

보충 회차 · 90분

# IT 기초

AI를 이야기하기 전에 필요한 컴퓨터 · 네트워크 · 데이터의 최소 지식

왜 이 회차가 있나

# 이 네 줄이 읽히면 됩니다

- “Node.js LTS를 설치하세요”
- “localhost:27777로 접속하세요”
- “읽기 전용 계정으로 붙이세요”
- “API 키를 발급하고 헤더에 넣으세요”

개발을 하기 위한 것이 아닙니다. 매뉴얼의 나머지를 막힘없이 읽고 IT 담당자와 같은 말을 쓰기 위한 것입니다.

1부

# 내 컴퓨터 한 대

저장 · 메모리 · 파일 · 터미널 · 런타임

## 부품

# 세 가지만 구분하면 됩니다

01

### 저장장치

꺼도 남는 것 — 서류 캐비닛

02

### 메모리(RAM)

지금 펼쳐 놓은 것 — 책상 위

03

### CPU

실제로 계산하는 것 — 일하는 사람

메모리에 있는 것은 저장을 거쳐야 남습니다. 전원이 나가면 책상 위는 쓸려나가고 캐비닛만 남습니다.

헛갈리는 짝

# 프로그램과 프로세스는 다릅니다

## 프로그램

무엇  
설치되어 있는 것

어디  
저장장치

개수  
하나

창을 닫으면  
그대로 있음

## 프로세스

무엇  
✓ 지금 돌고 있는 것

어디  
✓ 메모리

개수  
✓ 여러 개일 수 있음

창을 닫으면  
✓ 보통 같이 죽음(백그라운드는 예외)

“꼈다 커세요”가 듣는 이유 — 재시작은 메모리를 통째로 비웁니다. 두 번 해도 같으면 원인은 설정 쪽입니다.

## 파일

# 경로 · 확장자 · 두 종류의 파일

- 경로는 주소입니다 — /Users/jinho/work/report.md
- 절대 경로는 어디서 불러도 같은 곳. 에이전트에게는 절대 경로로
- 확장자는 이름표일 뿐 — .csv 를 .xlsx 로 고쳐도 엑셀이 되지 않습니다
- 텍스트 파일(.md .csv .json)은 AI에 그대로, 바이너리(.pdf .xlsx .hwp)는 뽑는 단계가 필요

점으로 시작하는 파일(.env)은 탐색기에서 안 보입니다. 열쇠가 거기 들어 있고, 안 보이니까 같이 공유됩니다.

## 터미널

# command not found 는 “설치 안 됨”이 아닙니다

1

**명령을 칩**`claude --version`

2

**PATH를 뒤집**

미리 정해진 폴더 목록만 찾습니다

3

**못 찾음**`command not found` — 설치와는  
다른 문제

4

**새 창을 연다**

이미 열린 창은 바뀐 PATH를 모릅니다

알아둘 것은 다섯 개 — `cd` · `ls` · `pwd` · `--version` · `Ctrl-C`. 나머지는 필요할 때 찾습니다.

언어와 런타임

# 왜 Node를 깔라고 하나

01

## 언어

사람이 읽고 쓰는 문법 — 파이썬 · 자바스크립트

02

## 런타임

그 코드를 실제로 실행해주는 프로그램 — Node.js  
· Python

03

## LTS

한동안 고쳐주기로 약속한 안정 버전. 업무용은 이것

HyperTeams가 자바스크립트로 만들어져서 Node가 필요합니다. 한글 문서를 읽으려고 뷰어를 까는 것과 같습니다 — 쓸 줄 알 필요는 없습니다.



2부

# 기계 밖으로

네트워크 · 서버 · HTTP · API

## 주소

# 통신은 주소 찾기입니다

01

### IP 주소

어느 컴퓨터인지 — 건물 주소

02

### 포트

그 안의 어느 프로그램인지 — 몇 호실

03

### 도메인 · DNS

이름과, 이름을 주소로 바꿔주는 안내소

04

### localhost

지금 이 컴퓨터. 밖으로 나가지 않습니다

localhost:27777 — 앞은 내 컴퓨터, 뒤는 그 안의 어느 프로그램. 웹 주소에 포트가 안 보이는 것은 443이 기본값이라 생략되기 때문입니다.

## 역할

# 서버는 특별한 기계가 아닙니다

### HyperTeams

어디서 도나

내 컴퓨터

설치

필요

내 파일

직접 만짐

컴퓨터를 꺼두면

멈춤

### Connect

어디서 도나

✓ 웹(SaaS)

설치

✓ 없음

내 파일

✓ 올린 것만

컴퓨터를 꺼두면

✓ 상관없음

요청을 받아 답하는 쪽이 서버, 묻는 쪽이 클라이언트입니다. 노트북에서 프로그램 하나를 켜면 그 노트북이 서버입니다.

## HTTP

# 상태 코드 네 개만 구분하면 됩니다

01

## 401 — 누구인지 모르겠음

열쇠 문제. 안 보냈거나 틀렸습니다

02

## 403 — 권한 없음

누군지는 앎. 권한을 줘야 풀립니다

03

## 404 — 그런 주소 없음

주소 문제. 열쇠를 새로 발급해도 안 풀립니다

04

## 500 — 서버가 터짐

대개 내 잘못이 아닙니다

비밀은 주소가 아니라 헤더에 실어 보냅니다 — 주소는 브라우저 기록 · 서버 로그에 그대로 남습니다.

## API

# 프로그램에게 내주는 창구

- 엔드포인트 — 창구 하나의 주소. GET /api/v1/tasks
- 메서드 + 주소의 조합이 곧 기능 — 같은 /tasks 라도 GET은 조회, POST는 생성
- 키 — 화면의 로그인에 해당. 용도마다 따로 발급합니다
- SDK — 그 창구를 언어별로 미리 감싸둔 도구 상자. 하는 일은 같습니다

한 건을 눈으로 보고 판단하는 일이면 화면이 낫습니다. API는 반복 · 대량 · 자동의 도구입니다.

3부

# 데이터와 열쇠

형식 · 데이터베이스 · 권한

# 왜 어떤 파일은 변환이 필요한가

## 그대로 읽힘

예  
JSON · CSV · 마크다운

성격  
텍스트

구조  
항목과 값이 살아 있음

실패하면  
드뭇

## 뽑아내야 함

예  
✓ PDF · 한글 문서 · 스캔본

성격  
✓ 인쇄 · 저장용 바이너리

구조  
✓ 좌표만 있고 표 구조는 없기도

실패하면  
✓ “AI가 우리 문서를 못 읽어요”

한글이 깨지면 파일이 아니라 인코딩을 의심하세요 — UTF-8로 저장한 것을 CP949로 읽으면 깨집니다.

데이터베이스

# 표 · 행 · 열 · 키

01

## 테이블 · 행 · 열

표 하나, 가로 한 줄이 사건 하나, 세로 한 칸이 항목

02

## 기본 키 · 외래 키

행을 지목하는 값과, 다른 표를 가리키는 값

03

## SQL

묻는 말. SELECT는 조회, UPDATE · DELETE는 되돌리기 어려움

UPDATE · DELETE 는 조건을 빠뜨리면 표 전체에 적용됩니다. 그래서 도구 설정이 아니라 읽기 전용 계정으로 막습니다 — 지키는 주체가 다릅니다.



권한

# 누구인가와 무엇을 해도 되는가

- 인증 — 누구인가. 실패하면 401
- 인가 — 무엇을 해도 되는가. 실패하면 403
- 최소 권한 — 필요한 것만 주고 막히면 그때 넓힙니다. 반대로 하면 “나중”은 오지 않습니다
- 에이전트도 자기 권한이 아니라 주어진 계정의 권한으로 움직입니다

열쇠는 발급보다 회수가 어렵습니다. 이름을 알아보게 짓고, 새면 고치는 것이 아니라 폐기합니다.

## 정리

# 이 회차가 남기는 것

- 1 재시작이 고치는 것과 못 고치는 것을 구분한다
- 2 command not found 를 보면 새 터미널부터 연다
- 3 401 · 403 · 404 가 서로 다른 문제라는 것을 안다
- 4 API 키는 용도마다 나누고, 새면 폐기한다
- 5 막는 것은 지시가 아니라 권한이 한다