한 기능 확장, 개발 속도 향상

Node.js

Node.js 설치와 환경 설정

1

1. 설치

공식 웹사이트에서 Node.js 설치

2

2. 버전 관리

nvm (Node Version Manager) 활용

3

3. 패키지 관리

npm (Node Package Manager) 사용



Node.js의 비동기 I/O

1

비동기 방식

동시에 여러 작업 처리 가능

2

콜백 함수

작업 완료 시 실행되는 함수

3

이벤트 루프

비동기 작업 관리, 효율적인 처리

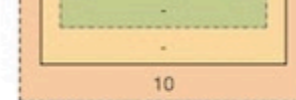


Get more done with the new Chrome

Now, more simple, secure, and faster than ever - with

```
.chr-headline-1, .chr-headline-1--medium, .chr-hero_heading, .home-dev .chr-hero_heading, .home-beta .chr-hero_heading, .home-canary .chr-hero_heading, .chr-stores_heading, .chr-privacy_container h1 {
  font-size: 2rem;
  line-height: 2.5rem;
}

.chr-headline-1, .chr-headline-1--medium, .chr-hero_heading, .home-dev .chr-hero_heading, .home-beta .chr-hero_heading, .home-canary .chr-hero_heading, .chr-stores_heading, .chr-privacy_container h1 {
  font-size: 2rem;
  line-height: 2.5rem;
}
```



Filter	<input type="checkbox"/> Show all
▶ box-sizing	border...
▶ color	■ rgb(...)
▶ display	block
▶ font-family	"Google..."
▶ font-size	40px
▶ font-weight	400
▶ height	99.2px
▶ letter-spacing	normal

웹 애플리케이션 개발



웹 서버

HTTP 요청 처리



데이터베이스

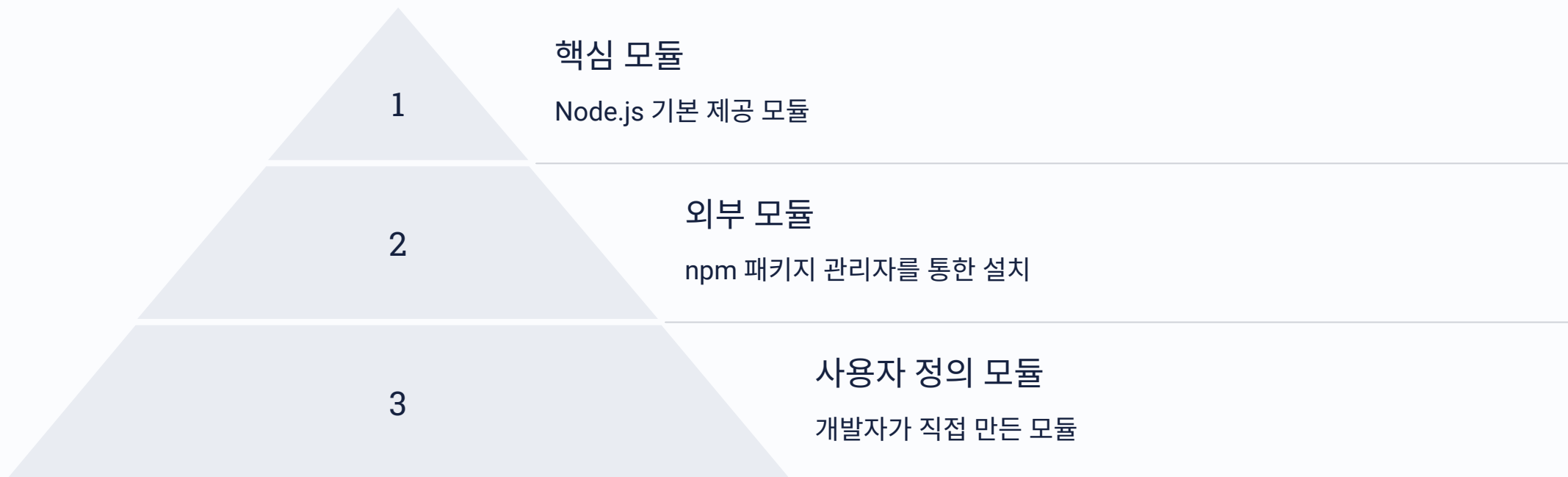
데이터 저장 및 관리



API 개발

다른 애플리케이션과의 통신

Node.js의 모듈 시스템





Node.js 이벤트 핸들링

1

이벤트 발생

특정 상황 발생 시 이벤트 발생

2

리스너 등록

이벤트 발생 시 실행될 함수 등록

3

이벤트 처리

이벤트 발생 시 등록된 리스너 실행



Node.js의 확장성과 스케일링

1

클러스터링

여러 CPU 코어 활용

2

로드 밸런싱

트래픽 분산

3

캐싱

데이터 캐싱