HYPERTEAMS 5회차 · 120분

실전 자동화

예약으로 파이프라인을 만들고, 절차를 스킬로 담고, 자료를 모읍니다

발표자 노트 · 01

여는 말

- 4회차가 “바깥으로 문을 여는” 이야기였다면 이 회차는 “아무도 안 볼 때 도는 것”입니다.
- 실제로 돌려본 파이프라인 하나를 가져와 예시로 쓰면 가장 잘 붙습니다.

파이프라인

예약은 두 종류입니다

task

무엇을
에이전트에게 맡김

언제
매번 판단이 필요

성격
비싸고 매번 다름

예
원고 작성 · 로그 분석

terminal

무엇을
✓ **명령을 그대로 실행**

언제
✓ **정해진 명령**

성격
✓ **싸고 결정적**

예
✓ **수집 · 형식 변환 · 렌더 · 백업**

판단 없는 단계를 task 로 두는 것이 운영비의 대부분입니다. 규칙으로 100% 표현되면 AI를 쓸 이유가 없습니다.

발표자 노트 · 02

예약 두 종류

- Phase 1 의 적합성 판단과 같은 질문입니다 — 매번 판단이 필요한가.
- 비용 줄이는 1순위가 판단 없는 단계를 terminal 로 옮기는 것입니다.
- task 만 자율성·엔진·모델을 고릅니다. terminal 은 해당 없음입니다.

단계를 잇는 법

세 가지 방법과 네 가지 원칙

01

시차를 둔다

가장 단순. 앞이 늦으면 뒤가 빈손으로 돕니다

02

앞이 뒤를 부른다

빈손 실행이 없음. 열쇠는 환경변수로

03

한 업무가 전부

권하지 않음 — 컨텍스트가 차서 뒤로 갈수록 흐려집니다

원칙 넷 — 단계마다 파일로 남기기 · 이미 있는 것은 건너뛰기 · 실패를 알리기 · 자율성은 낮게 시작.

발표자 노트 · 03

파이프라인 설계

- 중간 산출물이 파일로 있으면 실패한 단계부터 다시 돌릴 수 있습니다. 메모리로만 이으면 처음부터입니다.
- 예약이 쌓이면 목적이 끝난 것이 계속 돌며 비용을 씁니다. 분기마다 정리하세요.
- 파이프라인 특유의 문제 — 수집이 몇 주째 실패했는데 원고 단계는 계속 돌고 있는 상태입니다.

스킬

절차를 어디에 담나

재사용 단위

지시문 템플릿
매번 붙여넣는 글

서브 에이전트
역할 · 말투 · 규칙

스킬
절차와 그에 딸린 자료

사는 곳

지시문 템플릿
✓ 내 메모장 — 그 사람 것입니다

서브 에이전트
✓ Connect 워크스페이스

스킬
✓ 작업 폴더 안의 파일 — git 으로 팀이 공유

SKILL.md 하나면 됩니다. 폴더가 곧 상태라, 폴더를 복사하면 절차가 따라옵니다.

발표자 노트 · 04

스킬로 절차를 담기

- `description` 이 가장 중요합니다 — 모델은 그 한 줄을 읽고 이 스킬을 쓸지 판단합니다.
- "무엇을 하는지"만 적고 "언제 쓰는지"를 빼면 안 불립니다.
- 네 자리(project · user · policy · plugin) 중 편집하는 곳은 project 뿐입니다. 그래야 폴더와 함께 이동합니다.
- 터미널에서 직접 치든 화면에서 시키든 같은 스킬이 보입니다.

스킬 · 만들지 말 때

스킬은 절차이지 도구가 아닙니다

- 한 번 쓰고 말 것 → 그냥 대화로
- 지시가 매번 크게 달라짐 → 대화로 좁혀가기
- 팀 전체가 쓰는 말투·역할 → Connect 서브 에이전트
- 외부 시스템을 부르는 것 → MCP 도구

절차가 바뀌었는데 스킬이 그대로면 틀린 절차를 자신 있게 반복합니다. 점검 목록을 스킬 안에 남기세요.

발표자 노트 · 05

언제 만드나

- 스킬이 맞는 신호 셋 — 같은 절차를 반복한다, 순서를 틀리면 사고가 난다, 딸린 자료가 있다.
- "하지 말 것"을 반드시 적으세요. 규칙은 금지형이 가장 잘 지켜집니다.
- 폴더 이름과 `name` 을 맞추세요. 어긋나면 어느 스킬이 불렸는지 추적이 안 됩니다.

문서

PDF와 나머지는 사본의 역할이 다릅니다

PDF

원본을 직접 읽나
읽습니다 (쪽 범위로)

사본의 역할
보조 — 검색 · 요약용

표 · 그림
원본을 봐야 정확

docx · xlsx · pptx

원본을 직접 읽나
✓ 못 읽습니다

사본의 역할
✓ 전부

표 · 그림
✓ 사본에 담긴 만큼만

엑셀은 사본이 전부라 사본이 부실하면 그걸로 끝입니다 — 중요한 수치는 사람이 대조하세요.

발표자 노트 · 06

문서를 자료로 바꾸기

- 네 확장자만 됩니다(pdf · docx · xlsx · pptx). 확장자로 판단하므로 파일 이름이 정확해야 합니다.
- 사본 이름이 `보고서.pdf.md` 인 이유는 원본과 짝을 유지하기 위해서입니다.
- 문서가 많은 폴더는 사본 만들기가 첫 단계입니다 — 100개를 다 읽히는 게 아니라 검색으로 3개로 좁힌 뒤 읽힙니다.
- 텍스트 사본은 git 에 넣는 쪽을 권합니다. 작고 변경 이력이 의미 있습니다.

사례

콘텐츠 공장 — 6단계

1
수집
terminal · 정해진 곳에서 받아 저장

2
원고
task · 자료가 없으면 아무것도 하지 말 것

3
음성·이미지
terminal · 서로 안 기다림

4
렌더
terminal · 매번 같은 결과

5
점검
task · 발견해도 고치지 말고 보고까지

6
게시
사람 · 되돌릴 수 없음

단계마다 파일로 남기면 실패한 곳부터 다시 돌릴 수 있습니다. 폴더는 소스와 렌더로 나눕니다.

발표자 노트 · 07

콘텐츠 공장

- 여섯 중 task 는 둘뿐입니다. 판단이 필요한 것만.
- 게시만 사람인 이유는 하나 — 되돌릴 수 없기 때문입니다.
- 점검 단계가 고치기 시작하면 무엇이 원래 결과였는지 알 수 없게 됩니다. 보고까지만 시키세요.
- 구조는 그대로 두고 수집과 렌더만 바꾸면 일일 운영 리포트 · 가격 모니터링 · 회의록 자동화가 됩니다.

처음 2주

한 번에 자동화하지 마세요

시기

1~3일
손으로

4~7일
예약으로

2주차
자율성 올리기

이후
무인

하는 일

1~3일
✓ 각 단계를 한 번씩 실행. 어디서 막히는지 확인

4~7일
✓ 걸되 매일 결과를 눈으로 확인

2주차
✓ 점검 단계의 보고만 확인

이후
✓ 실패 알림이 올 때만 개입

1~3일을 건너뛰면 여섯 단계가 동시에 처음 도는 걸 보게 되고, 어느 단계가 원인인지 못 가립니다.

발표자 노트 · 08

흔한 실패

- 빈 원고가 매일 나온다 → 수집이 실패했는데 원고가 계속 돎. 전제 확인을 지시문에.
- 음성·이미지 수가 안 맞는다 → 실패한 항목을 건너뛰고 진행. 점검 단계에서 수를 대조.
- 비용이 예상보다 크다 → 판단 없는 단계를 task 로 둔 것. terminal 로 옮기세요.
- 며칠째 아무것도 안 나온다 → 노트북이 닫혀 있었습니다. 의외로 흔합니다.

수집

브라우저는 마지막 순서입니다

1
공식 API
있으면 여기서 끝

2
피드
RSS · 사이트맵 · JSON 엔드포인트

3
정적 HTML
fetch + 파서

4
브라우저
스크립트가 돌아야 나올 때만

브라우저는 가장 비싸고 가장 자주 깨집니다 — 상대가 마크업을 조금만 바꿔도 멈춥니다.

발표자 노트 · 09

수집 순서

- 1번에서 끝나는 경우가 대부분입니다. 4번부터 시작하는 사람이 많아서 순서를 못박아 둡니다.
- 기술보다 먼저인 질문은 “해도 되는 일인가”입니다 — robots.txt · 이용약관 · 로그인 장벽 · 개인정보.
- 가장 편한 경우는 내 사이트, 내 데이터, 허락받은 곳입니다. 그 밖이면 먼저 물어보는 게 빠릅니다.

스텔스를 얹으면 오히려 더 눈에 띌 수 있습니다

Crawlee 는 자체 지문 생성기가 기본으로 켜져 있습니다 — 둘이 같은 값을 각자 덮어씁니다

발표자 노트 · 10

지문은 한쪽만

- navigator.webdriver · UA · 언어 · 화면 크기를 두 도구가 각자 씁니다. 어긋난 조합이 나오면 아무것도 안 썼을 때보다 티가 납니다.
- “스텔스를 붙였는데 더 막힌다”의 흔한 정체입니다. useFingerprints: false 로 한쪽을 끄세요.
- 순서는 아무것도 안 붙이고 시도 → 안 되면 한쪽만. 붙이는 순간 puppeteer 버전을 따라다니는 유지비가 생깁니다.
- 탐지 쪽도 상대 마크업도 계속 바뀝니다. 한 번 만들고 끝나지 않습니다.

영상

받는 순서가 곧 비용입니다

필요한 것

무슨 말을 했나
내용

제목 · 길이 · 조회수
메타데이터

자막이 없는 영상
내용

화면에 나온 것
자료 화면 · 제품

받을 것 · 크기

무슨 말을 했나
✓ **자막 — KB**

제목 · 길이 · 조회수
✓ **JSON — KB**

자막이 없는 영상
✓ **오디오만 — MB**

화면에 나온 것
✓ **영상 — 수백 MB**

대부분의 업무는 첫 두 줄에서 끝납니다. 도구는 yt-dlp 하나이고 ffmpeg가 함께 필요합니다.

발표자 노트 · 11

영상에서 자료 뽑기

- 자동 자막은 원문 그대로가 아닙니다 — 고유명사·숫자·제품명이 자주 틀립니다. 인용하려면 확인이 필요합니다.
- yt-dlp 버전을 고정하지 마세요. 플랫폼이 바뀌고 yt-dlp 가 그것을 뒤따라 고칩니다.
- 수집이 0건이면 결과 파일을 쓰지 말고 실패로 알려야 합니다. 빈 결과가 내려가면 다음 단계가 그걸로 원고를 씁니다.
- 채널 전체를 한 번에 긁는 것이 가장 흔한 실수입니다.

공문서

hwp 는 읽기 전용, hwpx 는 쓰기까지

01

읽기

hwp · hwpx 모두 마크다운으로. 표는 HTML <table> 로 나옵니다

02

만들기

마크다운으로 내용만 쓰면 항목부호 8단계와 표준 서식은 맡깁니다

03

서식 채우기

무엇을 채울 수 있는지 먼저 보고, 값은 JSON 파일로 넘깁니다

04

고치기

원본 서식을 유지한 채. .hwp 는 이 자리에서 안 됩니다

넘기기 전에 validate 로 구조를 검사하세요. 제출은 사람이 합니다 — 되돌릴 수 없기 때문입니다.

공문서 다루기

- 가장 자주 사고가 나는 지점 — `.hwp` 는 바이너리라 읽기만 되고 그 포맷으로 다시 쓸 수 없습니다.
- 값을 명령줄이 아니라 `-j` 파일로 넘기는 이유는 셸 히스토리와 프로세스 목록에 남기 때문입니다.
- 신청서에는 개인정보가 들어갑니다. 원본을 덮어쓰지 말고 사본으로 작업하세요.
- 암호·DRM 문서는 열리지 않습니다. 스캔본은 이미지라 글자가 안 나옵니다.

실습 · 60분

파이프라인 하나를 끝까지 만들어보기

1

세 단계 그리기

각 단계가 **무엇을 파일로 남기는지** 적습니다

2

1단계만 먼저

파일이 생긴 것과 내용이 맞는 것은 다릅니다

3

2·3단계 잇기

간격은 평소 소요의 3배 이상

4

가운데 깨뜨리기

1단계 산출물을 지우고 뒷단계가 어떻게 되는지

5

비용과 정리

세 단계 × 매일이면 한 달에 90번입니다

가장 위험한 결과는 “예전 파일을 찾아 쓰는 것”입니다 — 오류가 안 나서 아무도 모르고, 어제 데이터로 오늘 보고가 나갑니다.

발표자 노트 · 13

실습 진행

- 첫 파이프라인에 발송이나 게시를 넣지 않게 하세요. 4단계에서 일부러 깨뜨릴 때 진짜 사고가 납니다.
- 각 단계가 파일을 남겨야 중간부터 다시 돌릴 수 있습니다 — 이게 "재실행 가능"의 실제 모습입니다.
- 한 단계의 실패가 다음 단계에서 정상으로 둔갑하는 것이 파이프라인의 가장 흔한 실패입니다.
- 60분이 부족하면 4단계까지만 하고 5단계는 숙제로 돌리세요.

정리

가져갈 것

1. 판단이 있으면 task, 없으면 terminal — 여기가 운영비의 대부분
2. 단계마다 파일로 남긴다. 그래야 실패한 곳부터 다시 돈다
3. 스킬은 폴더에 산다 — git 으로 팀이 공유하고 되돌릴 수 있다
4. 문서는 사본을 먼저 만들고 검색으로 좁힌 뒤 읽힌다
5. 수집은 API → 피드 → HTML → 브라우저 순서로만 내려간다
6. 영상은 메타데이터 → 자막 → 오디오 → 영상 순서로만 내려간다
7. 되돌릴 수 없는 동작 앞에만 사람을 세운다

발표자 노트 · 14

닫는 말

- 마지막 줄이 이 회차 전체를 관통합니다 — 가드레일의 기준과 같습니다.
- 다음은 각자 자기 업무에서 파이프라인 후보를 하나 골라 오는 것으로.