IT 기술실무
Node.js 2주차

목표 : Node.js에 대한 개념을 이해할 수 있다.

강의개요

주	주차별 목표	강의소개	교재
1	강의소개 등	강의소개 및 필수사항 공유	주/보조
2	Abstract	핵심 개념, 서버로서의 노드, 서버 외의 노드, 개발 환경 구축(23~64)	주교재
3	ES2015+ : ECMAScript 6	ES2015+ 및 프런트엔드 자바스크립트(65~92)	주교재
4	Node's Functions	REPL, JS 파일 실행하기, 모듈로 만들기, 노드 내장 객체/모듈 등(93~178)	주교재
5	http Module	요청/응답, REST와 라우팅, 쿠키/세션, https/http2, cluster(179~216)	주교재
6	Package Manager	npm, package.json, 패키지 배포하기(217~240)	주교재
7	Express Web Server	미들웨어, req/res, 템플릿 엔진 사용하기(241~290)	주교재
8	중간시험	중간시험 및 웹어플리케이션 추가 강의	주/보조
9	Databases : MySQL	MySQL 설치, 데이터베이스/테이블, CRUD, 시퀄라이즈(291~364)	주교재
10	Databases : MongoDB	몽고디비 설치, 데이터베이스/컬렉션, CRUD, 몽구스(365~416)	주교재
11	Web API Server	API 서버, JWT 토큰 인증하기, 다른 서비스 호출하기, SNS API, CORS 등(475~520)	주교재
12	Node Service Test	테스트 준비하기, 유닛/커버리지/통합/부하(521~560)	주교재
13	Web Socket	웹 소켓 이해하기, ws 모듈, Socket.io, 미들웨어와 소켓, 채팅 구현하기(561~608)	주교재
14	CLI	간단한 콘솔 명령어, Commander, Inquirer 사용하기(649~676)	주교재
15	기말시험	기말시험 및 웹개발자 로드맵 추가 강의	주/보조

목차

table of contents

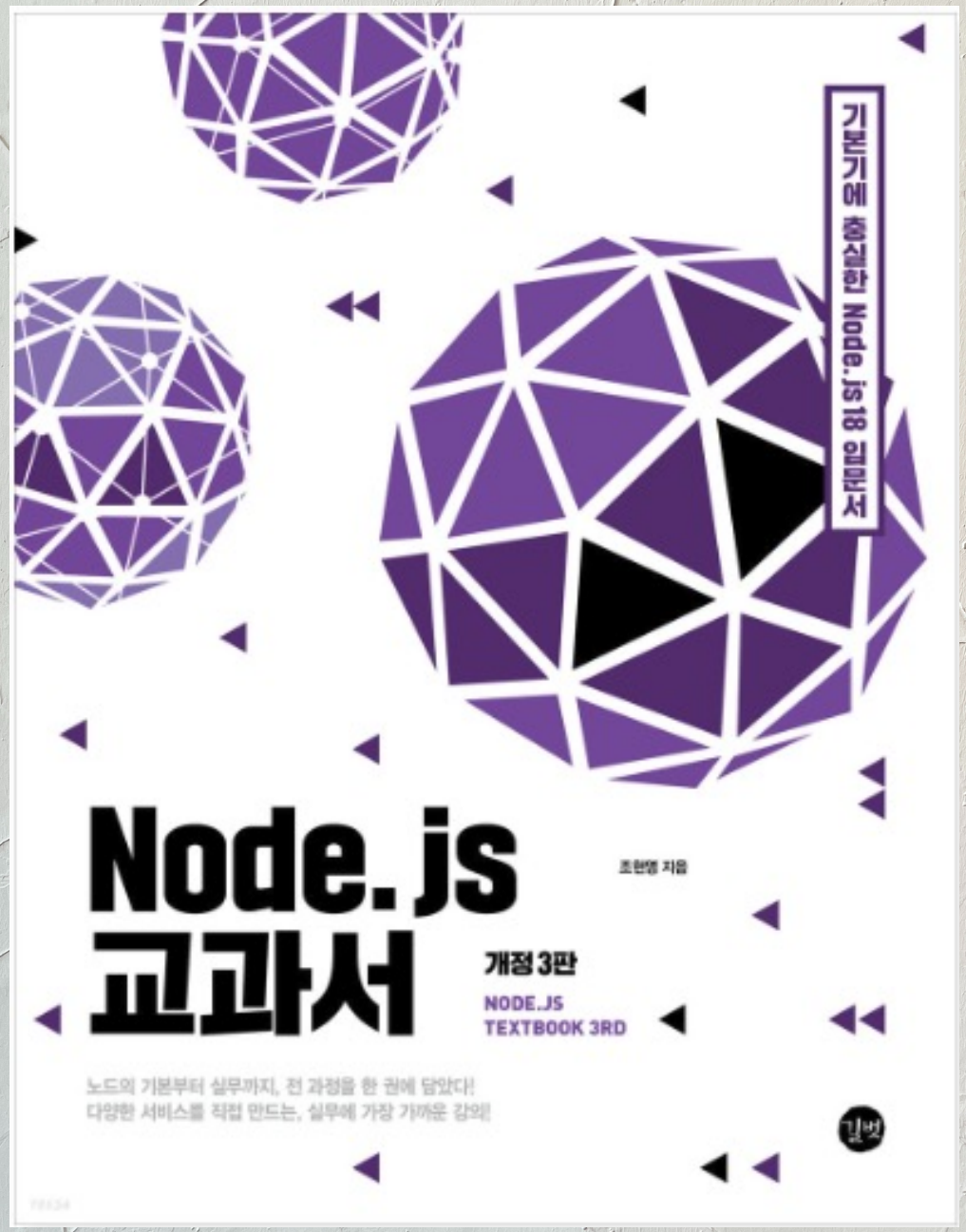
1 핵심 개념 이해하기

2 노드의 역할

서버로서의 노드

서버 외의 노드

4 개발 환경 설정하기



1.1 핵심 개념 이해하기

핵심 개념 이해하기

개 요

노드가 무엇인지에 대해 여러 가지 의견이 많지만, 어떠한 설명도 노드 공식 사이트의 설명보다 정확하지는 않을 것입니다. 노드 공식 사이트(https://nodejs.org/ko/)에서는 노드를 다음과 같이 설명함

Node.js® 는 Chrome V8 Javascript 엔진으로 빌드된 자바스크립트 런타임입니다.

- 대부분은 노드를 서버로 사용하는 방법을 익히기 위해 Node.js를 공부할 것임
- 그런데 공식 사이트의 노드 소개 글에는 서버라는 말이 없음
- 서버라는 말이 없는 이유는 노드가 서버만 실행할 수 있는 것이 아니기 때문

교재에서도 전반적으로 노드로 서버를 실행하는 방법을 다루지만, 서버 외의 자바스크립트 프로그램을 실행하는 런타임으로 사용하는 방법도 배우게 됨

핵심 개념 이해하기

1.1.1. 서버

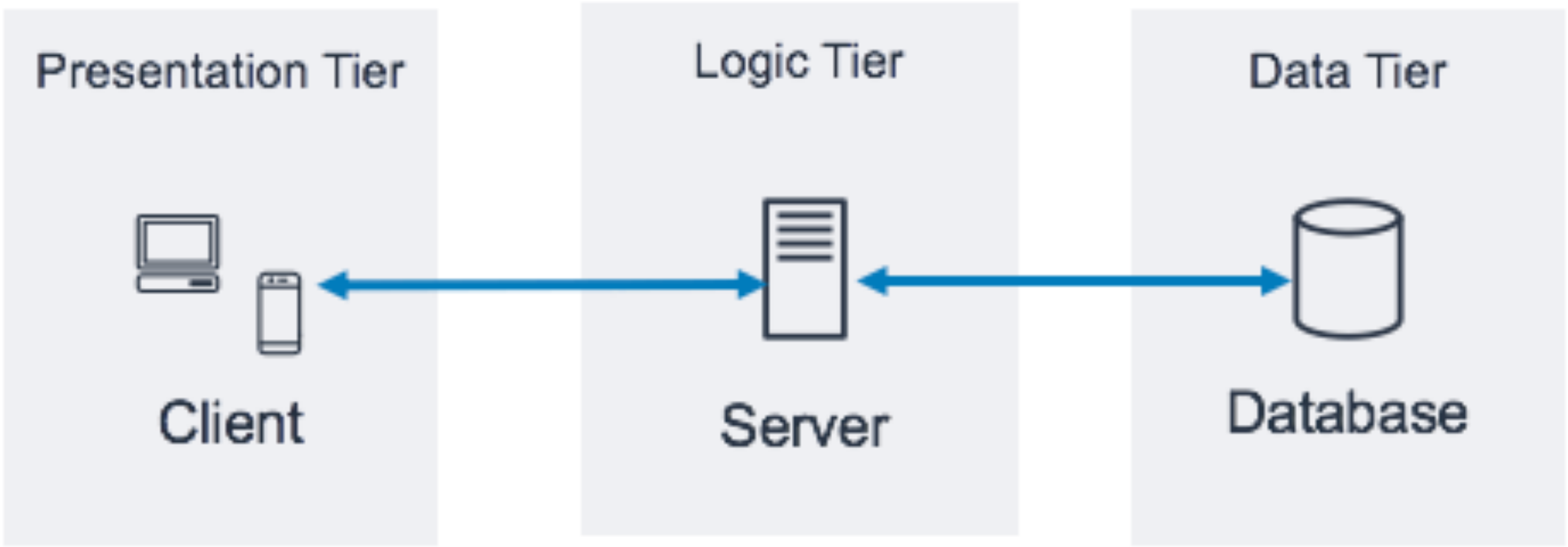
노드를 통해 다양한 자바스크립트 애플리케이션을 실행할 수 있지만, 노드는 서버 애플리케이션을 실행하는 데 제일 많이 사용함

그럼 서버란 무엇이며, 어떤 역할을 할까요?

- 서버는 네트워크를 통해 클라이언트에 정보나 서비스를 제공하는 컴퓨터 또는 프로그램임
- 클라이언트란 요청을 보내는 주체로 브라우저일 수도 있고,데스크톱 프로그램일 수도 있고, 모바일 앱일 수도 있고, 다른 서버에 요청을 보내는 서버일 수도 있음

평소에 사용하는 웹 사이트가 앱을 생각해보면, 웹사이트의 화면(HTML)은 어디에서 가져올까요? 앱 설치 파일은 어디에서 내려받는 것일까요?

핵심 개념 이해하기



참고 : https://hanamon.kr/네트워크-http와-클라이언트-서버-아키텍처/

핵심 개념 이해하기

1.1.1. 서버

정리하면,

- 서버는 클라이언트의 요청에 대해 응답을 함
- 응답으로 항상 Yes를 해야 하는 것은 아니고, No를 할 수도 있음
- 어떤 사이트로부터 접속을 차단당했다면 그 사이트의 서버는 여러분의 요청에 매번 No를 응답할 것임

노드는 자바스크립트 프로그램이 서버로서 기능하기 위한 도구를 제공하므로 서버 역할을 수행할 수 있음

왜 다른 언어를 사용하지 않고 굳이 노드를 사용해 서버를 만들까요?

핵심 개념 이해하기

1.1.2. 자바스크립트 런타임

Node.js® 는 Chrome V8 Javascript 엔진으로 빌드된 자바스크립트 런타임입니다.

- 런타임: 특정 언어로 만든 프로그램들을 실행할 수 있게 해주는 가상 머신(크롬의 V8 엔진 사용)의 상태
- ∴ 노드: 자바스크립트로 만든 프로그램들을 실행할 수 있게 해 줌
- 다른 런타임으로는 웹 브라우저(크롬, 엣지, 사파리, 파이어폭스 등)가 있음
- 노드 이전에도 자바스크립트 런타임을 만들기 위한 많은 시도
- But, 엔진 속도 문제로 실패

핵심 개념 이해하기

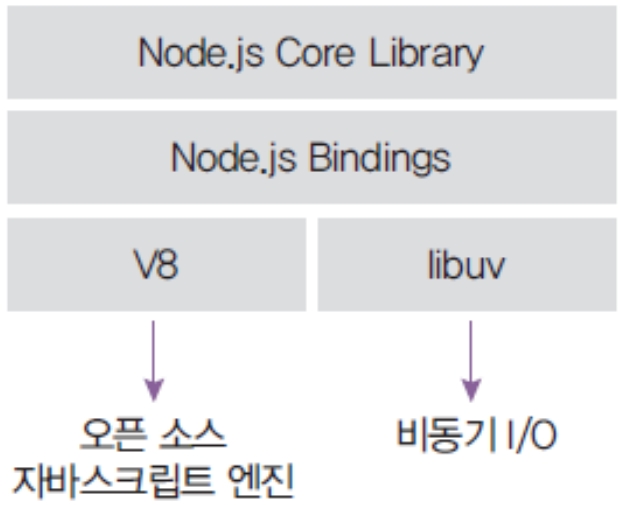
1.1.2. 자바스크립트 런타임

2008년 V8 엔진 출시, 2009년 노드 프로젝트 시작

노드는 V8과 libuv를 내부적으로 포함

- V8 엔진: 오픈 소스 자바스크립트 엔진]
  -> 속도 문제 개선
- libuv: 노드의 특성인 이벤트 기반,
  논블로킹 I/O 모델을 구현한 라이브러리

그림 1-3 노드의 내부 구조

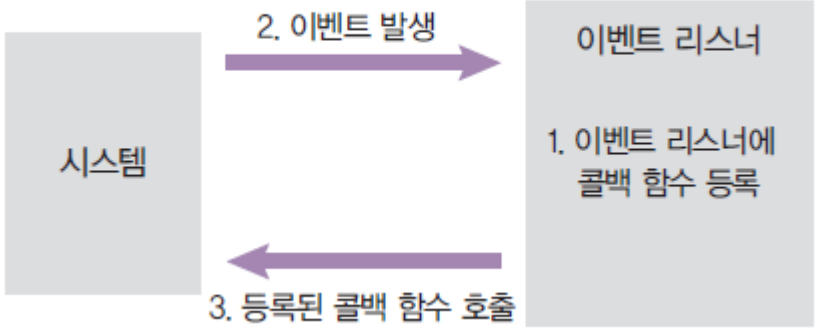


1.1.3. 이벤트 기반

이벤트가 발생할 때 미리 지정해둔 작업을 수행하는 방식

- 이벤트의 예: 클릭, 네트워크 요청, 타이머 등
- 이벤트 리스너: 이벤트를 등록하는 함수
- 콜백 함수: 이벤트가 발생했을 때 실행될 함수

그림 1-4 이벤트 기반



핵심 개념 이해하기

1.1.3. 이벤트 기반

이벤트 기반 모델에서는 이벤트 루프(event loop)라는 개념이 등장

- 여러 이벤트가 동시에 발생했을 때 어떤 순서로 콜백 함수를 호출할지를 이벤트 루프가 판단함

노드는 자바스크립트 코드의 맨 위부터 한 줄씩 실행함

- 함수 호출 부분을 발견했다면 호출한 함수를 호출 스택(call stack)에 넣고, 다음 코드가 콘솔(브라우저 콘솔을 사용하면 됨)에 어떤 로그를 남길지 실습

핵심 개념 이해하기

1.1.3. 이벤트 기반

first 함수가 제일 먼저 호출되고, 그 안의 second 함수가 호출된 뒤, 마지막으로 third 함수가 호출됨

- 호출된 순서와는 반대로 실행이 완료
- 따라서 콘솔에는 세 번째, 두 번째, 첫 번째 순으로 출력하게 됨

```
function first() {
  second();
  console.log( data[0]: '첫 번째');
}
function second() {
  third();
  console.log( data[0]: '두 번째');
}
function third() {
  console.log( data[0]: '세 번째');
}
first();
```

핵심 개념 이해하기

1.1.3. 이벤트 기반

특정 시간(밀리초, 1,000분의 1초) 이후에 코드를 실행하는 setTimeout을 사용한 예제로 실습

```
1   function run() {
2     console.log( data[0]: '3초 후 실행');
3   }
4   console.log( data[0]: '시작');
5   setTimeout( handler: run,  timeout: 3000);
6   console.log( data[0]: '끝');
```

3초 뒤에 run 함수를 실행하는 코드임

- setTimeout 함수의 콜백인 run이 호출 스택에 언제 들어가는지는 이벤트 루프, 태스크 큐(task queue), 백그라운드(background)를 알아야 이해할 수 있음

핵심 개념 이해하기

1.1.3. 이벤트 기반

- 이벤트 루프 : 이벤트 발생 시 호출할 콜백 함수들을 관리하고, 호출된 콜백 함수의 실행 순서를 결정하는 역할을 담당
  노드가 종료될 때까지 이벤트 처리를 위한 작업을 반복하므로 루프(loop)라고 부름
- 백그라운드 : setTimeout 같은 타이머나 이벤트 리스너들이 대기하는 장소
  자바스크립트가 아닌 다른 언어로 작성된 프로그램이라고 봐도 무방하며, 여러 작업이 동시에 실행될 수 있음
- 태스크 큐 : 이벤트 발생 후, 백그라운드에서는 태스크 큐로 타이머나 이벤트 리스너의 콜백 함수를 보냄
  정해진 순서대로 콜백들이 줄을 서 있으므로 콜백 큐라고도 함
  콜백들은 보통 완료된 순서대로 줄을 서 있지마, 특정한 경우 순서가 바뀌기도 함

핵심 개념 이해하기

1.1.3. 이벤트 기반

그림 1-6 이벤트 루프 1

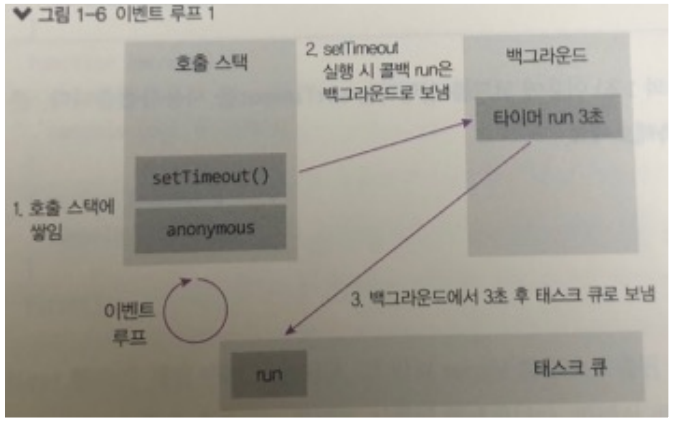


그림 1-7 이벤트 루프 2

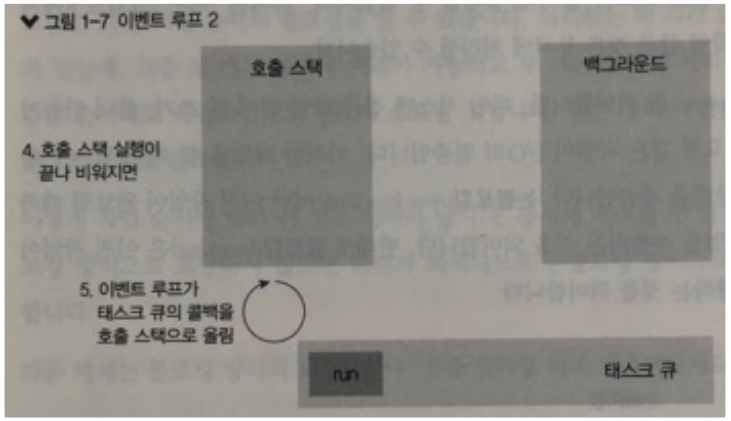
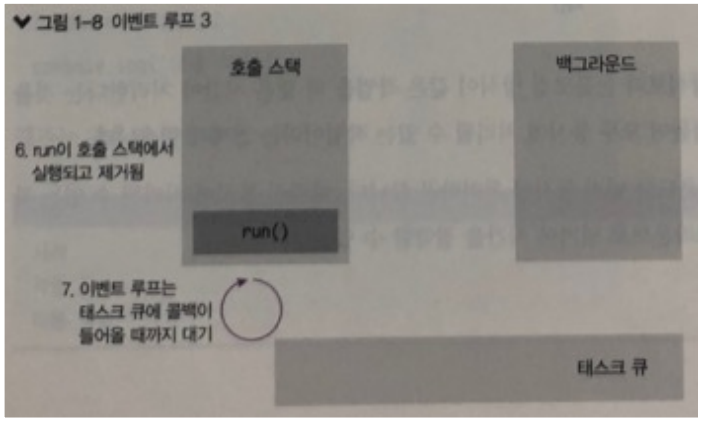


그림 1-8 이벤트 루프 3



핵심 개념 이해하기

1.1.4. 논블로킹 I/O

논블로킹: 오래 걸리는 함수를 백그라운드로 보내서 다음 코드가 먼저 실행되게 하고, 나중에 오래 걸리는 함수를 실행

- 논 블로킹 방식 하에서 일부 코드는 백그라운드에서 병렬로 실행됨
- 일부 코드: I/O 작업(파일 시스템 접근, 네트워크 요청), 압축, 암호화 등
- 나머지 코드는 블로킹 방식으로 실행됨
- ∴ I/O 작업이 많을 때 노드 활용성이 극대화

그림 1-9 블로킹과 논블로킹

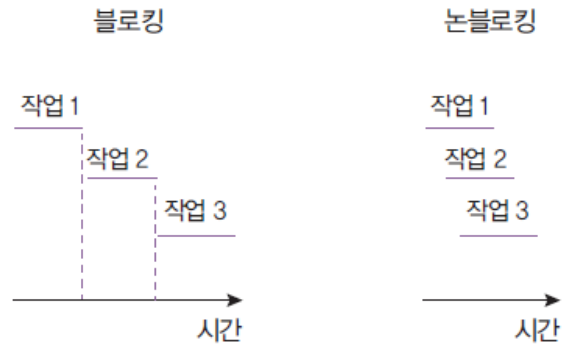
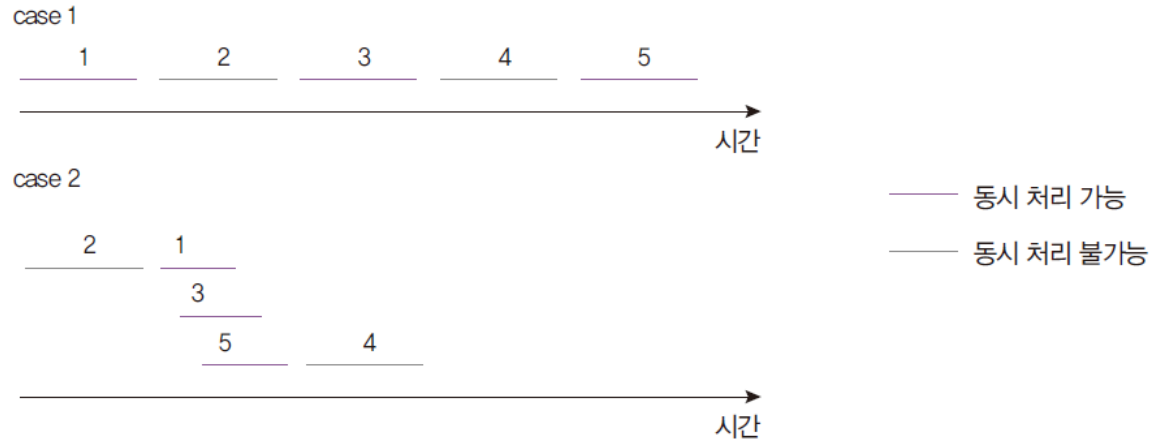


그림 1-10 동시 처리로 얻는 시간적 이득



핵심 개념 이해하기

1.1.4. 논블로킹 I/O

블로킹 방식의 코드 예시로 실습

```
function longRunningTask() {
  // 오래 걸리는 작업
  console.log( data[0]: '작업 끝');
}
console.log( data[0]: '시작');
longRunningTask();
console.log( data[0]: '다음 작업');
```

핵심 개념 이해하기

1.1.4. 논블로킹 I/O

논블로킹 방식의 코드 예시로 실습 : setTimeout(콜백, 0)은 코드를 논블로킹으로 만들기 위해 사용하는 기법 중 하나임

```
1   function longRunningTask() {
2     // 오래 걸리는 작업
3     console.log( data[0]: '작업 끝');
4   }
5   console.log( data[0]: '시작');
6   setTimeout( handler: longRunningTask,  timeout: 0);
7   console.log( data[0]: '다음 작업');
```

* 논블로킹과 동시가 같은 의미는 아니며, 동시성은 동시 처리가 가능한 작업을 논블로킹 처리해야 얻을 수 있음

핵심 개념 이해하기

1.1.5. 싱글 스레드

프로세스와 스레드

- 프로세스: 운영체제에서 할당하는 작업의 단위, 프로세스 간 자원 공유x
- 스레드: 프로세스 내에서 실행되는 작업의 단위, 부모 프로세스 자원 공유

노드 프로세스는 멀티 스레드이지만 직접 다룰 수 있는 스레드는 하나이기 때문에 싱글 스레드라고 표현

노드는 주로 멀티 스레드 대신 멀티 프로세스 활용

노드는 14버전부터 멀티 스레드 사용 가능

핵심 개념 이해하기

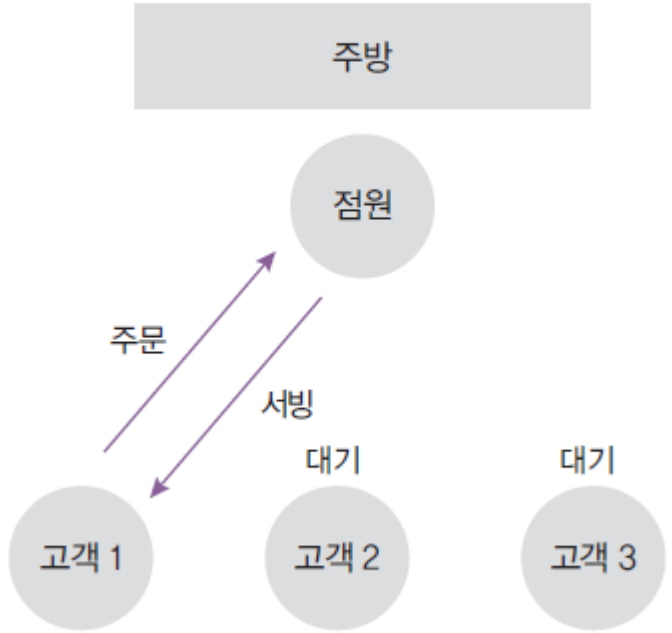
1.1.5. 싱글 스레드

싱글 스레드라 주어진 일을 하나밖에 처리하지 못함

- 블로킹이 발생하는 경우 나머지 작업은 모두 대기해야 함 -> 비효율 발생

주방에 비유(점원: 스레드, 주문: 요청, 서빙: 응답)

그림 1-10 싱글 스레드, 블로킹 모델

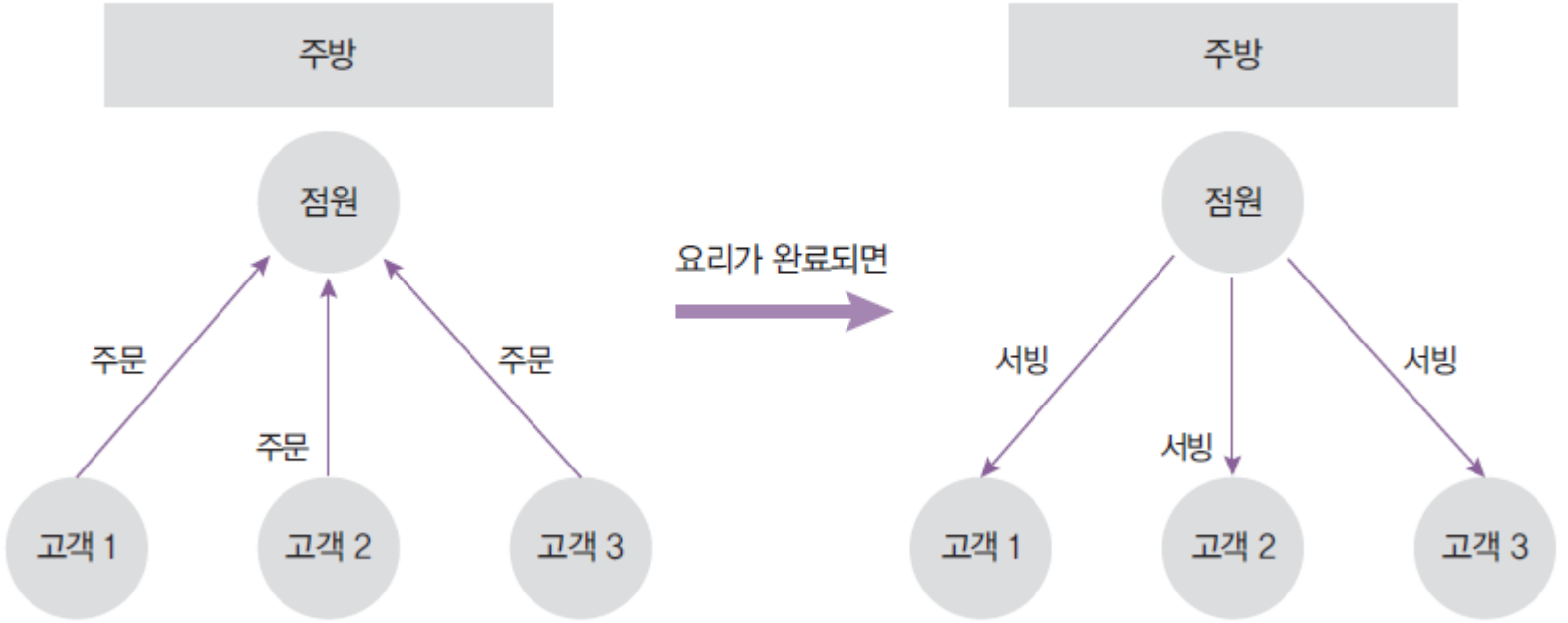


1.1.5. 싱글 스레드

대신 논 블로킹 모델을 채택하여 일부 코드(I/O)를 백그라운드(다른 프로세스)에서 실행 가능

- 요청을 먼저 받고, 완료될 때 응답함
- I/O 관련 코드가 아닌 경우 싱글 스레드, 블로킹 모델과 같아짐

그림 1-11 싱글 스레드, 논블로킹 모델



핵심 개념 이해하기

1.1.5. 싱글 스레드

싱글 스레드 모델은 에러를 처리하지 못하는 경우 멈춤

- 프로그래밍 난이도 쉽고, CPU, 메모리 자원 적게 사용

멀티 스레드 모델은 에러 발생 시 새로운 스레드를 생성하여 극복

- 단, 새로운 스레드 생성이나 놀고 있는 스레드 처리에 비용 발생
- 프로그래밍 난이도 어려움
- 스레드 수만큼 자원을 많이 사용함

점원: 스레드, 주문: 요청, 서빙: 응답

그림 1-12 멀티 스레드, 블로킹 모델

1.1.5. 싱글 스레드

노드 14버전 이후

- 멀티 스레드를 사용할 수 있도록 worker_threads 모듈 도입
- CPU를 많이 사용하는 작업인 경우에 활용 가능
- 멀티 프로세싱만 가능했던 아쉬움을 달래줌

표 1-1 멀티 스레딩과 멀티 프로세싱 비교

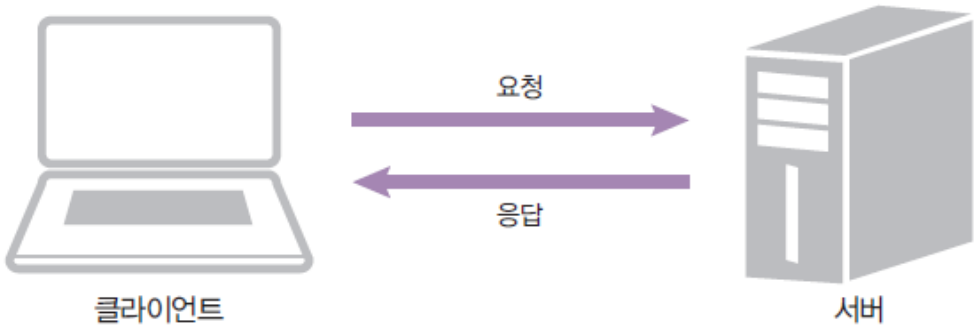
멀티 스레딩	멀티 프로세싱
하나의 프로세스 안에서 여러 개의 스레드 사용	여러 개의 프로세스 사용
CPU 작업이 많을 때 사용	I/O 요청이 많을 때 사용
프로그래밍이 어려움	프로그래밍이 비교적 쉬움

1.3 노드의 역할

1. 서버로서의 노드

- 서버: 네트워크를 통해 클라이언트에 정보나 서비스를 제공하는 컴퓨터 또는 프로그램
- 클라이언트: 서버에 요청을 보내는 주체(브라우저, 데스크탑 프로그램, 모바일 앱, 다른 서버에 요청을 보내는 서버)
- 예시
  - 브라우저(클라이언트, 요청)가 길벗 웹사이트(서버, 응답)에 접속
  - 핸드폰(클라이언트)을 통해 앱스토어(서버)에서 앱 다운로드
- 노드 != 서버
- But, 노드는 서버를 구성할 수 있게 하는 모듈(4장에서 설명)을 제공

그림 1-2 클라이언트와 서버



2. 서버로서의 노드

노드 서버의 장단점

표 1-1 노드의 장단점

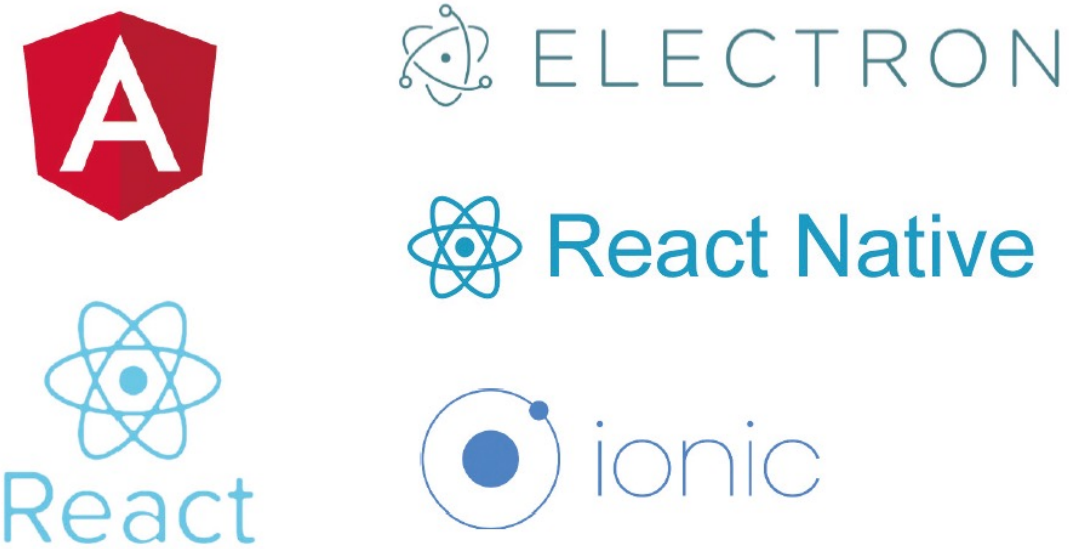
장점	단점
멀티 스레드 방식에 비해 컴퓨터 자원을 적게 사용함	싱글 스레드라서 CPU 코어를 하나만 사용함
I/O 작업이 많은 서버로 적합	CPU 작업이 많은 서버로는 부적합
멀티 스레드 방식보다 쉬움	하나뿐인 스레드가 멈추지 않도록 관리해야 함
웹 서버가 내장되어 있음	서버 규모가 커졌을 때 서버를 관리하기 어려움
자바스크립트를 사용함	어중간한 성능
JSON 형식과 호환하기 쉬움	

- CPU 작업을 위해 AWS Lambda나 Google Cloud Functions같은 별도 서비스 사용
- 페이팔, 넷플릭스, 나사, 월마트, 링크드인, 우버 등에서 메인 또는 서브 서버로 사용

3. 서버 외의 노드

- 자바스크립트 런타임이기 때문에 용도가 서버에만 한정되지 않음
- 웹, 모바일, 데스크탑 애플리케이션에도 사용
  - 웹 프레임워크: Angular, React, Vue, Meteor 등
  - 모바일 앱 프레임워크: React Native
  - 데스크탑 개발 도구: Electron(Atom, Slack, VSCode, Discord 등 제작)
- 위 프레임워크가 노드 기반으로 동작함

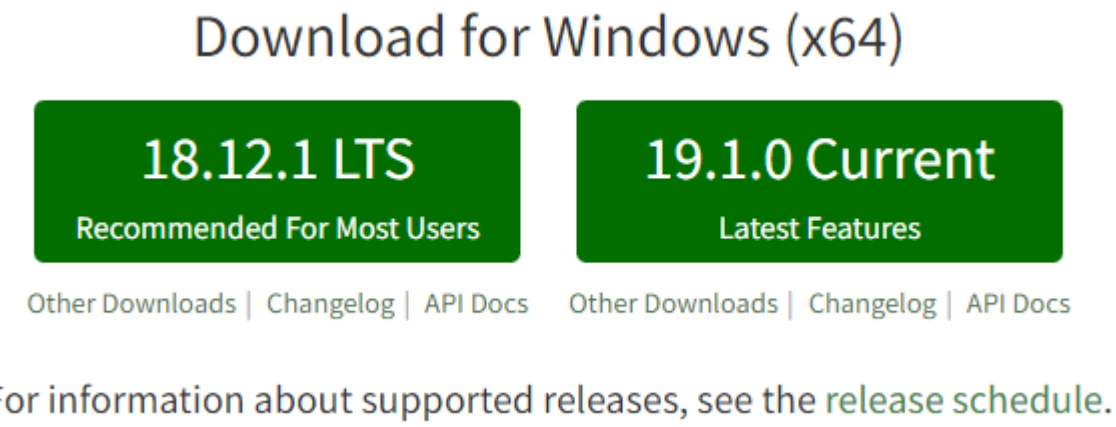
그림 1-16 노드 기반의 개발 도구



1.4 개발 환경 설정하기

1. 노드 설치하기

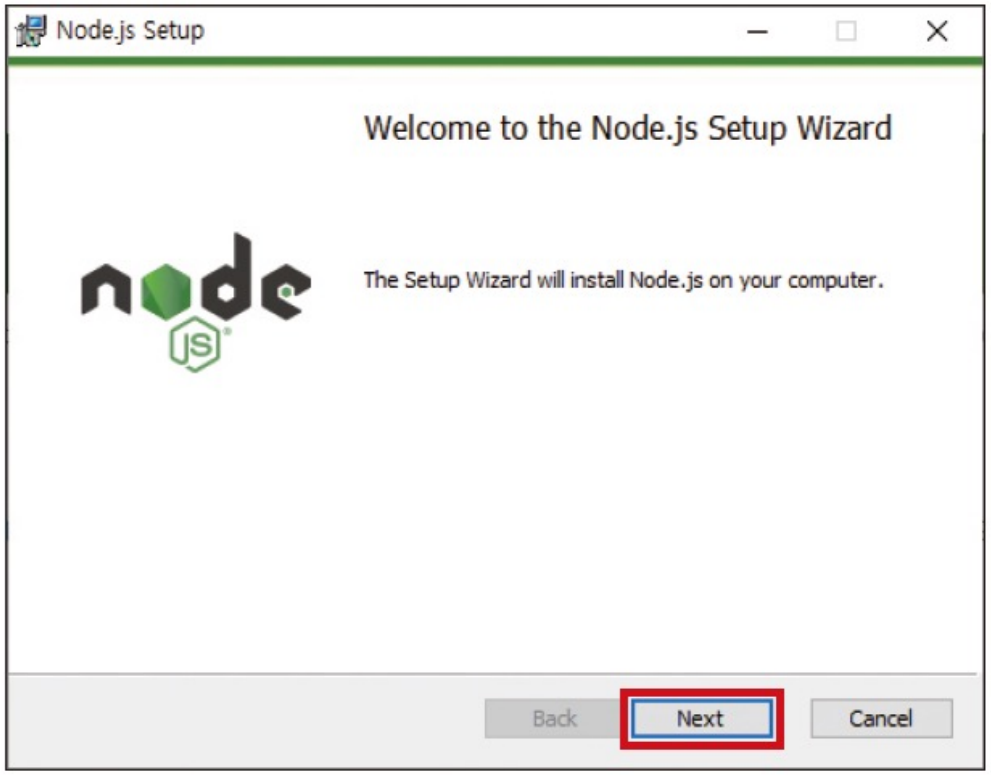
- 윈도(11 기준), 맥(벤투라 기준)
  - https://nodejs.org 접속
  - LTS 버전인 18버전 설치
  - LTS는 안정된 버전, Current는 최신 버전(실험적)



1. 노드 설치하기

» 계속 Next 버튼을 눌러 설치

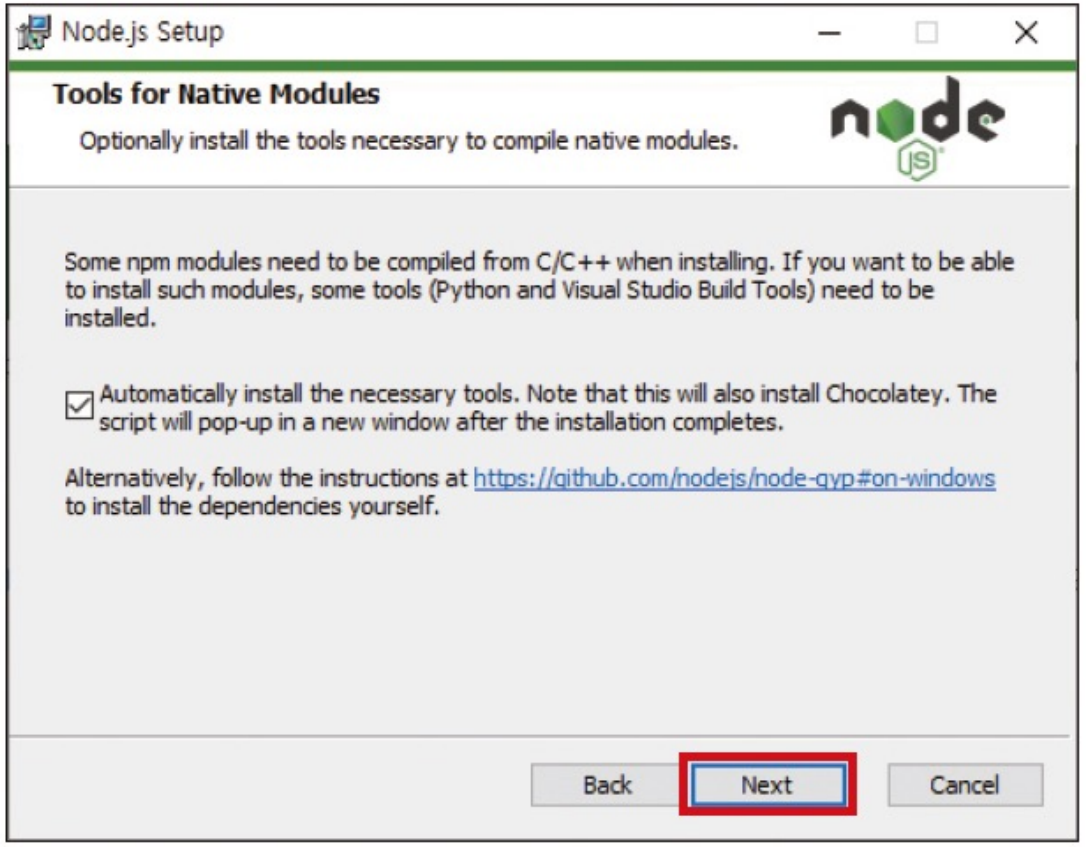
그림 1-19 Setup Wizard 실행



1. 노드 설치하기

» 필요 도구 반드시 설치할 것

그림 1-23 필요 도구 설치



1. 노드 설치하기

- 리눅스(우분투 20 LTS 기준)
  - 터미널에 아래 코드 입력

콘솔

```
$ sudo apt-get update
$ sudo apt-get install -y build-essential
$ sudo apt-get install -y curl
$ curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash --
$ sudo apt-get install -y nodejs
```

1. 노드 설치하기

» 설치 완료 후 윈도, 맥, 리눅스 모두 명령 프롬프트나 터미널 실행 후 다음 명령어 입력

콘솔

```
$ node -v
18.7.0
$ npm -v
8.15.0
```

» 버전은 다를 [illegible]공

- npm 버전을 업데이트 하려면 다음 명령어 입력
- 맥과 리눅스는 명령어 앞에 sudo 필요

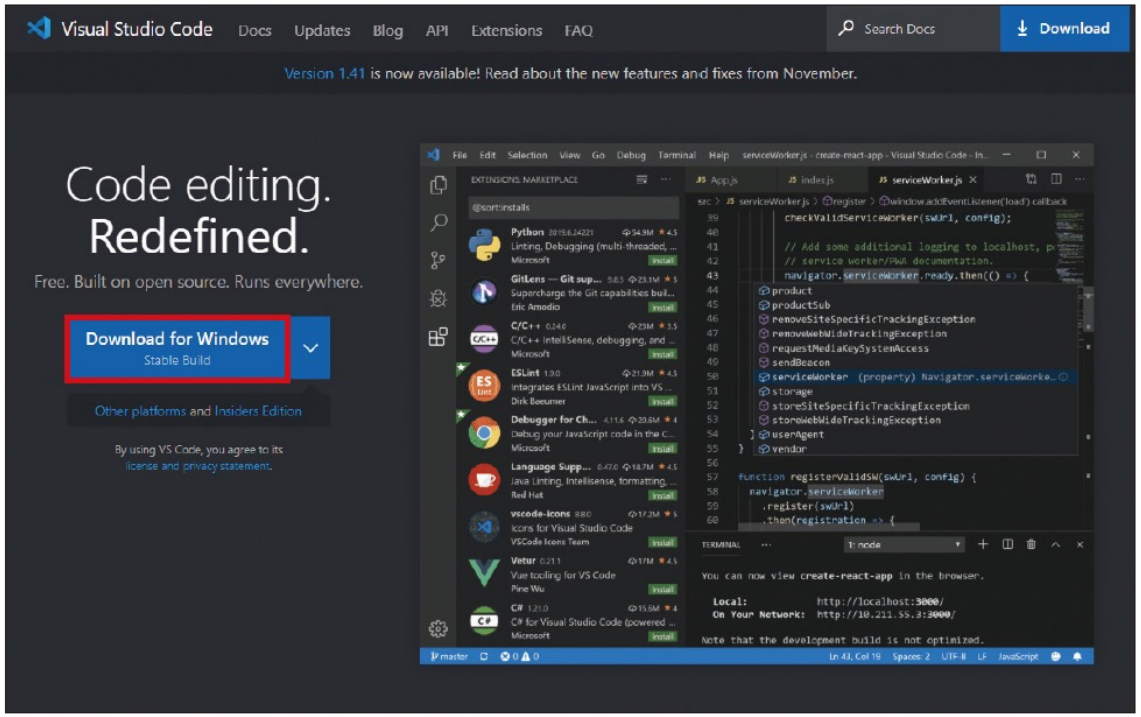
콘솔

```
$ npm install -g npm
```

2. VS Code 설치하기

- VS Code: 마이크로소프트에서 제공하는 오픈 소스 코드 에디터
  - 자바스크립트, 노드에 대한 지원이 탁월함
  - 윈도, 맥, 리눅스(GUI) 모두 https://code.visualstudio.com 접속
  - 운영체제에 맞는 파일 설치
  - VS Code 외의 다른 코드 에디터를 사용해도 됨

그림 1-41 VS Code의 공식 사이트



감사합니다.

단순하게 설명할 수 없다면
제대로 이해하지 못한 것이다.

아인슈타인