

보충 회차 · 90분

IT 기초

AI를 이야기하기 전에 필요한 컴퓨터 · 네트워크 · 데이터의 최소 지식

여는 말

- 기술 배경이 있는 수강생에게는 건너뛰어도 된다고 먼저 말해주세요. 붙잡아두면 나머지 회차의 집중이 떨어집니다.
- 반대로 비개발 수강생에게는 이 회차가 나머지 전부의 전제라고 못박으세요.
- 질문을 부끄러워하지 않게 만드는 것이 이 회차의 실질적인 목표입니다.

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.

목표 선언

- 네 줄을 소리 내어 읽고 “지금 몇 개가 읽히십니까”를 물어보세요. 손을 들게 하지 말고 그냥 물어만 봐도 됩니다.
- 회차 끝에 같은 슬라이드로 돌아와 다시 물으면 학습 효과가 눈에 보입니다.
- 이 네 줄은 각각 준비물 · 첫 접속 · 데이터베이스 · API 키 장에서 그대로 나옵니다.

1부

내 컴퓨터 한 대

저장 · 메모리 · 파일 · 터미널 · 런타임

1부 들어가기

- 순서가 “이 기계 → 기계 밖 → 오가는 데이터 → 누가 열 수 있나”입니다. 판서로 적어두면 수강생이 길을 잃지 않습니다.

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.

세 부품

- “메모리가 부족한데 디스크를 비우면 되나요”가 이 슬라이드로 정리됩니다. 캐비닛을 비워도 책상은 그대로입니다.
- 저장 버튼이 왜 존재하는지를 여기서 한 번 짚어주면 뒤가 편합니다.

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.

프로그램 vs 프로세스

- `hyperteams` 가 “이미 돌고 있습니다”라고 할 때는 프로세스 얘기입니다.
- `--background` 는 창에서 떼어내는 것이지 컴퓨터에서 떼어내는 것이 아닙니다 — 노트북을 덮으면 예약도 멈춥니다.
- 이 오해가 “밤사이 예약이 안 돌았어요” 문의의 1순위입니다.

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.

파일과 경로

- 작업 폴더 등록이 곧 “지금 있는 폴더”를 못박는 일이라고 이어주세요.
- Windows는 확장자를 기본으로 숨깁니다 — 보고서.pdf.exe 사례를 한 번 보여주면 강하게 남습니다.
- 마지막 줄은 뒤의 열쇠 관리 슬라이드에서 다시 나옵니다.

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.

터미널

- 설치 지원 문의의 절대다수가 이 네 단계에서 끝납니다. 실습으로 직접 겪게 하는 것이 가장 빠릅니다.
- Ctrl-C 를 꼭 손으로 눌러보게 하세요. 빠져나갈 방법을 알면 터미널 공포가 줄어듭니다.
- `rm · sudo · curl | bash` 세 가지는 붙여넣기 전에 물어보라고 못박으세요.

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.

언어와 런타임

- “Node를 깔면 자바스크립트를 배워야 하나요”가 실제로 자주 나오는 질문입니다.
- AI 예제가 파이썬인 것은 관행이라고 짚어주세요 — 못 하는 것이 아니라 보여주기 좋은 공용어입니다.
- HyperTeams 사용에 파이썬은 필요 없습니다.

2부

기계 밖으로

네트워크 · 서버 · HTTP · API

2부 들어가기

- 여기부터가 이 회차의 본론입니다. 1부는 발판이고 2부는 API 키·엔드포인트 같은 실무 단어가 쏟아지는 자리입니다.

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.

네트워크

- localhost 주소를 동료에게 보내면 안 되는 이유를 꼭 말해주세요 — 그 사람의 컴퓨터를 가리킵니다.
- 방화벽은 보통 밖에서 안으로만 막습니다. 터널이 “안에서 밖으로 먼저 걸어 그 통로를 되쓰는” 방식인 이유입니다.
- AI 응답이 느린 것은 대개 회선이 아니라 모델이 생각하는 시간입니다.

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.

서버 · 클라우드

- 클라우드는 남의 컴퓨터라는 정의를 그대로 쓰세요. 농담 같지만 정확합니다.
- 규제 산업이 온프레임을 고집하는 이유는 성능이 아니라 데이터의 물리적 위치입니다.
- 두 제품을 같이 쓰는 조직이면 시스템 연결 구조(Connect가 클라이언트, 내 컴퓨터가 서버)를 여기서 그려주세요.

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.

요청과 응답

- 401과 403을 구분하는 것이 실무에서 가장 값어치가 큼니다 — 고치는 사람이 다릅니다.
- 404가 나오는데 키를 계속 재발급하는 것이 가장 흔한 헛수고입니다.
- HTTPS는 전송 구간만 지킵니다. 도착한 서버가 그 데이터로 무엇을 하는지는 별개입니다.

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.

API

- “OpenAI SDK의 base 주소만 바꾸면 된다”가 왜 성립하는지 — 창구 모양이 같으면 도구 상자를 그대로 씁니다.
- 키를 용도마다 나누는 이유는 회수 때문입니다. 이름이 없으면 어느 것을 지울지 모르게 됩니다.
- 레이트 리밋(429)과 POST 재시도의 중복 생성은 자동화에 붙일 때 반드시 짚어야 합니다.

3부

데이터와 열쇠

형식 · 데이터베이스 · 권한

3부 들어가기

- 마지막 3부는 실무 사고와 가장 가까운 자리입니다. 시간이 모자라면 2부를 줄이지 말고 1부를 줄이세요.

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.

데이터 형식

- 회사 자료의 대부분이 비구조화(줄글)이고, 그것이 AI가 회사에서 값어치를 갖는 이유입니다.
- PDF는 주고받기에는 표준이지만 다시 가공하기에는 최악에 가깝습니다. 원본을 달라고 하세요.
- JSON은 에이전트에게 “표로 정리해줘”라고 시키면 됩니다. 괄호를 사람이 셀 일이 아닙니다.

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.

데이터베이스

- “최종.xlsx가 번식하기 시작하면 DB로 갈 때”라는 신호를 짚어주세요.
- 에이전트에 DB를 붙이면 SQL을 대신 써줍니다 — 사람이 SQL을 몰라도 되게 하는 것이 그 기능의 목적입니다.
- 백업은 도는지보다 복구를 해본 적이 있는지가 진짜 질문입니다.

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.

계정과 권한

- 관리자 계정의 2단계 인증은 예외 없이 켜라고 하세요. 사고 대부분은 다른 데서 새어나온 비밀번호에서 시작합니다.
- 권한은 사람이 아니라 역할에 붙입니다 — 퇴사·이동 때 드러납니다.
- 읽기 전용 계정을 준 에이전트는 시켜도 못 지웁니다. 이 한 줄이 3부의 결론입니다.

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

닫는 말

- 2번 슬라이드의 네 줄로 돌아가 다시 읽어주세요. 대부분 이제 읽힙니다.
- 본문 12장과 용어집의 「IT 기초」 절을 안내하고, 실습(요청 하나를 직접 뜯어보기)은 과제로 내주면 좋습니다.
- 다음 회차는 AI 기술 기초입니다 — 같은 눈으로 모델 자체를 봅니다.