컴퓨터 네트워크

신나는 수행평가 안내

- [논술형] 네트워크 이해하기
  - 만점 30점 / 기본 6점
  - 네트워크 개념 및 자바 스크립트 문법 수행

- [실습형] 웹 서버 구축하기
  - 만점 35점 / 기본 11점
  - node로 서버 제작, 쿠키 및 세션, 라우팅
  - 익스프레스 활용 서버 제작, req/res 객체 활용

- [프로젝트] 웹서비스 제작하기
  - 만점 35점 / 기본 11점
  - 배운 내용 토대로 SNS 서비스 (혹은 포트폴리오 사이트) 제작
  - 개인 프로젝트로 진행

1장

1.1 노드의 정의

1. 노드의 정의

» 공식 홈페이지의 설명

- Node.js®는 크롬 V8 자바스크립트 엔진으로 빌드된 자바스크립트 런타임입니다.

» 노드는 서버가 아닌가요? 서버라는 말이 없네요.

- 서버의 역할도 수행할 수 있는 자바스크립트 런타임
- 노드로 자바스크립트로 작성된 서버를 실행할 수 있음.
- 서버 실행을 위해 필요한 http/https/http2 모듈을 제공

▲ 그림 1-2 클라이언트와 서버

2. 런타임

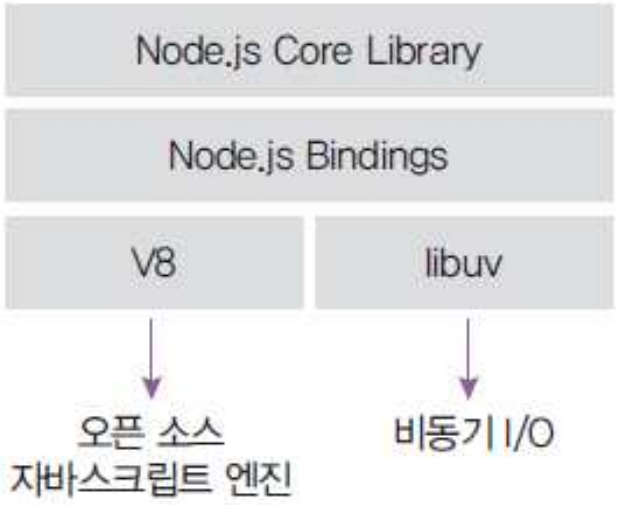
» 노드: 자바스크립트 런타임

- 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

3. 내부 구조

- 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



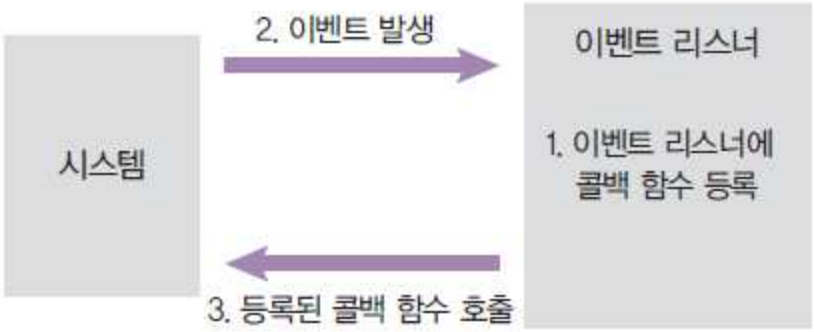
1.2 노드의 특성

1. 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

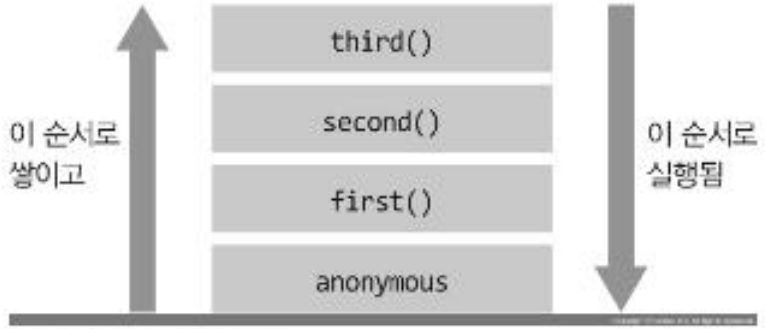


1. 이벤트 기반

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

- 어떤 순서로 콜백 함수를 호출할지 판단
- 함수 호출 발견 시, 호출 스택에 함수 삽입

```
function first() {
  second();
  console.log('첫 번째');
}
function second() {
  third();
  console.log('두 번째');
}
function third() {
  console.log('세 번째');
}
first();
```



▲ 그림 1-5 호출 스택

콘솔

```
세 번째
두 번째
첫 번째
```

1. 이벤트 기반

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

- 어떤 순서로 콜백 함수를 호출할지 판단
- 함수 호출 발견 시, 호출 스택에 함수 삽입

```
function run() {
  console.log('3초 후 실행');
}
console.log('시작');
setTimeout(run, 3000);

console.log('끝');
```

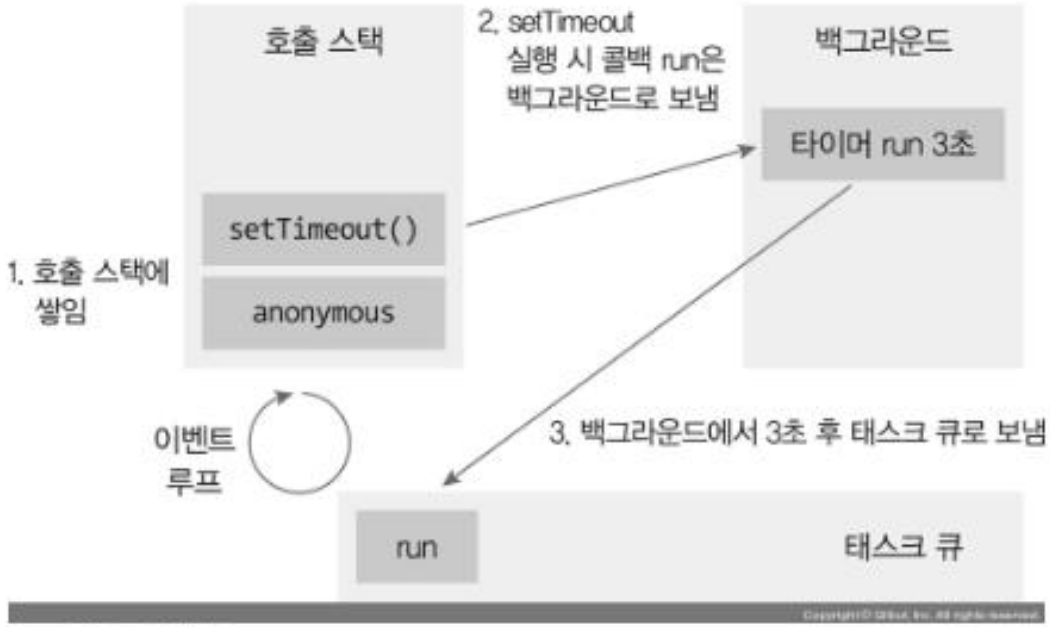
콘솔

```
시작
끝
3초 후 실행
```

1. 이벤트 기반

» 이벤트 루프, 백그라운드, 태스크 큐

- 이벤트 루프: 이벤트 발생 시 호출할 콜백 함수 관리, 실행 순서 결정
- 백그라운드: setTimeout 같은 타이머/이벤트 리스너가 대기하는 곳
- 태스크 큐: 백그라운드로부터 타이머/이벤트 리스너를 받는 곳

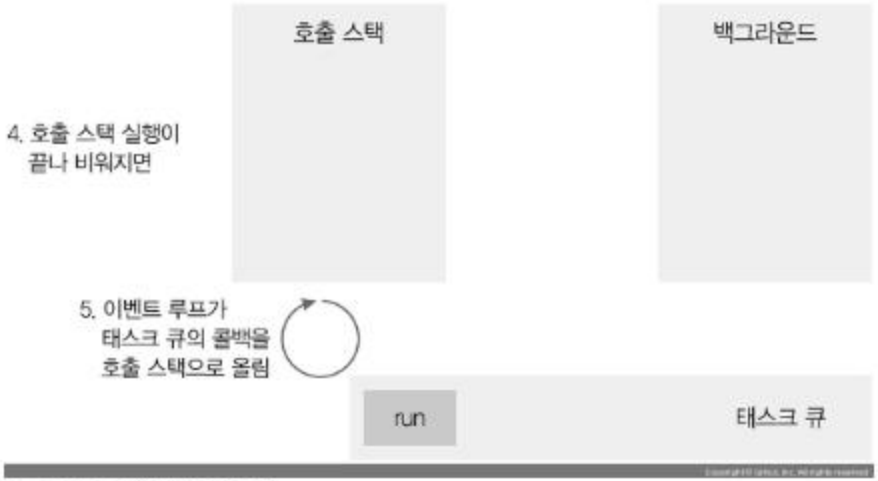


▲ 그림 1-6 이벤트 루프 1

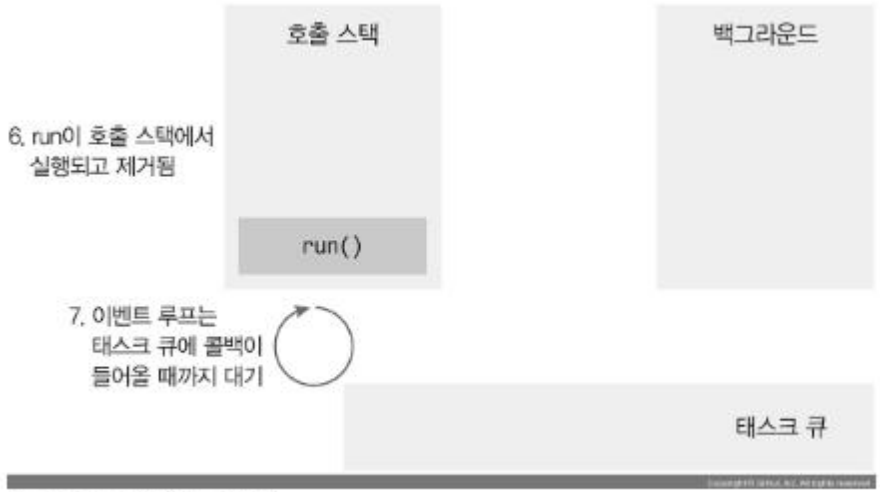
1. 이벤트 기반

» 이벤트 루프, 백그라운드, 태스크 큐

- 이벤트 루프: 이벤트 발생 시 호출할 콜백 함수 관리, 실행 순서 결정
- 백그라운드: setTimeout 같은 타이머/이벤트 리스너가 대기하는 곳
- 태스크 큐: 백그라운드로부터 타이머/이벤트 리스너를 받는 곳



▲ 그림 1-7 이벤트 루프 2



▲ 그림 1-8 이벤트 루프 3

2. 논블로킹 I/O

» 논 블로킹: 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

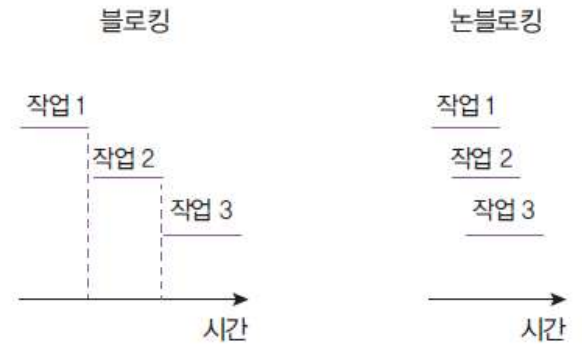
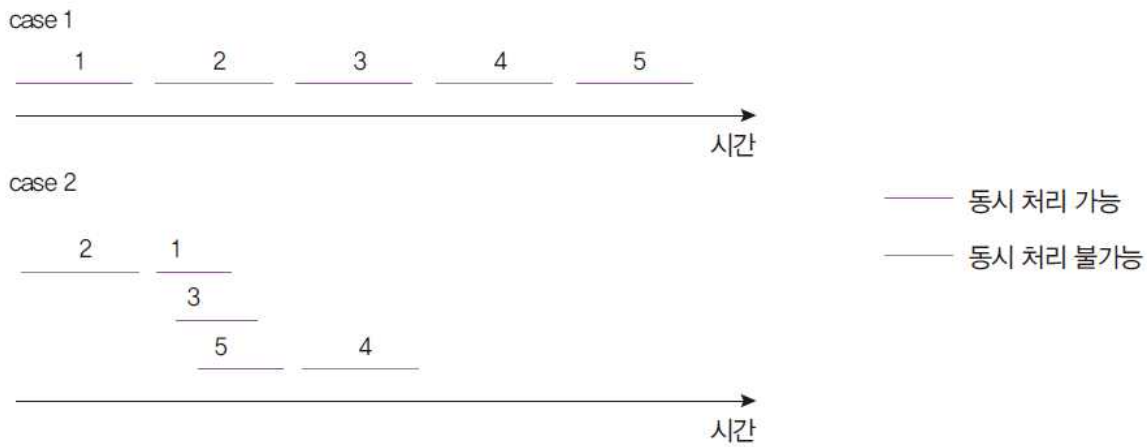


그림 1-10 동시 처리로 얻는 시간적 이득



2. 논블로킹 I/O

» 논 블로킹: 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

```
function longRunningTask() {
  // 오래 걸리는 작업
  console.log('작업 끝');
}

console.log('시작');
longRunningTask();
console.log('다음 작업');
```

콘솔

```
시작
작업 끝
다음 작업
```

2. 논블로킹 I/O

» 논 블로킹: 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

```
function longRunningTask() {
  // 오래 걸리는 작업
  console.log('작업 끝');
}
console.log('시작');
setTimeout(longRunningTask, 0);
console.log('다음 작업');
```

콘솔

```
시작
다음 작업
작업 끝
```

3. 프로세스 vs 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

그림 1-13 스레드와 프로세스

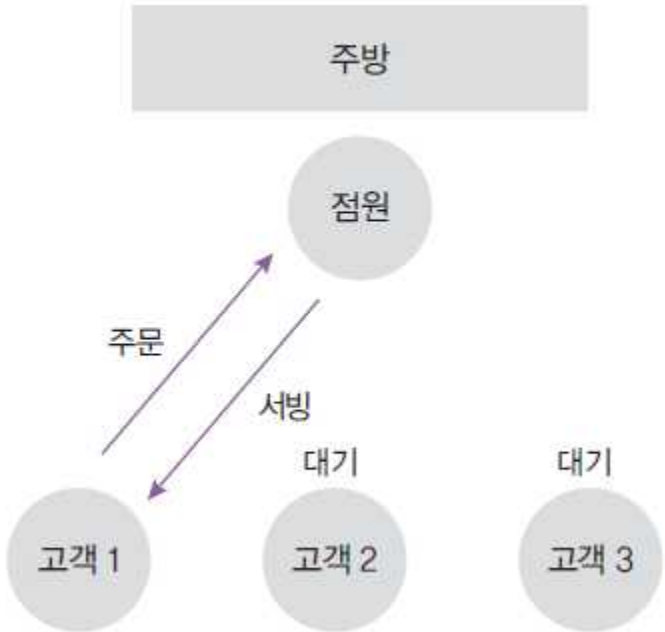
4. 싱글 스레드

» 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

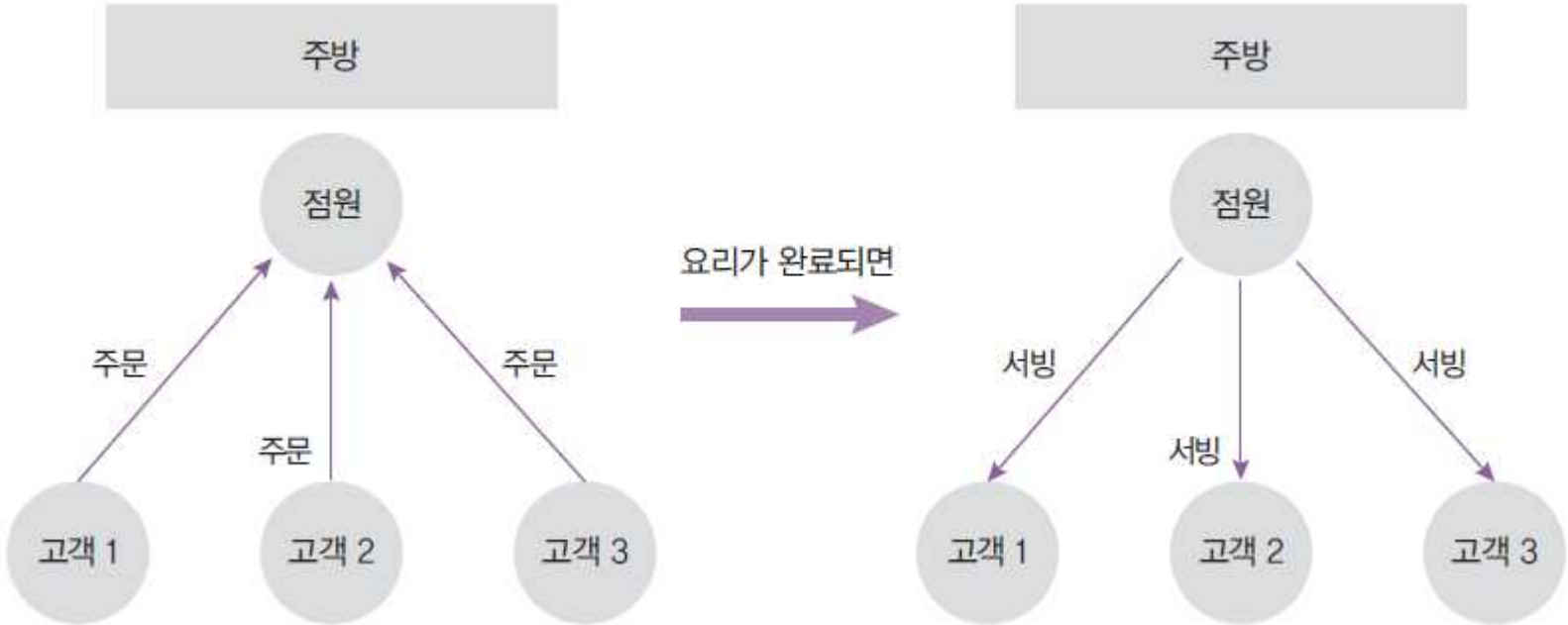


4. 싱글 스레드

» 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



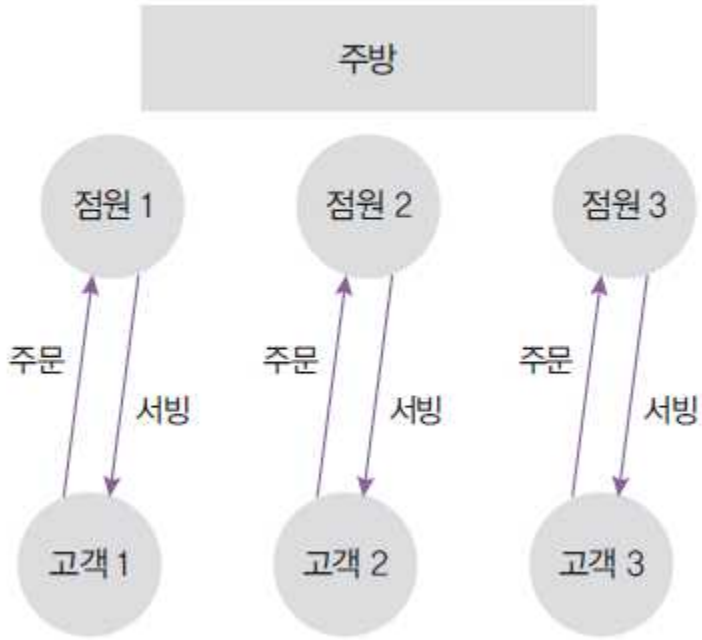
5. 멀티 스레드 모델과의 비교

» 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



6. 멀티 스레드의 활용

노드 14버전

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

표 1-1 멀티 스레딩과 멀티 프로세싱 비교

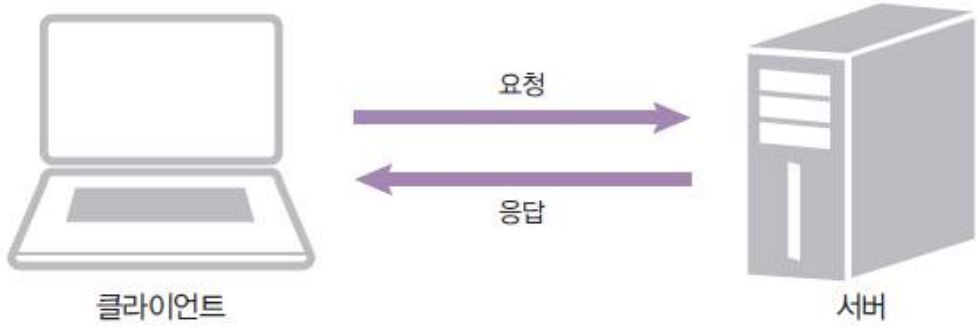
멀티 스레딩	멀티 프로세싱
하나의 프로세스 안에서 여러 개의 스레드 사용	여러 개의 프로세스 사용
CPU 작업이 많을 때 사용	I/O 요청이 많을 때 사용
프로그래밍이 어려움	프로그래밍이 비교적 쉬움

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

1.3 개발 환경 설정하기

1. 노드 설치하기

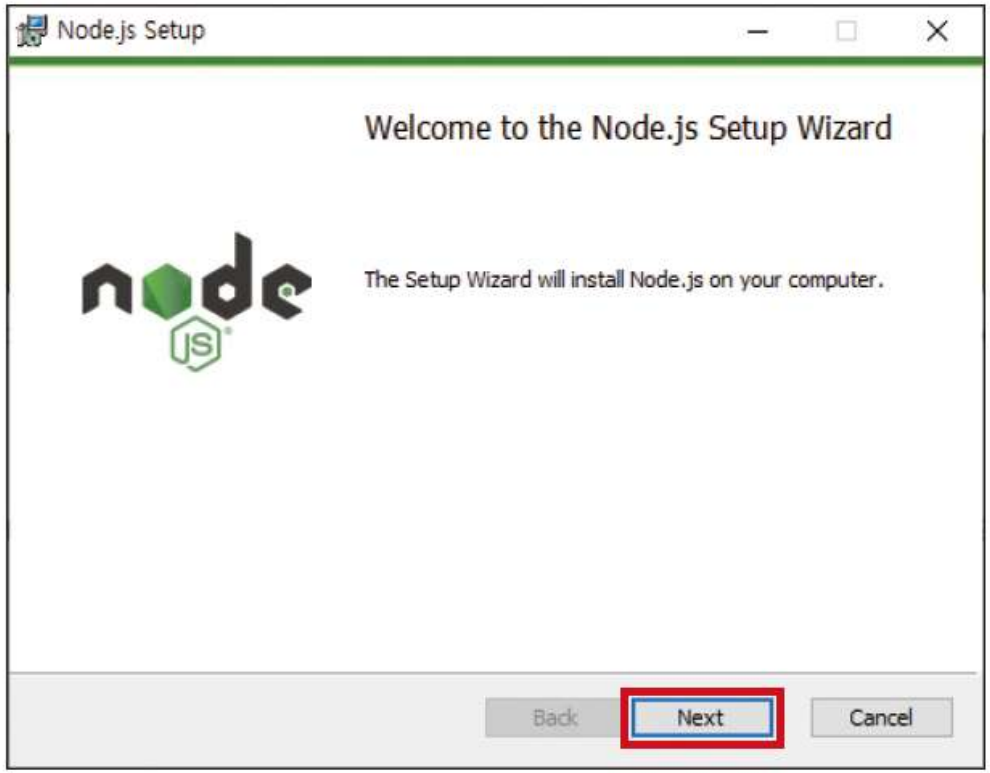
» 윈도(11 기준), 맥(벤투라 기준)

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

1. 노드 설치하기

» 계속 Next 버튼을 눌러 설치

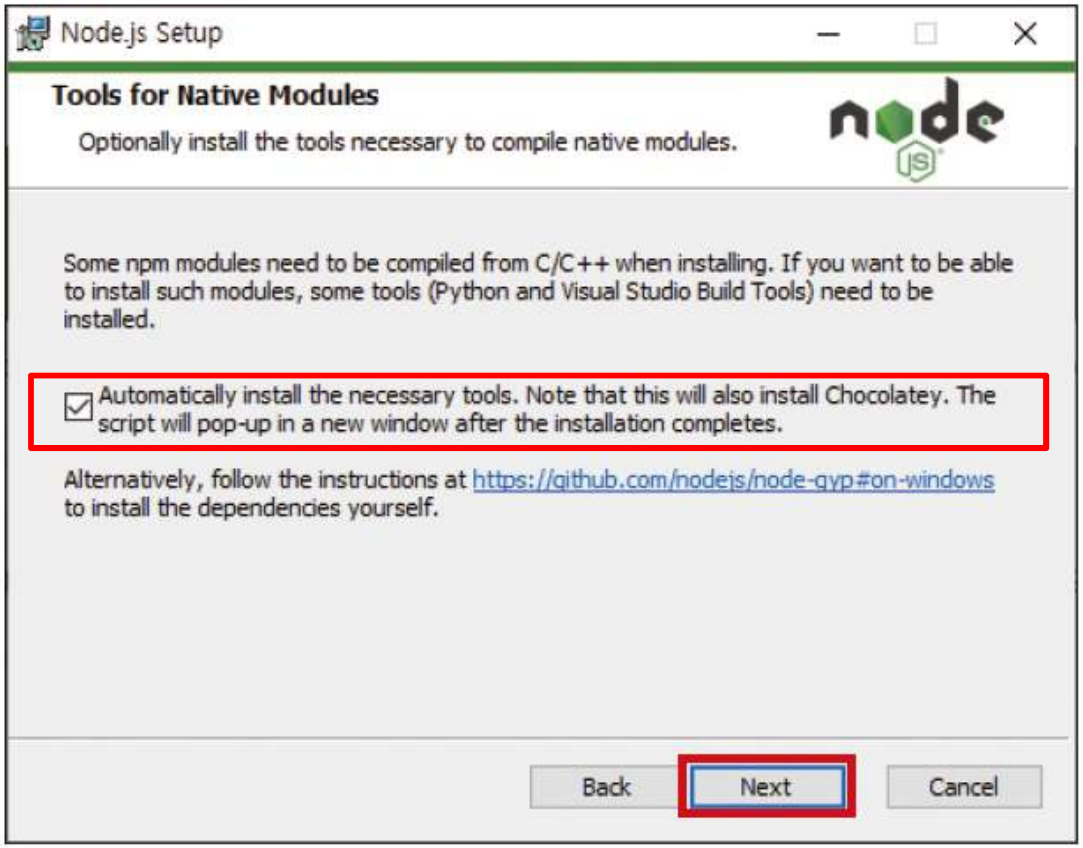
그림 1-19 Setup Wizard 실행



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 수 있지만 버전이 뜨면 설치 성공

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

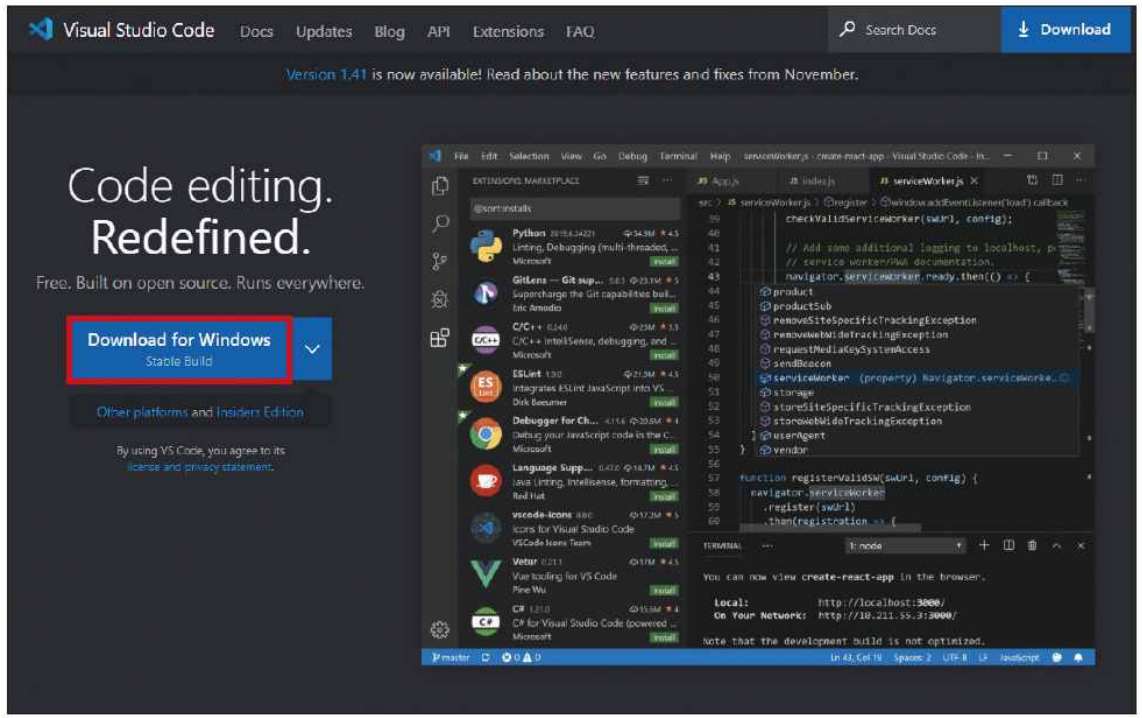
콘솔

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

2. VS Code 설치하기

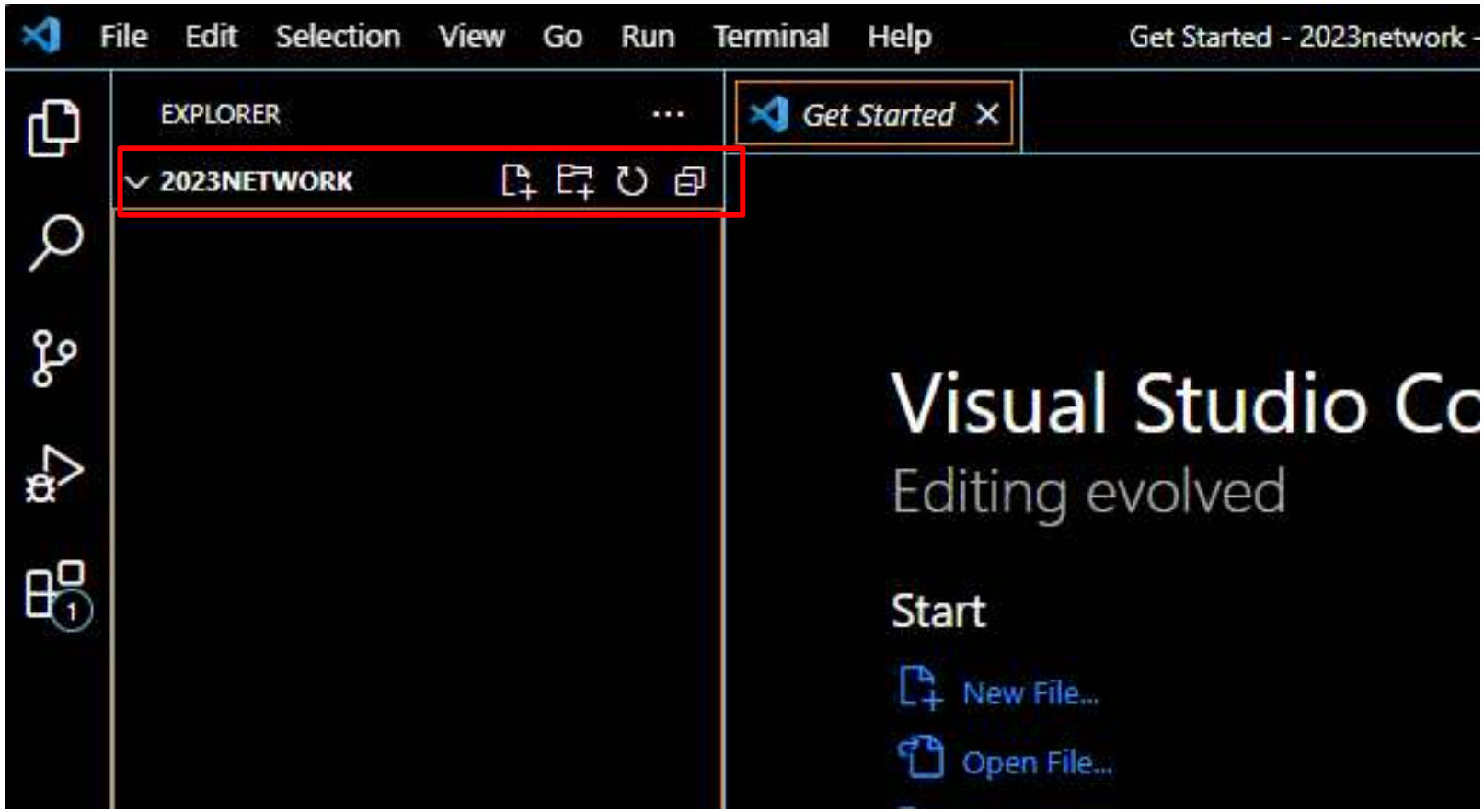
- VS Code: 마이크로소프트에서 제공하는 오픈 소스 코드 에디터
  - 자바스크립트, 노드에 대한 지원이 탁월함
  - 윈도, 맥, 리눅스(GUI) 모두 https://code.visualstudio.com 접속
  - 운영체제에 맞는 파일 설치
  - VS Code 외의 다른 코드 에디터를 사용해도 됨

그림 1-41 VS Code의 공식 사이트



3. 폴더 생성하기

» 한글 경로X 편한 위치에 폴더 생성
- 계정명이 한글이면 제일 고통스러움
- vscode에서 생성한 폴더 열기



수고하셨습니다♥