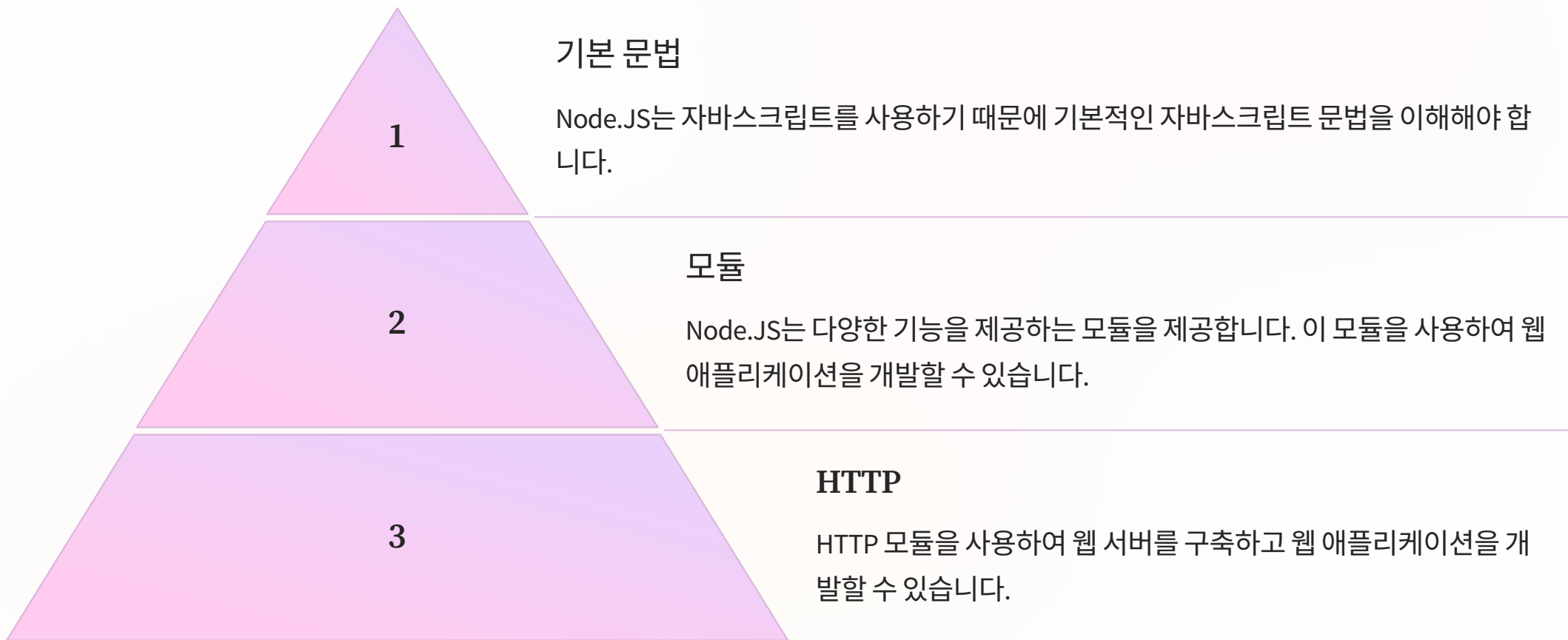




Unlocking the Power of Node.js: Building Scalable Web Applications



Node.JS 프로그래밍 기본



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



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

비동기 프로그래밍 이해

1

콜백 함수

비동기 작업이 완료되면 콜백 함수를 호출하여 결과를 처리합니다.

2

프로미스

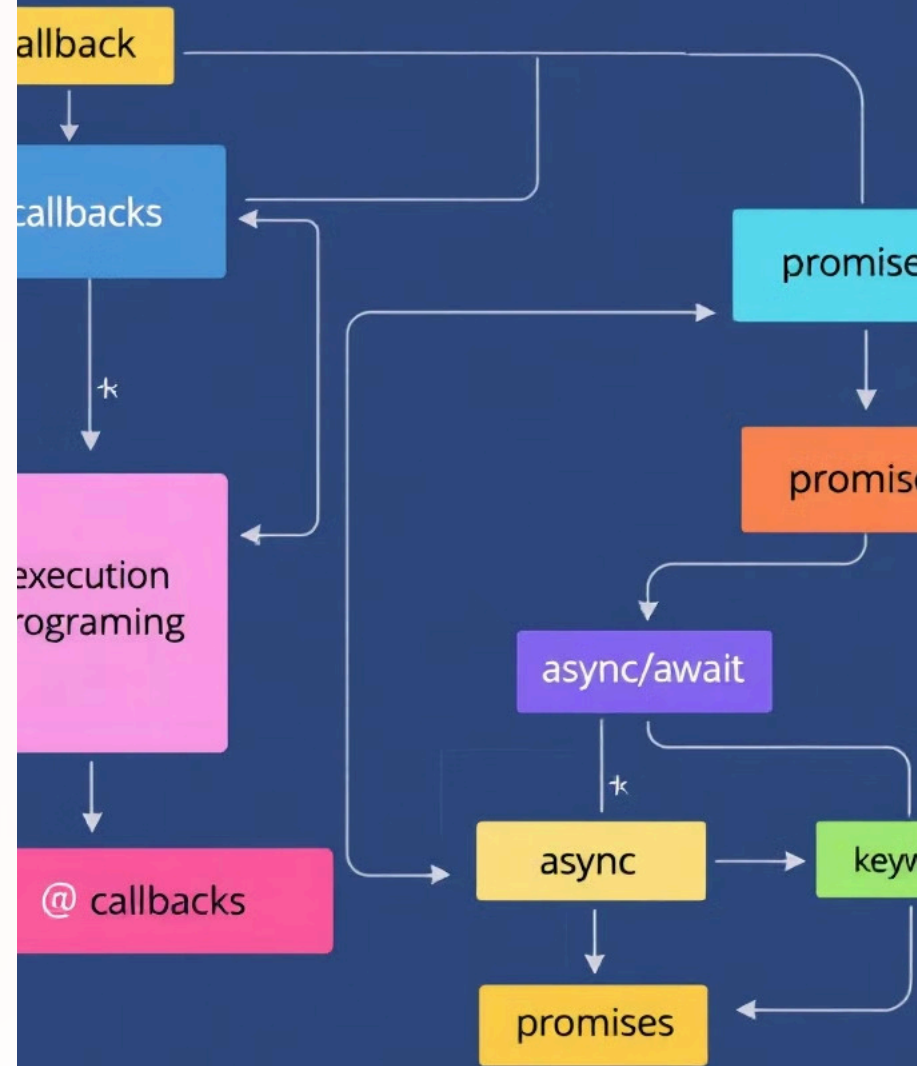
프로미스는 비동기 작업의 결과를 나타내는 객체입니다. 프로미스를 사용하면 비동기 코드를 보다 명확하고 관리하기 쉽게 작성할 수 있습니다.

3

Async/Await

Async/Await는 프로미스를 보다 쉽게 사용할 수 있도록 도와주는 기능입니다. 이를 통해 비동기 코드를 동기 코드처럼 작성할 수 있습니다.

asynchronous programming



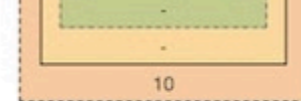


Get more done with the new Chrome

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



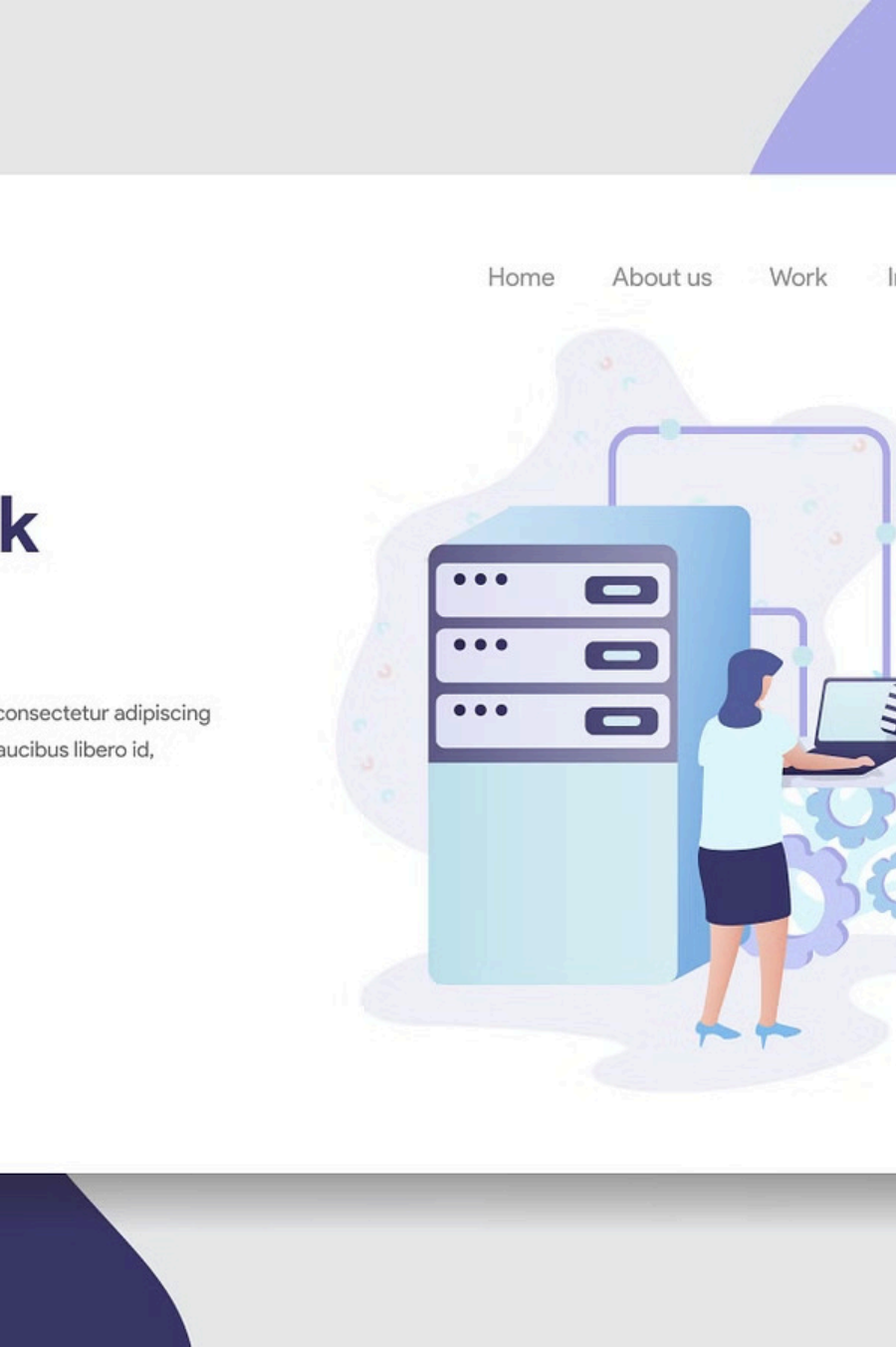
데이터베이스

데이터 저장 및 관리



API 개발

다른 애플리케이션과의 통신



Node.js를 이용한 웹 개발

1

웹 서버 구축

Node.js는 HTTP 모듈을 사용하여 웹 서버를 구축합니다. 웹 서버는 웹 애플리케이션을 호스팅하고 사용자의 요청을 처리합니다.

2

라우팅

라우팅은 사용자의 요청을 적절한 컨트롤러로 매핑하는 역할을 합니다. Node.js는 Express와 같은 라우팅 프레임워크를 제공합니다.

3

템플릿 엔진

템플릿 엔진은 동적인 웹 페이지를 생성하는 데 도움이 됩니다. Node.js는 Pug와 같은 다양한 템플릿 엔진을 지원합니다.

Node.js 활용 사례 및 미래

1

실시간 웹 애플리케이션

채팅, 게임, 스트리밍 서비스와 같은 실시간 웹 애플리케이션 개발에 사용됩니다.

2

API 개발

Node.js는 RESTful API를 개발하는 데 널리 사용됩니다. Node.js를 사용하면 빠르고 효율적인 API를 구축할 수 있습니다.

3

데이터 스트리밍

데이터 스트리밍은 실시간으로 데이터를 전송하는 기술입니다. Node.js는 데이터 스트리밍 애플리케이션을 구축하는 데 적합합니다.



Thanks!

Do you have any questions?

youremail@email.com

+91 620 421 838

www.yourwebsite.com

@yourusername

