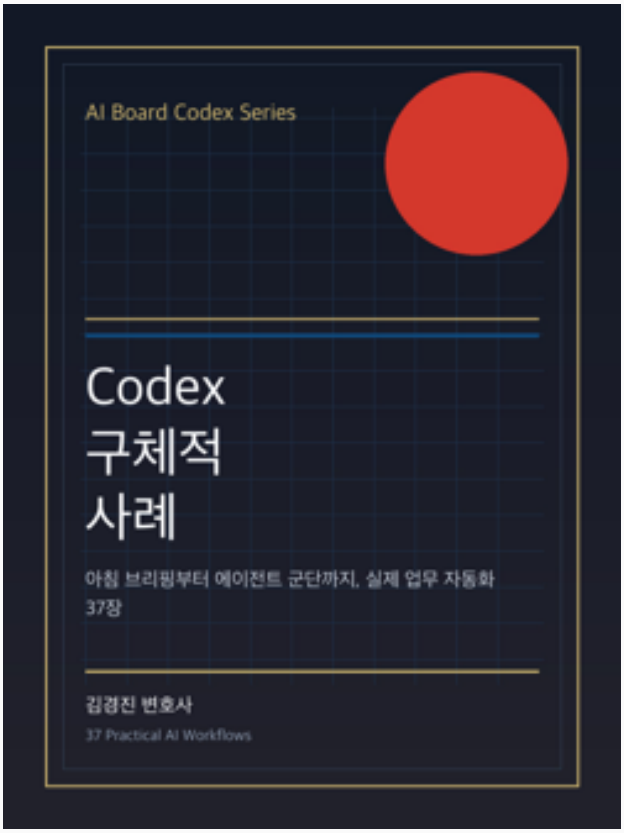
# Codex 구체적 활용사례 37

아침 브리핑부터 에이전트 군단까지, 실제 업무 자동화 37장




김경진 변호사

Korean PDF edition

# Codex 구체적 활용사례 37

김경진 변호사

이 글은 Codex와 AI 에이전트로 개인 업무, 데이터 처리, 마케팅, 영업, 문서, 개발, 브라우저 제어를 실제 업무에 연결하는 37개 사례를 묶은 안내서입니다.

## 목차

Codex 구체적 활용사례 37 - 제1장. 개인 맞춤형 아침 브리핑 및 일정 자동화

Codex 구체적 활용사례 37 - 제2장. 영수증 및 청구서 대량 자동 처리 프로세스

Codex 구체적 활용사례 37 - 제3장. 웹사이트 및 방문 페이지(Landing Page) 완벽 복제 및 생성

Codex 구체적 활용사례 37 - 제4장. 풀스택 SaaS 및 웹 애플리케이션 구축

Codex 구체적 활용사례 37 - 제5장. 모바일 앱(iOS/Android) 기획 및 개발

Codex 구체적 활용사례 37 - 제6장. 크롬 익스텐션(Chrome Extension) 자체 제작: 5분 만에 만드는 나만의 생산성 도구

Codex 구체적 활용사례 37 - 제7장. 지저분한 CSV 데이터 자동 정제 및 전처리: 정규식 없이 자연어로 데이터 다루기

Codex 구체적 활용사례 37 - 제8장. 복수의 스프레드시트 데이터 병합 및 피벗 테이블 생성: 맥락을 이해하는 AI 데이터 결합

Codex 구체적 활용사례 37 - 제9장. 보고서 내 이상치(Outlier) 및 누락 데이터 사전 감지: 경영진 회의 직전, AI로 완벽한 데이터 감사(Audit) 진행하기

Codex 구체적 활용사례 37 - 제10장. SQL 쿼리 디버깅, 속도 최적화 및 뷰(View) 자동 변환: 느리고 복잡한 쿼리를 AI로 완벽하게 다듬기

Codex 구체적 활용사례 37 - 제11장. 레거시 코드 및 데이터베이스 스키마 자동 문서화(Data Dictionary): 암호 같은 과거의 유산을 AI로 해독하기

Codex 구체적 활용사례 37 - 제12장. 경쟁사 웹사이트, SEO, 가격 정책 및 리뷰 스크래핑 분석: 우리 비즈니스의 약점과 기회 한눈에 파악하기

Codex 구체적 활용사례 37 - 제13장. 구글 맵 기반 리드(Lead) 수집 및 맞춤형 콜드 이메일 작성: 영업을 자동화하는 마법

Codex 구체적 활용사례 37 - 제14장. 메타(Meta) 광고 라이브러리 스크래핑 및 광고 성과 분석 대시보드: 이기는 광고의 비밀 풀기

Codex 구체적 활용사례 37 - 제15장. 유튜브 채널 전략 기획 및 썸네일/에셋 생성 자동화: 터지는 콘텐츠 무한 생성기

Codex 구체적 활용사례 37 - 제16장. 유튜브 및 회의록 기반 소셜 미디어(LinkedIn/Twitter) 콘텐츠 재가공: 터지는 콘텐츠 무한 생성기

Codex 구체적 활용사례 37 - 제17장. 긴 동영상 하이라이트 컷 자동 편집 및 브랜딩 에셋 추가: 비디오 편집의 완전 자동화

Codex 구체적 활용사례 37 - 제18장. 노션(Notion) 문서를 커뮤니티(Circle) 포스트로 실시간 동기화: 지루한 발행 작업 자동화

Codex 구체적 활용사례 37 - 제19장. 이메일 수신함 자동 분류(Triage) 및 맞춤형 답장 초안 작성: AI 비서가 내 이메일 톤을 훔치다

Codex 구체적 활용사례 37 - 제20장. 고객 피드백 서베이 데이터를 분석한 전문 경영진 보고서 제작: 수천 개의 피드백을 단 1분 만에 시각화하기

Codex 구체적 활용사례 37 - 제21장. 주간 CEO 리뷰 및 비즈니스 핵심 지표 추적 보고서: AI가 기억하는 자동화된 경영진 보고

Codex 구체적 활용사례 37 - 제22장. 회의록 비교 분석을 통한 누락된 액션 아이템(Action Item) 추출: 아무것도 놓치지 않는 AI 비서

Codex 구체적 활용사례 37 - 제23장. 고객 이탈(Churn) 위험 예측 및 업셀링(확장) 기회 모니터링: 통화 녹음에서 돈을 찾아내는 마법

Codex 구체적 활용사례 37 - 제24장. 실패한 영업 기회(Lost Deals) 부활을 위한 CRM 데이터 연동: 죽은 계약을 살려내는 영업 사원 AI

Codex 구체적 활용사례 37 - 제25장. 신규 프로덕트 GTM(Go-to-Market) 전략 수립 및 이메일 시퀀스 작성: 15분 만에 끝내는 마케팅 런칭 전략

Codex 구체적 활용사례 37 - 제26장. 인터랙티브 슬라이드쇼 및 프레젠테이션 자동 디자인: 브랜드 맞춤형 PPT 10분 만에 찍어내기

Codex 구체적 활용사례 37 - 제27장. 데스크톱 앱 및 웹사이트에 대한 전문적인 UX 오딧(Audit) 수행: AI를 우리 회사의 QA 테스터로 고용하기

Codex 구체적 활용사례 37 - 제28장. n8n, Make.com 등 자동화 워크플로우(JSON) 코드 변환 및 구축: 시각적 툴과 코딩의 완벽한 융합

Codex 구체적 활용사례 37 - 제29장. 브라우저 제어(Computer Use)를 통한 온라인 쇼핑 및 웹 제어 자동화: AI 개인 쇼핑 비서 두기

Codex 구체적 활용사례 37 - 제30장. 스마트폰/스마트워치 원격 지시(Dispatch)를 통한 데스크톱 백그라운드 작업: 시간과 공간의 제약 허물기

Codex 구체적 활용사례 37 - 제31장. 에어테이블(Airtable) 데이터를 쓴 맞춤형 인보이스/문서 일괄 생성: 수작업 서류 업무의 완전한 자동화

Codex 구체적 활용사례 37 - 제32장. 주식/가상화폐 포트폴리오 분석 및 매매 자동화 대시보드 구축: 나만의 AI 프라이빗 뱅커 고용하기

Codex 구체적 활용사례 37 - 제33장. Git Worktree를 쓴 다국어(번역) 및 다수 디자인 동시 병렬 개발: 분신술로 일하는 AI 개발팀 만들기

Codex 구체적 활용사례 37 - 제34장. 코드 리뷰, 테스트 주도 개발(TDD) 및 아키텍처 자동 리팩토링: 내 코드를 뜯어고치는 엄격한 시니어 엔지니어 고용하기

Codex 구체적 활용사례 37 - 제35장. 거대한 PDF 및 책 데이터를 챕터별로 자동 분할 및 요약 처리: 400MB 문서의 장벽을 부수는 AI

Codex 구체적 활용사례 37 - 제36장. 유튜브 댓글 데이터 감성 분석 및 맞춤형 대시보드 구축: 고객의 목소리를 데이터 앱으로 만들기

Codex 구체적 활용사례 37 - 제37장. 대규모 에이전트 군단(Agent Swarm)을 쓴 자동 이력서 작성 및 일자리 검색: 취업 준비의 판도를 바꾸다

# Codex 구체적 활용사례 37 - 제1장. 개인 맞춤형 아침 브리핑 및 일정 자동화

제1장. 개인 맞춤형 아침 브리핑 및 일정 자동화

AI 에이전트(예: Claude Co-work,

Codex

)를 써서 매일 아침 데스크톱이나 스마트폰으로 완벽하게 개인화된 브리핑을 받는 방법입니다. 과거에는 뉴스레터를 일일이 열어보고 캘린더를 확인해야 했지만, 이제는 AI가 지정된 시간에 백그라운드에서 모든 정보를 수집하고 요약하여 Slack이나 바탕화면으로 전송합니다. 이 워크플로우의 핵심은 예약된 작업(Scheduled Tasks)과 커넥터(Connectors)의 사용입니다. AI에게 사용자의 Gmail, Google 캘린더, 그리고 Slack 권한을 부여하면, AI가 스스로 이메일을 스캔하고 일정을 파악합니다. 업계 최신 뉴스나 트렌드를 웹 검색을 통해 수집하여 사용자에게 가치 있는 인사이트만 선별해 줍니다.

핵심 설정 및 워크플로우:

커넥터 연결: AI 설정 메뉴에서 Gmail, Google Calendar, Slack 커넥터를 활성화합니다.

자동화 스케줄링: 매일 아침 특정 시간(예: 오전 7시)에 실행되도록 예약 작업(Scheduled Task)을 설정합니다. 컴퓨터가 켜져 있어야 하므로 백그라운드 실행 설정을 켜두거나, 원격 서버(Mac Mini 등)를 쓸 수 있습니다.

예제 프롬프트 (프롬프트 창에 입력 또는 스킬로 저장):

"매일 아침 7시에 실행되는 'Morning Briefing' 자동화 작업을 만들어줘. 먼저 내 Google 캘린더를 확인하여 오늘의 핵심 일정과 미팅을 시간순으로 요약해 줘. 다음으로 Gmail의 수신함을 스캔하여 클라이언트나 팀원에게서 온 '긴급' 또는 '중요' 이메일이 있다면 리스트업 해줘. 마지막으로 웹 검색을 통해 'AI 자동화 및

SaaS

트렌드'와 관련된 인기 있는 뉴스 기사 3개를 찾아 인사이트를 요약해 줘. 이 모든 결과물을 잘 포맷팅된 텍스트로 정리하여 내 Slack의 #daily-brief 채널로 즉시 전송해 줘."

이 프롬프트를 한 번 설정해 두면, 에이전트는 사용자가 자는 동안에도 메일을 읽고, 뉴스를 스크랩하여 완벽한 아침 보고서를 만들어 냅니다.

# Codex 구체적 활용사례 37 - 제2장. 영수증 및 청구서 대량 자동 처리 프로세스

제2장. 영수증 및 청구서 대량 자동 처리 프로세스

재무팀이나 1인 기업가에게 수십, 수백 장의 영수증과 청구서를 정리하는 것은 엄청난 시간 낭비입니다. AI 에이전트를 사용하면 폴더에 쌓인 PDF 및 이미지 형식의 영수증을 한 번의 명령으로 완벽하게 분류하고 데이터를 추출할 수 있습니다. 이 워크플로우의 AI가 단일 에이전트로 모든 문서를 순차적으로 읽는 것이 아니라, 문서의 개수만큼 하위 에이전트(Sub-agents)를 병렬로 생성(Spawn)하여 작업을 분산 처리한다는 것입니다. 50장의 영수증이 있다면 50개의 미니 에이전트가 동시에 투입되어 단 몇 분 만에 작업을 완료합니다.

핵심 설정 및 워크플로우:

작업 폴더 구성: 바탕화면에 invoices라는 원본 폴더와 결과를 저장할 빈 CSV 파일(log_invoices.csv), 그리고 처리된 파일을 옮길 processed 폴더를 준비합니다.

추출 스킬(Skill) 정의: AI가 영수증에서 어떤 데이터(발행일, 보낸 사람, 금액 등)를 추출해야 하는지 명시한 skill.md를 미리 작성해 두면 정확도가 크게 늘어납니다.

예제 프롬프트:

"/process_invoices 스킬을 실행해 줘. 현재 invoices 폴더에 있는 모든 PDF 및 이미지 영수증 파일(총 50여 장)을 병렬 하위 에이전트(Sub-agents)를 사용해 스캔해 줘. 각 영수증에서 '청구서 번호(Invoice Number)', '발신자(Sender)', '청구 날짜(Date)', '총 결제 금액(Total Amount)'을 추출하여 log_invoices.csv 파일의 새 행에 정확히 기록해 줘. 데이터 추출이 완료된 파일은 processed 폴더 내에 결제 날짜 기준(YYYY-MM)으로 하위 폴더를 자동 생성하여 이동시켜 줘. 작업이 끝나면 총 청구 금액과 누락된 데이터가 있는 파일 목록을 요약해 줘."

이 지시를 받은 AI는 불과 3~5분 만에 지루한 데이터 입력 작업을 끝내고, CSV 파일과 깔끔하게 폴더별로 정리된 파일 시스템을 제공합니다.

# Codex 구체적 활용사례 37 - 제3장. 웹사이트 및 방문 페이지 (Landing Page) 완벽 복제 및 생성

제3장. 웹사이트 및 방문 페이지(Landing Page) 완벽 복제 및 생성

디자인 사이트(Dribbble 등)의 캡처 이미지나 타사 웹사이트의 URL만으로 완벽하게 작동하는 HTML/React 웹페이지를 코딩하는 워크플로우입니다. AI의 강력한 비전(Vision) 분석 기능과 프런트엔드 코딩 능력을 결합하여, 바닥부터 코드를 짜지 않고도 고품질의 레이아웃을 즉시 얻을 수 있습니다. 이 장의 핵심은 무턱대고 AI에게 디자인을 맡기는 '바이브 코딩'을 피하고, 사전에

design.md

와 같은 자체 디자인 시스템 문서나 템플릿을 제공하여 결과물이 일관된 브랜드 톤을 유지하도록 만드는 것입니다.

핵심 설정 및 워크플로우:

디자인 시스템 구축: 프로젝트 루트 폴더에

design.md

파일을 생성하고 브랜드 컬러(Hex 코드), 폰트(Geist 등), 버튼 레이아웃 규칙을 사전 정의합니다.

시각적 컨텍스트 제공: Dribbble에서 마음에 드는 랜딩 페이지 레이아웃을 캡처하여 프롬프트 창에 드래그 앤 드롭으로 첨부합니다.

예제 프롬프트:

"첨부된 웹사이트 UI 캡처 이미지의 레이아웃과 레이아웃 구조를 완벽하게 복제하여 최신 Next.js 및 Tailwind CSS 기반의 랜딩 페이지 코드를 작성해 줘. 단, 캡처본의 색상과 폰트는 무시하고 이 프로젝트 루트에 있는

design.md

파일의 브랜드 디자인 시스템(버튼 색상, 타이포그래피, 다크 모드 지원 등)을 엄격하게 적용해야 해. 상단에는 네비게이션 바를, 중앙에는 사용자의 이메일을 수집할 수 있는 Tally.so 임베드 폼을 반응형으로 배치해 줘. 코드가 완성되면 Playwright 브라우저를 열어 화면이 제대로 렌더링되는지 스스로 시각적 QA(품질 검사)를 진행해 줘."

AI는 디자인을 해석하고 코드를 작성한 뒤, 내장된 브라우저 제어 권한을 통해 실제 렌더링된 화면을 스크린샷으로 찍어 검증하고 에러가 있다면 스스로 디버깅하여 최종 코드를 완성합니다.

# Codex 구체적 활용사례 37 - 제4장. 풀스택 SaaS 및 웹 애플리케이션 구축

제4장. 풀스택 SaaS 및 웹 애플리케이션 구축

간단한 정적 웹사이트를 넘어, 사용자 인증(Auth), 백엔드 데이터베이스, 결제 시스템 통합이 필요한 복잡한

SaaS

애플리케이션이나 크롬 공룡 게임(Chrome Dino Run) 같은 인터랙티브 브라우저 게임을 개발하는 워크플로우입니다. 복잡한 프로젝트일수록 AI가 코드를 바로 작성하게 하는 대신, plan 모드를 사용하여 데이터베이스 스키마와 아키텍처를 먼저 설계하고 승인받도록 유도하는 것이 중요합니다.

핵심 설정 및 워크플로우:

기술 스택 강제 (

agents.md

): AI가 오래된 방식이나 임의의 라이브러리를 쓰지 못하도록

agents.md

파일에 "항상 Next.js 클라이언트 컴포넌트를 사용하고, 인증은 Clerk, DB는

Supabase

, UI는 Tailwind CSS를 사용하라"고 명시합니다.

의존성 설치 및 스캐폴딩: AI에게 터미널 명령어를 실행할 권한을 부여하여 패키지를 자동 설치하게 합니다.

예제 프롬프트 (SaaS 및 브라우저 게임 통합 예시):

"현재 빈 디렉토리에 사용자 인증이 가능한 풀스택

SaaS

애플리케이션을 구축할 거야.

/plan

모드로 시작하여 구현 계획을 먼저 세워 줘. 요구사항은 다음과 같아: 1) Clerk을 사용한 소셜 로그인 시스템 구현. 2)

Supabase

를 연결하여 사용자별 최고 점수를 저장할 데이터베이스 스키마 작성. 3) 로그인한 사용자만 접근 가능한 대시보드 내에 '크롬 공룡 달리기 게임(Dino Run)'의 클론 버전을 HTML/Canvas 기반으로 코딩하여 플레이할 수 있게 구현. 4) 게임 오버 시

Supabase

에 점수가 기록되어야 해. 내

agents.md

에 명시된 규칙에 따라 Next.js 프로젝트 스캐폴딩(초기 설정)부터 시작하고, 각 마일스톤이 끝날 때마다 작동 여부를 테스트 코드로 검증해 줘."

AI는 계획을 수립한 후, 사용자 인증 백엔드를 연동하고, 물리 엔진과 충돌 감지가 포함된 자바스크립트 기반의 게임 루프를 차례대로 구현하여 풀스택 앱을 완성합니다.

# Codex 구체적 활용사례 37 - 제5장. 모바일 앱(iOS/Android) 기획 및 개발

제5장. 모바일 앱(iOS/Android) 기획 및 개발

코딩 지식이 전혀 없어도 아이디어 스케치와 기획 문서만으로 완전하게 작동하는 모바일 애플리케이션(예: 음악 플레이어, 도파민 차단 앱 등)을 만들고, 로컬 Mac의 Xcode 시뮬레이터에서 직접 구동해보는 혁신적인 워크플로우입니다.

Codex

의 강력한

/goal

커맨드를 사용하면, 수십 개의 태스크가 담긴 로드맵 문서를 던져주고 AI가 수 시간 동안 사람의 개입 없이 자율적으로 로드맵 전체를 구현하도록 지시할 수 있습니다.

핵심 설정 및 워크플로우:

로드맵 및 PRD 준비: 프로젝트 루트에 docs/prd.md(제품 요구사항 문서)와 product_roadmap.md(태스크 리스트)를 상세히 작성해 둡니다.

자율 주행(

/goal

) 실행: AI에게 목표(Goal)를 부여하여 모든 태스크 리스트의 체크박스가 완료될 때까지 스스로 코딩하고 디버깅하는 무한 루프를 실행합니다.

예제 프롬프트:

"

/goal

docs/prd.md에 정의된 iOS 모바일 앱 'Math Block(수학 블록)'의 전체 코드를 Swift로 작성해 줘. 이 앱은 인스타그램 등 SNS 앱을 열기 전에 사용자가 수학 문제를 풀어야만 스크롤을 허용하는 도파민 차단 앱이야. 작업은 product_roadmap.md에 있는 60개의 태스크 리스트를 순서대로 따라가며 진행하고, 각 마일스톤(쉬움/보통/어려움 난이도 선택, UI 렌더링 등)이 끝날 때마다 로드맵 문서에 완료(X) 표시를 해 줘. Medium(보통) 이상의 난이도는 RevenueCat을 연동하여 월 5달러의 페이월(Paywall) 결제 화면으로 넘어가도록 구현해 줘. 모든 태스크가 완료되고 Xcode 빌드 에러가 0개가 될 때까지 멈추지 말고 자율적으로 작업(Auto-mode)을 진행해."

이 명령어를 받은 AI는 프로젝트를 스스로 설정하고, 수천 줄의 코드를 작성하며 발생한 버그를 스스로 수정합니다. 약 1~2시간 후 코딩이 완료되면 사용자는 터미널에서 로컬 Xcode를 열고 시뮬레이터를 띄워 내 폰 화면에서 작동하는 완성된 모바일 앱을 즉시 확인할 수 있습니다.

# Codex 구체적 활용사례 37 - 제6장. 크롬 익스텐션(Chrome Extension) 자체 제작: 5분 만에 만드는 나만의 생산성 도구

제6장. 크롬 익스텐션(Chrome Extension) 자체 제작: 5분 만에 만드는 나만의 생산성 도구

우리가 매일 업무를 하면서 사용하는 웹 브라우저, 구글 크롬(Chrome)은 수많은 확장 프로그램(Extension)을 통해 기능을 무한대로 확장할 수 있습니다. 과거에는 이러한 크롬 확장 프로그램을 하나 만들려면 자바스크립트(JavaScript), HTML, CSS는 물론 크롬의 복잡한 매니페스트(Manifest) 권한 시스템까지 이해해야 하는 높은 진입 장벽이 있었습니다. 하지만 이제는

Codex

나

Claude Code

같은 AI 에이전트를 쓰면, 코딩 지식이 전혀 없는 초보자라도 단 5분 만에 자신만의 크롬 확장 프로그램을 개발하고 브라우저에 직접 설치할 수 있습니다.

이 장에서는 웹페이지의 내용을 복사해서 마크다운(Markdown)으로 변환한 뒤, 깔끔한 PDF 파일로 다운로드해 주는 확장 프로그램을 만드는 사례를 실습해 보겠습니다. 이 기능은 웹의 방대한 자료를 AI에게 학습시킬 컨텍스트(Context)로 저장하거나 오프라인 문서를 스크랩할 때 유용합니다,.

1. 작업 환경 준비하기

확장 프로그램을 만들기 위해 거창한 개발 도구가 필요하지 않습니다. 바탕화면에 빈 폴더를 하나 만들고 시작합니다.

바탕화면에 Chrome_Extension이라는 이름으로 빈 폴더를 생성합니다.

터미널(또는 명령 프롬프트)을 열고 해당 폴더로 이동한 뒤,

codex

명령어를 입력하여 AI 에이전트를 실행합니다. 이제 빈 캔버스가 준비되었습니다.

2. 에이전트에게 지시하기 (예제 프롬프트)

프롬프트를 작성할 때는 '내가 무엇을 원하는지', 그리고 '초보자이므로 설치 방법까지 알려달라'고 구체적으로 지시하는 것이 핵심입니다.

예제 프롬프트 입력: "현재 열려 있는 웹페이지의 내용을 마크다운(Markdown) 포맷으로 변환한 다음, 내 컴퓨터의 다운로드 폴더에 저장해 주는 구글 크롬 확장 프로그램(Chrome Extension)을 만들어줘. 나는 코딩 초보자이니까 이 확장 프로그램을 크롬에 어떻게 배포하고 설치해서 내가 직접 사용할 수 있는지 그 전체 과정도 차근차근 단계별로 안내해 줘야 해. 복잡하게 만들 필요 없이 심플하게 작동하도록 구현해 줘."

3. AI의 작업 과정 및 결과 확인

명령을 내리면 AI는 즉시 이 작업이 구현 가능하다고 답변하며, 확장 프로그램에 필요한 manifest.json 파일부터 팝업 화면을 위한 HTML, 마크다운 변환을 담당하는 JavaScript 코드를 스스로 계획하고 작성하기 시작합니다.

이 과정에서 1~2분 정도의 시간이 소요되며, AI가 폴더 내에 4~5개의 파일을 자동으로 생성하는 것을 터미널 창을 통해 실시간으로 볼 수 있습니다. 작업이 완료되면 AI는 "확장 프로그램이 준비되었습니다. 이제 다음과 같이 설치하세요"라는 친절한 가이드를 제공합니다.

4. 크롬 브라우저에 직접 설치 및 배포하기 (초보자 가이드)

AI가 만들어준 파일을 크롬 브라우저에 등록하는 과정은 클릭 몇 번이면 끝납니다. 다음 단계에 따라 설치해 보세요.

크롬 브라우저를 열고 주소창에 chrome://extensions/를 입력하여 확장 프로그램 관리 페이지로 이동합니다.

페이지 우측 상단에 있는 '개발자 모드(Developer mode)' 토글 버튼을 클릭하여 활성화합니다.

좌측 상단에 새롭게 나타난 '압축해제된 확장 프로그램을 로드합니다(Load unpacked)' 버튼을 클릭합니다.

방금 AI가 파일들을 생성해 둔 바탕화면의 Chrome_Extension 폴더를 선택합니다.

설치가 완료되었습니다! 브라우저 우측 상단의 퍼즐 모양 아이콘을 눌러 방금 만든 확장 프로그램을 고정(Pin)해 보세요,.

5. 작동 테스트 및 응용

이제 복잡한 내용이 담긴 뉴스 기사나 기술 문서 페이지로 이동하여, 방금 만든 확장 프로그램 아이콘을 클릭해 봅니다. 클릭 한 번에 웹페이지의 불필요한 광고나 메뉴는 사라지고, 텍스트와 핵심 구조만 완벽한 마크다운 형태의 파일로 다운로드 폴더에 저장되는 것을 확인할 수 있습니다. 이 마크다운 포맷은 AI에게 문서를 먹일 때 이해하기 쉬운 최고의 포맷입니다.

실전 팁: 이 뼈대를 바탕으로 무궁무진한 응용이 가능합니다. AI에게 "링크드인 프로필을 스크랩해서 내

CRM

으로 바로 넘겨주는 확장 프로그램을 만들어줘" 혹은 "지메일에서 특정 내용을 클릭 한 번에 구글 드라이브로 저장하는 확장 프로그램으로 수정해 줘"라고 추가 프롬프트를 입력해 보세요. 비싼 돈을 주고 유료 생산성 툴을 결제하지 않고도, 나만의 맞춤형 도구를 무한대로 찍어낼 수 있습니다.

# Codex 구체적 활용사례 37 - 제7장. 지저분한 CSV 데이터 자동 정제 및 전처리: 정규식 없이 자연어로 데이터 다루기

제7장. 지저분한 CSV 데이터 자동 정제 및 전처리: 정규식 없이 자연어로 데이터 다루기

데이터 분석가, 마케터, 영업 관리자라면 누구나 공감하는 최악의 업무가 있습니다. 바로 동료나 외부에서 전달받은 규칙 없고 지저분한 엑셀(CSV) 데이터를 하나하나 수작업으로 정리하는 일입니다. 날짜 형식이 2024-01-01과 Jan 1st, 01/01로 뒤섞여 있거나, 회사 이름의 대소문자가 제멋대로이고, 전화번호 포맷이 전부 다르며, 텍스트로 적힌 비고(Notes) 란에 매출 금액이 숨어 있는 경우 등입니다,,.

과거에는 이를 해결하기 위해 파이썬(Python)의 판다스(Pandas) 라이브러리를 다루거나, 머리 아픈 정규표현식(Regex)을 작성해야만 했습니다. 하지만 이제는 자연어로 AI에게 "어떻게 정리해달라"고 지시하기만 하면, AI가 자동으로 전처리 스크립트를 작성하여 몇 분 만에 완벽하게 정제된 파일을 만들어 냅니다,,.

1. 작업 환경 및 문제 상황 정의

바탕화면의 프로젝트 폴더에 지저분한 데이터가 담긴 raw_contacts.csv 파일이 있다고 가정해 보겠습니다. 이 파일이 가진 문제는 다음과 같습니다.

전화번호: 1234567890, 123-456-7890, (123) 456 7890 등 중구난방입니다.

회사명: Apple Inc., apple llc, APPLE 등 대소문자와 띄어쓰기가 엉망입니다,.

비고(Notes): "이번 딜의 예산은 약 $45,000 정도 될 것 같습니다" 처럼 텍스트 사이에 숫자가 섞여 있어 금액만 추출해야 합니다.

2. 에이전트에게 지시하기 (예제 프롬프트)

초보자분들은 복잡한 파이썬 코드를 짤 필요가 없습니다. 파일의 문제점과 내가 원하는 최종 결과물의 형태만 구체적으로 나열하면 됩니다.

예제 프롬프트 입력: "현재 폴더에 있는 raw_contacts.csv 파일은 너무 지저분한 연락처 데이터야. 이 파일을 분석해서 다음 규칙에 맞게 데이터를 완벽하게 정제해 줘.

전화번호 열: 포맷이 전부 제멋대로인데, 모두 정규식(Regex)을 사용해서 (XXX) XXX-XXXX 형식으로 통일해 줘.

회사명 열: 불규칙한 대소문자를 정리(Title case)하고, 이름 끝에 붙은 마침표나 쉼표 같은 불필요한 구두점을 제거해 줘.

비고(Notes) 열: 텍스트 중간에 달러($)나 금액이 섞여 있어. 여기서 숫자(금액)만 추출해서 'estimated_deal_value(예상 거래액)'라는 새로운 열을 만들고, 숫자가 여러 개면 큰 숫자만 남겨줘. 금액이 없으면 Null(빈칸)로 처리해 줘. 이 모든 정제 작업을 파이썬 스크립트를 작성해서 자동으로 수행하고, 결과물은 clean_raw_contacts.csv라는 새 파일로 저장해 줘.",

3. AI의 작업 과정 및 결과 확인

프롬프트를 입력하면 AI는 데이터를 읽어 들이고, 스스로 파이썬 코드를 작성하기 시작합니다. 정규표현식(Regex)을 이용해 복잡한 전화번호 패턴을 파악하고, 판다스(Pandas) 라이브러리를 써 데이터를 변환하는 100줄이 넘는 코드를 작성한 뒤 직접 실행까지 완료합니다.

약 1~2분 뒤, 에이전트는 작업을 완료했다는 메시지와 함께 폴더에 clean_raw_contacts.csv 파일을 생성해 둡니다. 엑셀로 이 새 파일을 열어보면, 모든 전화번호가 (XXX) XXX-XXXX 형태로 깔끔하게 통일되어 있고, 지저분했던 회사 이름이 정렬되었으며, 긴 텍스트 문장에서 45000이라는 숫자만 정확히 추출되어 새로운 열에 정리된 것을 볼 수 있습니다.

초보자를 위한 조언: 정규표현식(Regex)은 괄호를 넣을지 말지 고민하며 한 시간을 훌쩍 낭비하게 만드는 까다로운 문법입니다. AI에게 이렇게 자연어로 예시(Before/After)를 주고 맡기면, 오타 없이 완벽한 코드를 작성해 냅니다. 데이터 전처리에 들이는 시간을 5시간에서 단 5분으로 단축하는 기적을 경험해 보세요,.

# Codex 구체적 활용사례 37 - 제8장. 복수의 스프레드시트 데이터 병합 및 피벗 테이블 생성: 맥락을 이해하는 AI 데이터 결합

제8장. 복수의 스프레드시트 데이터 병합 및 피벗 테이블 생성: 맥락을 이해하는 AI 데이터 결합

비즈니스 현장에서는 데이터가 한 곳에 예쁘게 모여 있는 경우가 드뭅니다. 예를 들어, A팀은

CRM

소프트웨어에서 다운로드한 '고객 데이터'를 쓰고, B팀은 '청구서 데이터'를, C팀은 '마케팅 웨비나 참여 데이터'를 엑셀로 관리합니다. 상사는 당신에게 "이 3개의 파일을 하나로 합쳐서 고객별 청구 현황과 마케팅 반응도를 한눈에 볼 수 있는 통합 보고서를 만들어라"고 지시합니다,.

일반적인 엑셀의 VLOOKUP이나 INDEX/MATCH 함수를 쓰려면 '고객 ID'와 같은 완벽하게 일치하는 식별자(Key) 키값이 있어야 합니다. 하지만 마케팅팀 파일에는 고객 ID 대신 '회사 이름'만 적혀 있고, 그마저도

CRM

에는 Apple Inc.라고 적혀 있는데 마케팅 파일에는 Apple이라고 적혀 있는 등 이름이 정확히 일치하지 않습니다. 예전 같으면 일일이 눈으로 확인하며 수작업(Fuzzy matching)을 해야 했지만, AI는 문맥과 의미를 이해하여 스마트하게 병합(Join)해 냅니다,.

1. 작업 환경 준비

프로젝트 폴더 내에 3개의 파일이 준비되어 있다고 가정합니다.

crm

_data.csv (고객ID, 회사명, 계약 규모, 담당 영업사원)

billing_data.csv (고객ID, 연체된 청구서 금액 유무)

marketing_engagement.csv (회사명, 웨비나 참석 여부, NPS 만족도 점수)

2. 에이전트에게 데이터 병합 지시하기 (예제 프롬프트)

AI에게 파일 간의 불일치성을 솔직하게 설명하고 문맥적 병합을 요구하는 것이 포인트입니다.

예제 프롬프트 입력 (데이터 병합): "현재 폴더에

crm

_data.csv, billing_data.csv, marketing_engagement.csv 3개의 파일이 있어. 이 파일들을 하나로 병합해서 각 고객의 '계약 가치', '청구서 연체 여부', '마케팅 NPS 점수'를 보여주는 하나의 뷰로 만들어야 해. 주의할 점은, 마케팅 파일은 '고객 ID'를 사용하지 않고 '회사 이름(Company Name)'을 식별자로 쓰고 있어. 게다가

CRM

파일과 마케팅 파일의 회사 이름이 Apple Inc.와 Apple처럼 완벽하게 똑같지 않을 수 있어. 네가 데이터의 문맥을 이해해서 유사한 회사 이름끼리 스마트하게 병합해 줘. 병합이 끝나면 customer_combined_view.csv로 저장해 줘."

프롬프트를 전송하면 AI는 정확 일치가 아닌 유사도 분석(Fuzzy matching)을 쓴 파이썬 스크립트를 작성하여 단 1분 만에 3개의 파일을 완벽하게 하나로 합쳐냅니다.

3. 병합된 데이터로 피벗 테이블 생성 및 성과 분석하기

데이터가 하나로 합쳐졌다면, 이제 경영진이 좋아할 만한 요약 피벗 테이블(Pivot Table)과 인사이트를 뽑아낼 차례입니다. 동일한 채팅창에서 이어서 질문해 봅니다.

예제 프롬프트 입력 (피벗 테이블 및 분석): "정말 잘했어! 이제 방금 만든 customer_combined_view.csv 데이터를 바탕으로 엑셀의 피벗 테이블 같은 형태를 만들어줘. 행(Row)에는 '담당 영업 사원'을 넣고, 열(Column)에는 '제품 종류'를 넣은 뒤, 값(Value)에는 '총 계약 가치(Total Revenue)'의 합계를 배치해서 요약해 줘. 추가로 이 표를 분석해서 이번 달에 목표 할당량을 채우지 못하고 실적이 뒤처지는 영업 사원이 누구인지, 그 갭(Gap)이 얼마인지 분석해서 텍스트로 요약해 줘.",

4. AI의 분석 결과와 실무 적용

AI는 엑셀에서 마우스 클릭으로 만들던 피벗 테이블을 텍스트와 코드 형태(Pandas pivot_table 등)로 순식간에 출력하여 화면에 보여줍니다,. 표에는 제임스, 안젤라 등 영업 사원별로 각 제품 카테고리(하드웨어, 소프트웨어 등)에서 얼마의 매출을 올렸는지 깔끔하게 요약되어 나타납니다.

더 나아가 프롬프트에서 요구한 대로 데이터를 깊게 분석한 인사이트를 제공합니다. "분석 결과: 오직 한 명의 영업 사원(마커스 존슨)만이 목표치에 미달했습니다. 마커스의 목표는 960K였으나 현재 822K 달성으로 137K(약 15%)가 부족한 상태입니다. 제임스 윌슨과 안젤라는 오직 소프트웨어 라이선스에만 집중하여 좋은 성과를 내고 있습니다.",,.

이 워크플로우를 쓰면 초보자라도 코딩 없이 고급 데이터 엔지니어와 데이터 분석가를 고용한 것과 같은 효과를 낼 수 있습니다. 이메일을 켜고 상사에게 "마커스 존슨의 매출이 15% 부족하니 조치가 필요합니다"라는 내용의 보고서를 즉시 전송할 수 있는 생산성을 확보하게 됩니다.

# Codex 구체적 활용사례 37 - 제9장. 보고서 내 이상치(Outlier) 및 누락 데이터 사전 감지: 경영진 회의 직전, AI로 완벽한 데이터 감사(Audit) 진행하기

제9장. 보고서 내 이상치(Outlier) 및 누락 데이터 사전 감지: 경영진 회의 직전, AI로 완벽한 데이터 감사(Audit) 진행하기

초보자를 위한 개념 이해: 왜 AI 데이터 감사가 필요한가? 데이터 분석가나 실무자라면 누구나 한 번쯤 겪어본 아찔한 상황이 있습니다. 경영진 회의나 중요한 발표를 위해 엑셀 보고서를 준비하고 전송 버튼을 눌렀는데, 회의 시작 5분 만에 누군가가 "잠깐만요, 이 매출 숫자가 이상한데요?"라고 지적하는 순간입니다. 터무니없이 높은 매출 값, 음수로 입력된 수익, 혹은 누락된 데이터가 포함된 채로 보고서가 올라가면 데이터의 신뢰성은 바닥으로 떨어집니다.

과거에는 이를 방지하기 위해 수천, 수만 줄의 엑셀 데이터를 눈으로 훑어보거나, 복잡한 필터와 수식을 걸어 이상치를 찾아내야 했습니다. 하지만 이제는

Claude Code

나

Codex

와 같은 AI 에이전트를 우리의 '두 번째 눈(Second set of eyes)'으로 쓸 수 있습니다. AI에게 "이 데이터에서 이상한 점을 찾아줘"라고 자연어로 지시하기만 하면, AI가 논리적 오류를 사전에 감지하고 보고해 줍니다.

실무 사용 사례: Q1 매출 보고서(CSV) 긴급 감사 바탕화면에 Q1_revenue_report.csv라는 파일이 있다고 가정해 봅시다. 이 파일에는 고객명, 지역, 제품, 매출, 판매 담당자, 마진율 등의 데이터가 수만 줄 들어있습니다. 이 파일에는 사람이 눈으로 찾기 힘든 6가지의 치명적인 오류가 숨어있습니다.

일반적인 매출이 1만 달러 수준인데, 오타로 인해 '980,000달러(약 13억 원)'가 입력된 행이 존재합니다.

수익(Revenue) 열에 마이너스(-) 값이 입력되어 있습니다.

1분기(Q1, 1월~3월) 보고서인데, 날짜가 '2024년 12월 15일'로 잘못 입력된 행이 섞여 있습니다.

마진율(Margin)이 누락된 빈칸(Null)이 존재합니다.

고객 ID가 중복으로 입력된 행이 존재합니다.

초보자를 위한 구체적인 AI 명령어 (프롬프트) 이러한 상황에서 AI 에이전트에게 내릴 프롬프트는 구체적이고 배경 설명(Context)이 풍부해야 합니다. AI에게 상황의 시급성과 어떤 형태의 오류를 찾고 싶은지 명확히 지시하세요.

예제 프롬프트 입력: "현재 폴더에 Q1_revenue_report.csv 파일이 있어. 나는 1시간 뒤에 경영진 매니저에게 이 Q1(1분기) 매출 보고서를 발표해야 해. 회의에 들어가기 직전인데, 내가 이 데이터를 발표하기 전에 이 데이터 안에 의심스럽거나 잘못된 논리적 오류가 없는지 감사(Audit)해 줄 수 있어?

다음 항목들을 중점적으로 스캔해 줘:

터무니없이 높거나 낮은 이상치(Outlier) 값 (예: 비정상적인 매출액)

누락된 데이터(Missing values)나 빈칸

중복된 행(Duplicates)

Q1(1분기, 1월~3월) 데이터가 아닌 잘못된 날짜 형식

기타 비즈니스 논리에 맞지 않는 모든 사항

문제를 찾으면 어떤 행(Row)에서 오류가 발생했는지, 왜 이것이 문제라고 판단했는지 목록으로 요약해서 알려줘."

AI의 작업 과정 및 결과 확인 이 프롬프트를 입력하면 AI 에이전트는 즉시 CSV 파일을 읽어 들이고 데이터를 스캔하기 시작합니다. 불과 1~2분 만에 AI는 다음과 같은 결과를 화면에 출력합니다.

치명적 오류 1 (이상치): "매출(Revenue) 열에서 $980,000라는 이상치를 발견했습니다. 다른 평균 매출이 $5,000~$15,000 사이인 것을 감안할 때 이는 심각한 입력 오류로 보입니다."

치명적 오류 2 (논리 오류): "음수(-)로 표기된 매출 값을 발견했습니다. 일반적인 판매 데이터에서 매출은 음수가 될 수 없습니다."

치명적 오류 3 (잘못된 날짜): "회계연도 기준 1분기(Q1) 보고서인데, 12월 15일(12/15)로 기록된 날짜가 포함되어 있습니다. 이는 Q1 데이터 셋에 포함되어서는 안 됩니다."

경고 (누락 데이터): "12번째 행에서 마진율(Margin percentage) 데이터가 누락되어 비어 있습니다."

초보자를 위한 실무 꿀팁 (프롬프트 반복의 중요성): AI는 훌륭하지만 한 번의 지시로 모든 것을 완벽하게 찾지 못할 때도 있습니다. 예를 들어, 위 결과에서 AI가 '중복된 데이터(Duplicates)'를 찾아내지 못하고 넘어갔다면, 실망하지 말고 AI에게 다시 추궁하듯 질문해야 합니다.

추가 프롬프트 입력: "정말 고마워! 그런데 네가 보내준 결과를 보니 '중복된 데이터(Duplicates)'에 대한 언급이 없네. 중복된 값을 놓친 것 같은데 다시 한번 꼼꼼하게 중복 데이터나 다른 불일치 요소가 없는지 확인해 줄래?"

이렇게 다시 질문하면 AI는 "죄송합니다. 다시 깊게 스캔해 본 결과, 고객 ID가 동일한 중복 결제 건을 발견했습니다. 11번 행에서 유닛(Units) 판매량이 일반적인 20~50개가 아닌 500개로 입력된 또 다른 이상치를 추가로 발견했습니다."라며 놓친 부분까지 완벽하게 잡아냅니다. 이렇게 AI를 통해 회의 5분 전, 치명적인 오류를 사전에 수정하여 완벽한 보고서를 제출할 수 있습니다.

# Codex 구체적 활용사례 37 - 제10장. SQL 쿼리 디버깅, 속도 최적화 및 뷰(View) 자동 변환: 느리고 복잡한 쿼리를 AI로 완벽하게 다듬기

제10장. SQL 쿼리 디버깅, 속도 최적화 및 뷰(View) 자동 변환: 느리고 복잡한 쿼리를 AI로 완벽하게 다듬기

초보자를 위한 개념 이해: 실행은 되지만 '나쁜'

SQL

쿼리 데이터를 다루다 보면

SQL

쿼리와 마주하게 됩니다. 초보자들이

SQL

을 작성할 때 겪는 큰 좌절 중 하나는, "구문 오류(Syntax Error)도 없고 실행도 되는데, 결과 값이 완전히 틀리게 나오거나 쿼리가 실행되는 데 무려 4분 이상이 걸리는 상황"입니다. 어디서 잘못되었는지 찾으려고 수십 줄의 JOIN과 SUBQUERY를 들여다보는 것은 큰 고통입니다. 매주 월요일마다 실행해야 하는 쿼리에 날짜가 '2024-01-01'처럼 하드코딩되어 있어서 매주 수작업으로 날짜를 수정해야 하는 귀찮은 작업도 존재합니다. AI 에이전트를 쓰면 이러한

SQL

논리 오류를 디버깅하고, 속도를 최적화하며, 깔끔한 자동화 뷰(View)로 변환하는 작업을 순식간에 처리할 수 있습니다.

실무 사용 사례 1: 4분이 넘게 걸리는 느린 쿼리 최적화 (Optimization) 회사 데이터베이스에서 데이터를 추출하는 slow_query.

sql

파일이 있다고 가정해 보겠습니다. 결과는 정확하게 나오지만, 비효율적으로 작성되어 한 번 실행할 때마다 4분이 넘게 걸립니다.

예제 프롬프트 입력: "현재 폴더에 있는 slow_query.

sql

파일의 쿼리는 결과는 맞게 나오지만 실행하는 데 4분 이상이 걸려. 쿼리 작성에 분명 비효율적인 부분이 있는 것 같아. 이 쿼리를 분석해서 도대체 무엇이 속도를 느리게 만들고 있는지 원인을 설명해 줘. 그리고 쿼리 실행 속도가 훨씬 빨라지도록 완벽하게 최적화된 새로운 쿼리로 다시 작성해 줘. 마지막으로 최적화된 쿼리가 기존보다 얼마나 빨라질지 예상 속도 개선율도 알려줘."

AI의 작업 과정 및 결과: 프롬프트를 입력하면 AI는 약 1~2분 정도 코드를 분석한 뒤 다음과 같은 분석 리포트와 수정된 코드를 반환합니다.

병목 현상 원인: "분석 결과, 속도 저하의 주범은 EXISTS 절 내부에 작성된 '상관 서브쿼리(Correlated Subquery)' 때문입니다. 이 구조는 메인 쿼리의 각 행마다 서브쿼리를 반복 실행하게 만들어, 데이터가 커질수록 기하급수적으로 속도가 느려집니다(Cubic blowup). 이것이 4분이나 걸린 이유입니다."

해결책 및 최적화된 쿼리: AI는 불필요한 DISTINCT와 무거운 서브쿼리를 제거하고, 훨씬 깔끔한 JOIN 구조로 재작성된 최적화

SQL

코드를 제공합니다.

성능 예상: "실제 프로덕션 데이터 셋을 기준으로 이 최적화된 쿼리는 기존보다 약 20배에서 50배 더 빠르게 실행될 것입니다."

실무 사용 사례 2: 하드코딩된 쿼리를 동적 뷰(View)로 변환하기 매주 월요일마다 '주간 활성 사용자(WAU)', '매출 대비 목표 달성률' 등을 뽑아내는 3개의

SQL

파일이 있습니다. 이 쿼리들에는 날짜가 수동으로 하드코딩되어 있어 매주 수정해야 합니다. 이를 영구적으로 사용할 수 있는 자동화된 View로 만들고 싶습니다.

예제 프롬프트 입력: "여기 매주 월요일 아침마다 내가 수동으로 실행하는 3개의

SQL

쿼리 파일이 있어. 이 쿼리들은 작동은 하지만 너무 지저분해.

이 3개의 쿼리 각각을 깔끔한

SQL

뷰(View) 테이블을 생성하는 코드(CREATE VIEW)로 변환해 줘. 중요한 건, 쿼리 안에 하드코딩된 특정 날짜(예: '2024-03-01')들을 모두 제거하고, CURRENT_DATE와 같은 동적 날짜 함수를 사용해서 매주 월요일에 실행해도 알아서 지난 7일의 데이터를 가져오도록 수정해 줘. 코드가 무슨 역할을 하는지 주석도 꼼꼼히 달아줘."

AI의 작업 과정 및 결과: AI는 3개의 파일을 읽고 문제점을 파악한 뒤, 뷰 생성 스크립트를 작성합니다. 하드코딩된 날짜를 식별하여 WHERE order_date 〉 CURRENT_DATE - 7과 같은 동적 함수로 완벽하게 교체합니다. 이제 초보자도 복잡한 쿼리창을 열어 날짜를 수정하지 않고도, 그저 SELECT * FROM weekly_active_users 한 줄만 입력하면 항상 최신의 올바른 데이터를 뽑아볼 수 있는 강력한 자동화 환경을 구축하게 됩니다.

# Codex 구체적 활용사례 37 - 제11장. 레거시 코드 및 데이터베이스 스키마 자동 문서화(Data Dictionary): 암호 같은 과거의 유산을 AI로 해독하기

제11장. 레거시 코드 및 데이터베이스 스키마 자동 문서화(Data Dictionary): 암호 같은 과거의 유산을 AI로 해독하기

초보자를 위한 개념 이해: 주석 없는 코드와 이름 없는 데이터베이스 새로운 부서에 배치되거나 전임자가 퇴사하고 남긴 업무를 물려받았을 때, 당혹스러운 순간은 '문서화가 전혀 되어있지 않은 과거의 산출물(레거시)'을 마주할 때입니다. 파이썬 스크립트를 열어보니 주석은 단 한 줄도 없고 변수 이름은 x, y, temp_data로 되어 있습니다. 데이터베이스를 열어보니 컬럼 이름이 col_mrr_24, sub_tier, acq_src_1처럼 암호같이 적혀 있어 도대체 어떤 데이터를 의미하는지 알 수가 없습니다. 과거에는 이를 파악하기 위해 기존 코드를 한 줄씩 디버깅하며 유추하거나 다른 부서원들을 찾아다니며 물어봐야 했습니다. 하지만 AI를 쓰면 이 암호 같은 코드와 데이터베이스 스키마를 완벽한 설명서(데이터 사전)로 자동 변환할 수 있습니다.

실무 사용 사례 1: 이름 없는 과거의 파이썬 스크립트 해독 및 리팩토링 아무런 주석 없이 방치된 undocumented_script.py 파일이 프로젝트 폴더에 덩그러니 놓여 있습니다.

예제 프롬프트 입력: "현재 폴더에 undocumented_script.py라는 파일이 있어. 몇 달 전에 작성된 스크립트인데, 주석도 없고 변수명도 엉망이라 솔직히 이 코드가 정확히 무슨 역할을 하는지 전혀 기억이 안 나.

이 코드를 처음부터 끝까지 읽고, 이 스크립트가 도대체 무슨 목적을 가진 코드인지 초보자도 이해할 수 있는 쉬운 일상 용어(Plain English)로 설명해 줘. 그리고 x, df1 같이 의미 없는 변수 이름들을 이 코드가 수행하는 역할에 맞게 직관적이고 설명적인 이름으로 전부 다 수정(Rename)해 줘. 마지막으로 코드의 각 기능 블록마다 적절한 주석을 달고, 맨 위에는 이 스크립트의 사용법을 설명하는 포괄적인 독스트링(Docstring)을 추가해서 당장 실무 프로덕션에 투입할 수 있는 깔끔한 코드로 다시 저장해 줘."

AI의 작업 과정 및 결과: AI는 즉각 코드를 분석하여 터미널 창에 "이 스크립트는 1단계부터 6단계까지 사용자 데이터를 불러와 매출을 집계하는 파이썬 코드입니다"라고 쉬운 말로 요약해 줍니다. 그런 다음, 알 수 없던 df1 변수를 monthly_revenue_data로 바꾸고, 코드 곳곳에 친절한 주석과 문서화 블록을 삽입한 완벽히 새로운 파이썬 코드를 완성하여 저장합니다. 죽어있던 레거시 코드가 누구나 이해할 수 있는 현대적인 코드로 부활하는 순간입니다.

실무 사용 사례 2: 암호 같은 데이터베이스 스키마를 완벽한 데이터 사전(Data Dictionary)으로 문서화하기 데이터 엔지니어가 데이터베이스 스키마(테이블 구조) 덤프 파일을 하나 던져주고 갔습니다. 테이블과 컬럼 이름들이 축약되어 있어 의미를 알 수 없습니다. 데이터 분석가로서 이를 사용하려면 각 컬럼이 무엇을 의미하는지 정리된 '데이터 사전(Data Dictionary)'이 필수적입니다.

예제 프롬프트 입력: "엔지니어에게 받은 데이터베이스 스키마 덤프 파일(schema.

sql

)이 폴더에 있어. 여기에는 암호같이 축약된 컬럼 이름들을 가진 수많은 테이블이 존재해.

이 스키마 파일 전체를 분석해서, 나를 위한 완벽한 '데이터 사전(Data Dictionary)'을 마크다운(Markdown) 표 형식으로 작성해 줘. 중요한 점: 나는 이 데이터베이스가 '

SaaS

(서비스형 소프트웨어) 비즈니스의 데이터 분석용 데이터베이스'라고 가정하고 있어. 이 비즈니스 배경지식을 바탕으로, 암호 같은 컬럼 이름들이 실제 비즈니스에서 무엇을 의미하는지 문맥에 맞게 논리적으로 유추(Inference)해서 표에 설명을 달아줘."

AI의 작업 과정 및 결과: 프롬프트에서 "이것은

SaaS

비즈니스의 데이터베이스다"라고 비즈니스 컨텍스트(문맥)를 지정해 준 것이 신의 한 수입니다. 이 정보를 바탕으로 AI는 통찰력을 발휘하여 데이터 사전을 구축합니다.

AI는 MRR이라는 컬럼을 발견하고,

SaaS

비즈니스 문맥을 써서 이를 그저 문자가 아닌 "월간 반복 매출(Monthly Recurring Revenue):

SaaS

비즈니스의 핵심 지표"라고 완벽하게 해독하여 문서화합니다.

acq_source라는 컬럼은 "고객 획득 출처(Customer Acquisition Source)"로 유추하여 설명을 적어줍니다.

AI는 discount_applied 컬럼을 보면서, *"참고: 이 스키마만으로는 MRR이 할인 적용 전(Pre-discount) 금액인지 적용 후(Post-discount) 금액인지 모호합니다. 데이터 엔지니어에게 확인이 필요합니다"*라는 예리한 비즈니스 팁까지 마크다운 문서 최하단에 덧붙여 줍니다.

이렇게 AI를 쓰면, 온종일 엔지니어를 쫓아다니며 질문해야 만들 수 있었던 데이터 사전을 단 2~3분 만에 아름다운 마크다운 표 형식으로 완성하여 즉시 실무에 투입할 수 있게 됩니다.

# Codex 구체적 활용사례 37 - 제12장. 경쟁사 웹사이트, SEO, 가격 정책 및 리뷰 스크래핑 분석: 우리 비즈니스의 약점과 기회 한눈에 파악하기

제12장. 경쟁사 웹사이트, SEO, 가격 정책 및 리뷰 스크래핑 분석: 우리 비즈니스의 약점과 기회 한눈에 파악하기

초보자를 위한 개념 이해: 왜 경쟁사 분석을 AI에게 맡겨야 할까? 새로운 비즈니스를 시작하거나 경쟁이 치열한 시장(예: 조경업체, 웹 디자인 에이전시 등)에 진입할 때, 경쟁사가 어떻게 트래픽을 얻고 있는지, 가격은 어떻게 설정했는지, 고객 리뷰는 어떤지 분석하는 것은 필수입니다. 하지만 일일이 수십 개의 경쟁사 웹사이트를 돌아다니며 데이터를 수집하고 엑셀에 정리하는 작업은 수주에서 수 개월이 걸리는 고된 노동입니다. 이제 AI 에이전트(

Claude Code

,

Codex

등)를 쓰면, 단 하나의 프롬프트만으로 경쟁사의 SEO 상태(백링크, 오가닉 트래픽), 소셜 미디어 인지도, 가격 구조를 스크래핑하여 완벽한 대시보드로 요약할 수 있습니다. 심지어 AI가 데이터를 분석하여 "이 경쟁사는 트래픽은 많지만 모바일 속도가 느리니 우리는 모바일 최적화를 노려야 합니다"와 같은 전략적 조언까지 제공합니다.

실무 사용 사례: 경쟁사 웹사이트 SEO 및 건강도 점검 대시보드 구축 여러분이 음성 AI 에이전트 에이전시를 운영하고 있고, 경쟁사 웹사이트의 검색 엔진 최적화(SEO) 상태를 분석하고 싶다고 가정해 보겠습니다.

1. 환경 준비 및 API 설정 경쟁사 데이터를 가져오기 위해서는 'Data for SEO'와 같은 API(특정 소프트웨어의 데이터를 가져올 수 있는 일종의 비밀번호/통로)를 사용하는 것이 좋습니다. 프로젝트 폴더에 agents.md 파일을 만들고, AI에게 "SEO 분석 시 Data for SEO API를 사용하라"고 사전 지식을 부여합니다.

2. 에이전트에게 지시하기 (예제 프롬프트) 초보자도 쉽게 이해할 수 있도록, AI에게 어떤 역할을 해야 하고 어떤 결과물을 원하는지 명확하게 지시합니다.

예제 프롬프트 입력: "나는 현재 새로운 조경 회사를 창업하려고 해. 내가 벤치마킹하고 싶은 경쟁사의 웹사이트 URL(예: competitor-landscape.com)을 제공할 테니, 이 사이트의 현재 비즈니스 상태를 스크래핑하고 분석해 줘.

다음 항목들을 Data for SEO API나 웹 스크래핑 도구(Firecrawl 등)를 사용해 수집하고 HTML 대시보드 형태로 시각화해 줘.

SEO 건강도 점검: 오가닉 트래픽(자연 검색 유입량), 백링크(외부에서 걸어준 링크) 개수, 그리고 끊어진 링크(Broken links) 오류가 있는지 확인해 줘.

마케팅 전략: 현재 구글 애즈나 페이스북 광고를 실행 중인지, 소셜 미디어 인지도는 어떤지 모니터링해 줘.

가격 구조 및 리뷰: 웹사이트에 명시된 가격 정책을 요약하고, 수집 가능한 고객 리뷰들을 분석해 줘.

수집이 완료되면 분석된 데이터를 바탕으로, 경쟁사의 약점(Weaknesses), 개선점(Improvements), 그리고 우리가 노려야 할 타겟 키워드 제안(Keyword suggestions)을 포함한 경영진 보고서를 함께 작성해 줘."

AI의 작업 과정 및 결과 프롬프트를 입력하면 AI는 즉시 스크래퍼를 가동하여 경쟁사 웹사이트를 탐색합니다. 약 1~2분 뒤, 멋진 UI를 갖춘 대시보드를 생성하여 보여줍니다. 대시보드에는 "경쟁사 점수: 0/100 (백링크 없음, 오가닉 트래픽 존재)" 등의 직관적인 진단 결과가 나타납니다. 더 것은 AI가 제공하는 분석입니다. "경쟁사의 약점: 가격 정책이 불투명하여 고객 이탈이 발생할 수 있음. 제안: 우리는 투명한 가격 테이블을 메인 페이지에 배치하여 차별화하세요." 라는 식의 훌륭한 비즈니스 컨설팅 결과를 단 몇 분 만에 얻을 수 있습니다. 10명의 직원을 고용한 것과 같은 엄청난 생산성입니다.

# Codex 구체적 활용사례 37 - 제13장. 구글 맵 기반 리드(Lead) 수집 및 맞춤형 콜드 이메일 작성: 영업을 자동화하는 마법

제13장. 구글 맵 기반 리드(Lead) 수집 및 맞춤형 콜드 이메일 작성: 영업을 자동화하는 마법

초보자를 위한 개념 이해: 기초적인 리드 수집을 넘어선 초개인화(Enrichment) B2B 영업의 핵심은 양질의 잠재 고객(Lead)을 찾고, 그들에게 응답률 높은 콜드 이메일(영업 이메일)을 보내는 것입니다. 일반적으로 사람들은 구글 맵을 켜서 '로스앤젤레스 배관공'을 검색하고, 엑셀에 회사명과 전화번호를 복사 붙여넣기 합니다. 하지만 AI 에이전트는 이를 완벽히 자동화할 뿐만 아니라, '데이터 풍부화(Enrichment)'라는 마법을 부립니다.

그저 주소만 긁어오는 것이 아니라, 그 회사의 틱톡, 인스타그램, 페이스북, 유튜브까지 자동으로 방문하여 사장님의 이름과 최근 행보를 수집합니다. 그리고 이를 바탕으로 사람이 직접 쓴 것보다 더 정성스러운 콜드 이메일 초안을 무한대로 찍어냅니다.

실무 사용 사례: LA 지역 지붕 수리 업체(Roofer) 리드 수집 및 영업 자동화 여러분이 웹사이트를 제작해 주는 프리랜서나 영업 사원이라고 가정해 보겠습니다.

1. 에이전트에게 지시하기 (예제 프롬프트) 구글 맵 스크래핑 기능과 맞춤형 이메일 작성 기능을 연결하여 지시합니다.

예제 프롬프트 입력: "나는 웹사이트 제작 에이전시를 운영하고 있어. 지금부터 잠재 고객(Lead)을 수집하고 영업 이메일을 작성하는 파이프라인 대시보드를 구축해 줘.

스크래핑: 검색어 'Roofers(지붕 수리공)'와 지역 'Los Angeles(로스앤젤레스)'를 입력하면, Apify나 구글 맵 스크래퍼 API를 사용해 최소 20개의 업체 정보를 수집해 줘. 회사명, 리뷰 평점, 웹사이트, 전화번호를 가져와야 해.

데이터 풍부화(Enrichment): 각 업체의 웹사이트와 소셜 미디어(페이스북 등)를 스캔해서 비즈니스의 최신 활동이나 특징을 파악해 줘.

초개인화 아이스브레이커(Icebreaker): 수집된 데이터를 바탕으로 영업 통화나 콜드 이메일 첫 문장에 쓸 '아이스브레이커' 문장을 업체별로 하나씩 작성해서 구글 스프레드시트에 정리해 줘. 문장은 기계적이지 않고 완전히 개인화되어야 해."

AI의 작업 과정 및 결과 AI 에이전트는 스크래퍼를 가동해 18개의 웹사이트를 가진 업체를 찾아냅니다. 그리고 스프레드시트에 수집된 데이터를 기록하면서 다음과 같은 기가 막힌 맞춤형 이메일 초안(아이스브레이커)을 작성해 줍니다.

"수집된 데이터 예시: 80년 전통, 130회 이상의 수리 완료 기록 보유 업체"

"AI가 작성한 콜드 이메일 초안: 대표님, 80년이 넘는 인상적인 역사와 130회 이상의 지붕 수리를 성공적으로 마친 품질에 큰 감명을 받았습니다. LA 시장에서 보여주신 대표님의 철학에 대해 더 배우고, 이를 담아낼 완벽한 웹사이트 리뉴얼에 대해 짧게 논의하고 싶습니다."

초보자가 보기에도 이 이메일은 기초적인 복사/붙여넣기 스팸 이메일과 차원이 다릅니다. 나아가 다른 예시로는 "페이스북을 보니 새벽 2시에 파라마타 지역에서 터진 배관을 수리하러 빗속을 40분이나 운전해 가셨더군요! 그 헌신적인 스토리가 고객들의 마음을 훔쳤을 겁니다" 와 같은 스토리텔링 방식의 초개인화 이메일도 자동으로 생성해 냅니다. 이제 여러분은 전송 버튼만 누르면 됩니다.

# Codex 구체적 활용사례 37 - 제14장. 메타(Meta) 광고 라이브러리 스크래핑 및 광고 성과 분석 대시보드: 이기는 광고의 비밀 풀기

제14장. 메타(Meta) 광고 라이브러리 스크래핑 및 광고 성과 분석 대시보드: 이기는 광고의 비밀 풀기

초보자를 위한 개념 이해: 페이스북/인스타에서 잘 나가는 광고 훔쳐보기 제품을 팔거나 서비스를 홍보할 때 "어떤 광고를 만들어야 할까?" 고민하는 것은 초보 마케터의 흔한 고충입니다. 경쟁사가 현재 돈을 쏟아부으며 실행 중인 광고를 모두 볼 수 있는 곳이 바로 '메타(Meta) 광고 라이브러리'입니다. 하지만 수백 개의 광고를 눈으로 보며 어떤 게 영상인지, 이미지인지, 문구가 뭔지 분석하는 것은 불가능에 가깝습니다. AI를 쓰면 경쟁사의 광고 라이브러리 링크를 입력하는 것만으로, 수십 개의 광고 데이터를 스크래핑하고 AI가 직접 카피와 후킹(Hook) 포인트를 분석해 주는 전문 대시보드를 구축할 수 있습니다.

실무 사용 사례: 피부과(Med Spa) 경쟁사 광고 40개 일괄 분석

1. 에이전트에게 지시하기 (예제 프롬프트) 경쟁사의 메타 광고 라이브러리 URL을 준비한 뒤, 에이전트에게 이를 스크래핑하고 분석할 대시보드 구축을 지시합니다.

예제 프롬프트 입력: "페이스북 광고(Meta Ads)를 분석해 주는 '광고 분석 대시보드'를 구축해 줘. 내가 특정 브랜드의 메타 광고 라이브러리 URL(예: 미국 Med Spa 업체)을 입력하면, 스크래퍼 API를 사용해 최근 실행 중인 활성 광고 40개를 추출해 줘.

대시보드에는 다음 기능이 포함되어야 해.

형태 분류: 수집된 40개의 광고 중 비디오(Video), 이미지(Image), 캐러셀(Carousel) 형태가 각각 몇 개인지 분류해서 어떤 형태의 광고가 주력인지 시각화해 줘.

AI 광고 성과 분석: 특정 광고를 클릭하면 AI가 해당 광고의 '효과성(Effectiveness)'을 분석해 줘. 초반 훅(Hook), 타겟 오디언스 설정, 그리고 콜투액션(CTA) 문구가 적절한지 평가해 줘.

개선 제안: 성과가 저조할 것으로 예상되는 광고에 대해 더 매력적인 카피나 제안(Suggestion)을 텍스트로 피드백해 줘."

AI의 작업 과정 및 결과 AI가 구축한 대시보드에 URL을 넣고 '스크랩 실행'을 누르면, 단 몇 분 만에 완벽한 분석이 뜹니다.

광고 전략 파악: "총 40개 광고 중 캐러셀 19개, 이미지 12개, 영상 9개로 분석됩니다. 경쟁사 'Vagar Pro'는 캐러셀 광고를 주력으로 사용하고 있습니다." 이를 통해 우리도 캐러셀 광고를 테스트해야 한다는 힌트를 얻습니다.

광고 카피 분석 및 팩트폭행: AI는 특정 광고의 문구를 분석한 뒤, "콜투액션(CTA) 분석: 피부과 서비스 비즈니스에서 '문의하기(Contact us)'라는 문구는 너무 모호하고 효과적이지 않습니다. '무료 상담 예약하기(Book a consultation)'나 '치료 일정 잡기(Schedule treatment)'로 구체화해야 성과를 높일 수 있습니다" 라며 마케팅 전문가 수준의 피드백을 제공합니다. 마케팅 대행사에 수백만 원

을 주지 않아도, 초보자 누구나 완벽한 광고 분석 전략을 세울 수 있는 워크플로우입니다.

# Codex 구체적 활용사례 37 - 제15장. 유튜브 채널 전략 기획 및 썸네일/에셋 생성 자동화: 터지는 콘텐츠 무한 생성기

제15장. 유튜브 채널 전략 기획 및 썸네일/에셋 생성 자동화: 터지는 콘텐츠 무한 생성기

초보자를 위한 개념 이해: 유튜브 전략부터 디자인까지 AI로 끝내기 유튜브 채널을 키우기 위해서는 경쟁 채널의 조회수를 분석하고, 클릭을 유도하는 강력한 후킹(Hook) 대본을 쓰고, 시선을 사로잡는 썸네일을 만들어야 합니다. 과거에는 이를 위해 데이터 분석 도구를 돌리고, 챗GPT로 대본을 쓴 다음, 포토샵이나 캔바를 열어 일일이 썸네일 디자인을 편집해야 했습니다. 이제 AI 에이전트는 이 모든 것을 하나의 파이프라인으로 연결합니다. 경쟁사의 데이터를 스크래핑하여 기회를 찾고(기획), 대본을 쓰고(콘텐츠), 텍스트의 길이에 맞춰 반응형으로 레이아웃이 변하는 썸네일(디자인 에셋)까지 단 몇 분 만에 자동 생성합니다.

실무 사용 사례: 경쟁 유튜브 채널 분석 및 동적 썸네일 일괄 생성

1. 에이전트에게 지시하기 (예제 프롬프트) 전략 기획과 에셋 생성을 동시에 처리하도록 AI에게 지시합니다. (사전에 썸네일 템플릿의 가이드가 되는 이미지를 프로젝트 폴더에 넣어두면 좋습니다).

예제 프롬프트 입력: "나의 유튜브 채널 성장을 위한 '유튜브 전략가 에이전트' 파이프라인을 구축해 줘.

1단계: 경쟁사 분석 내가 경쟁 채널들의 이름(예: Jack Roberts 채널 등)을 주면, 조회수, 구독자 수, 참여율 지표를 분석해서 내 채널의 지표와 비교해 줘. 그리고 경쟁 채널에서 성과가 좋았던 콘텐츠를 바탕으로, 내가 만들어야 할 '콘텐츠 갭(Content gaps)'과 새로운 영상 아이디어 5개를 뽑아내 줘. 각 아이디어마다 바이럴 가능성이 높은 훅(Hook)과 제목을 함께 작성해 줘. 이 분석 결과는 구글 슬라이드 형식으로 요약해서 URL로 보여줘.

2단계: 브랜드 에셋(썸네일) 자동 생성 아이디어가 결정되면, 프로젝트 폴더에 있는 내 사진(아바타)과 게스트의 로고/사진을 사용해서 유튜브 썸네일을 자동으로 디자인해 줘. 제목 텍스트의 길이에 따라 폰트 크기가 자동으로 조절되어야 하고, 나와 게스트 두 명의 사진이 들어갈 경우 레이아웃이 그에 맞춰 동적으로 예쁘게 생성되도록 코드를 작성해 줘."

AI의 작업 과정 및 결과

구글 슬라이드 보고서: AI는 수 분 내에 구글 슬라이드 URL을 던져줍니다. 슬라이드를 열어보면 경쟁사(Jack Roberts 등)의 탑 퍼포먼스 영상 지표가 요약되어 있고, 우리 채널이 공략해야 할 5가지의 터지는 제목과 대본의 첫 3초 훅(Hook)이 전문적으로 작성되어 있습니다.

동적 썸네일 자동 디자인: 에이전트에게 "다음 주 워크숍 주제는 '

Claude Code

for Beginners'이고 호스트는 나와 Alex 두 명이야"라고 알려줍니다. AI는 디자인 스킬을 사용하여 폰트 크기를 완벽하게 맞추고, 두 사람의 아바타 이미지를 깔끔하게 합성한 썸네일을 즉시 생성해 냅니다. 초보 유튜버나 마케터는 이제 매주 썸네일 편집과 제목 짓기로 밤을 새울 필요가 없습니다. 단 한

번의 명령어로 유튜브 기획부터 디자인까지 모든 에셋을 찍어낼 수 있는 강력한 시스템을 갖추게 된 것입니다.

# Codex 구체적 활용사례 37 - 제16장. 유튜브 및 회의록 기반 소셜 미디어(LinkedIn/Twitter) 콘텐츠 재가공: 터지는 콘텐츠 무한 생성기

제16장. 유튜브 및 회의록 기반 소셜 미디어(LinkedIn/Twitter) 콘텐츠 재가공: 터지는 콘텐츠 무한 생성기

초보자를 위한 개념 이해: 왜 콘텐츠 재가공을 AI에게 맡겨야 할까? 크리에이터나 1인 기업가, 마케터에게 소셜 미디어(링크드인, 트위터 등) 업로드는 필수입니다. 하지만 매일 새로운 아이디어를 짜내고 글을 쓰는 것은 창작의 고통을 수반합니다. 여러분은 이미 수많은 훌륭한 콘텐츠 원석을 가지고 있습니다. 바로 여러분이 촬영한 유튜브 영상, 팟캐스트 녹음본, 심지어 매주 진행하는 화상 회의(Gong, Granola 등 기록 앱 사용)에서 나눈 인사이트 넘치는 대화들입니다.

과거에는 이를 재가공하기 위해 영상을 처음부터 다시 보며 타이핑하고, 트위터 길이에 맞춰 요약하고, 링크드인 스타일에 맞춰 어조를 수정하는 등 수작업에만 몇 시간이 걸렸습니다. 이제 AI 에이전트를 쓰면 여러분의 기존 콘텐츠를 스캔하여 '바이럴 지수(Viral Score)'를 평가하고, 여러분 특유의 어조를 복제하여 즉시 발행 가능한 포스트 초안을 무한대로 뽑아낼 수 있습니다.

실무 사용 사례: 매주 토요일 아침 8시, 자동 콘텐츠 기획 대시보드 구축 이 시스템의 핵심은 AI에게 "내가 작성했던 과거의 베스트 콘텐츠"를 학습시키는 것입니다. 여러분 특유의 어조(예: 직설적이고, 반대 의견을 제시하며, 구체적인 숫자를 쓰고, 짧은 문장 사용)를 에이전트에게 지시합니다.

1. 환경 준비 바탕화면에 Content_Machine 폴더를 만듭니다. 그 안에 여러분의 유튜브 대본, 팟캐스트 녹취록 문서들을 모아둡니다. 그리고 my_style.md 파일을 만들어 여러분의 과거 잘된 링크드인 포스트 예시들을 붙여넣습니다.

2. 에이전트에게 지시하기 (예제 프롬프트) 에이전트에게 콘텐츠 재가공 스킬(Skill)을 만들도록 구체적으로 지시합니다.

예제 프롬프트 입력: "나는 매주 소셜 미디어에 글을 올려야 하는 비즈니스 대표야. 현재 폴더에 있는 내 최근 유튜브 영상 대본들과 Granola 앱으로 기록한 이번 주 회의록 텍스트 파일들을 모두 스캔해 줘.

인사이트 추출: 이 텍스트들에서 바이럴 가능성(1~100점 점수 부여)이 높은 핵심 주제, 인사이트, 인용구 5가지를 추출해 줘. 사람들이 뻔하게 생각하는 것과 반대되는 의견(Contrarian takes)을 우선적으로 찾아줘.

초개인화 글쓰기: 추출된 5개의 주제를 바탕으로, 각각 '6개의 트윗으로 이뤄진 트위터 스레드'와 '전문적인 링크드인 포스트'를 하나씩 작성해 줘.

어조 강제: 글을 쓸 때는 반드시 폴더에 있는 my_style.md 파일에 정의된 내 말투(직설적, 이모티콘 최소화, 구체적 숫자 사용)를 100% 모방해서 써야 해.

이 모든 결과를 구글 문서나 HTML 대시보드로 요약해서 내 이메일이나 Slack으로 매주 토요일 아침 8시에 전송하는 자동화 스케줄을 만들어 줘."

AI의 작업 과정 및 결과 AI는 즉시 대본과 회의록을 읽고, "SEO 최적화만 하면 이미 뒤처진 것이다" 라는 여러분의 회의 중 발언을 포착해 냅니다. 그리고 이를 바탕으로 완벽하게 후킹(Hook)되는 트위터 스레드와 링크드인 포스트를 작성해냅니다. 여러분은 커피를 마시며 AI가 만든 글을 읽고, 마음에 드는 것을 골라 복사 및 예약 발행만 하면 됩니다. 아이디어 고갈에서 영원히 해방되는 마법입니다.

# Codex 구체적 활용사례 37 - 제17장. 긴 동영상 하이라이트 컷 자동 편집 및 브랜딩 에셋 추가: 비디오 편집의 완전 자동화

제17장. 긴 동영상 하이라이트 컷 자동 편집 및 브랜딩 에셋 추가: 비디오 편집의 완전 자동화

초보자를 위한 개념 이해: 동영상 편집을 코딩 에이전트가 한다고? "코딩을 도와주는 AI가 동영상을 편집한다고?" 초보자분들은 의아해하실 수 있습니다. 하지만 동영상 편집 역시 내부적으로는 코드로 제어할 수 있는 작업입니다. 1시간짜리 웨비나, 줌(Zoom) 미팅, 인터뷰 영상을 틱톡이나 유튜브 쇼츠에 올릴 수 있게 1분짜리 핵심 영상 5개로 자르고 자막과 아웃트로(로고 영상)를 붙이는 작업은 편집자에게 맡기면 건당 수십만 원과 며칠의 시간이 소요됩니다. 하지만 에이전트에게 FFmpeg(비디오 처리용 도구) 사용 권한을 주거나 스킬(Skill)로 지정하면, AI가 트랜스크립트(자막)를 읽고 "이 부분이 제일 재밌군"이라고 판단한 뒤 정확한 타임스탬프(시간)를 기준으로 영상을 싹둑 잘라냅니다.

실무 사용 사례: 1시간 라이브 워크숍에서 5개의 링크드인용 쇼츠 추출하기

1. 환경 준비 Video_Highlights 폴더를 만듭니다. 그 안에 워크숍 원본 영상 파일(workshop_speaker_view.mp4 및 workshop_screenshare_view.mp4)과 텍스트 트랜스크립트 파일(transcript.txt)을 넣습니다. 그리고 마지막에 붙일 브랜드 로고 아웃트로 영상(9x_outro.mp4)도 준비합니다.

2. 에이전트에게 지시하기 (예제 프롬프트) 화면이 어떻게 전환되어야 하는지 감독처럼 구체적으로 지시합니다.

예제 프롬프트 입력: "현재 폴더에 있는 워크숍 녹화 영상과 전체 트랜스크립트(대본)를 분석해 주는 비디오 편집 파이프라인을 구축해 줘.

하이라이트 추출: 전체 대본을 읽고, 사람들이 흥미로워할 만한 핵심 주제나 펀치라인이 담긴 1분 내외의 하이라이트 구간 5곳을 찾아내 줘.

비디오 컷 자동화: 찾아낸 5개 구간의 시작 시간과 종료 시간(타임스탬프)을 파악해서 원본 영상을 잘라내 줘(FFmpeg 스크립트 작성 및 실행).

다이나믹 컷 편집: 강연자가 화면의 글을 읽을 때는 screenshare_view.mp4를 보여주고, 일반적인 설명을 할 때는 speaker_view.mp4로 교차 편집(Cross-cut) 되도록 코드를 짜줘.

브랜딩 에셋 결합: 5개의 클립 영상이 끝날 때마다 폴더에 있는 9x_outro.mp4 브랜드 아웃트로 영상을 뒤에 자연스럽게 이어 붙여 줘. 최종적으로 Social_Clips라는 폴더를 만들고 5개의 MP4 완성본을 저장해 줘."

AI의 작업 과정 및 결과 AI는 스스로 영상 처리 스크립트를 작성하여 약 5분 만에 작업을 완료합니다. 폴더를 열어보면 완벽하게 잘린 5개의 비디오 파일이 들어있습니다. 영상을 재생해 보면 강연자가 "이 스킬은 사실 기초적인 텍스트일 뿐입니다!"라고 말하는 임팩트 있는 구간에서 스피커 화면이 보이고, 마지막에는 회사의 로고가 멋지게 등장하며 마무리됩니다. 매주 1~2시간씩 걸리던 컷 편집 노동이 단 5분 만에 끝납니다.

# Codex 구체적 활용사례 37 - 제18장. 노션(Notion) 문서를 커뮤니티(Circle) 포스트로 실시간 동기화: 지루한 발행 작업 자동화

제18장. 노션(Notion) 문서를 커뮤니티(Circle) 포스트로 실시간 동기화: 지루한 발행 작업 자동화

초보자를 위한 개념 이해: 포맷이 깨지는 복사 붙여넣기는 이제 그만 팀의 튜토리얼이나 매뉴얼을 보통 노션(

Notion

)에 아름답게 작성합니다. 글머리 기호, 볼드체, 그리고 설명 중간중간 들어가는 수많은 스크린샷 이미지들이 있죠. 하지만 이를 회사의 공식 커뮤니티(예: Circle, WordPress, 네이버 카페 등)로 옮겨 발행하려고 하면 악몽이 시작됩니다. 전체를 복사해서 붙여넣으면 텍스트는 넘어가지만, 이미지들은 복사가 안 되어 엑스박스가 뜨고 유튜브 임베드 링크는 깨집니다. 결국 에디터를 열고 수십 장의 이미지를 다시 다운로드해서 일일이 제자리에 끼워 넣어야 합니다. AI를 쓰면 노션 문서의 링크 하나만 던져주더라도 완벽하게 마이그레이션(이동)할 수 있습니다.

실무 사용 사례: 노션 '발행 대기' 상태 변경 시 자동 퍼블리싱

1. 환경 준비 노션 데이터베이스를 사용하고, API 연동이 가능한 커뮤니티(Circle)의 API 키를 발급받아 에이전트에게 제공합니다.

2. 에이전트에게 지시하기 (예제 프롬프트)

예제 프롬프트 입력: "/publish_

notion

_to_circle 스킬을 만들어줘. 이 작업은 노션의 튜토리얼 문서를 우리 회사의 Circle 커뮤니티로 완벽하게 마이그레이션 및 자동 발행하는 워크플로우야.

내가 특정 노션 문서의 링크를 주면, 해당 페이지의 모든 텍스트 데이터와 포맷(볼드, 헤딩 등)을 그대로 다운로드해 줘.

노션 문서 중간중간에 삽입된 모든 스크린샷 이미지 파일들을 로컬로 다운로드한 뒤, Circle API를 통해 커뮤니티 서버로 업로드하고 본문 제자리에 맞게 이미지를 삽입해 줘.

만약 유튜브 영상 링크가 있다면 커뮤니티에 맞게 완벽한 임베드(Embed) 포맷으로 변환해 줘.

모든 변환이 완료되면 Circle의 'AI 튜토리얼' 게시판에 이 문서를 새 포스트로 즉시 발행(Publish)하고, 완료되었다는 링크를 나에게 알려줘."

AI의 작업 과정 및 결과 프롬프트를 전송하면 에이전트가 파이썬 스크립트를 작성하여 노션과 커뮤니티의 API를 다리로 연결합니다. 작업이 완료되었다는 커뮤니티 링크를 클릭해 보면, 노션에서 작성했던 수많은 스크린샷과 유튜브 영상, 텍스트 서식이 단 한 줄의 어긋남 없이 완벽하게 복제되어 포스팅된 것을 확인할 수 있습니다. 복붙 노동에 쏟던 에너지를 이제 더 좋은 글을 쓰는 데 집중할 수

있습니다.

# Codex 구체적 활용사례 37 - 제19장. 이메일 수신함 자동 분류 (Triage) 및 맞춤형 답장 초안 작성: AI 비서가 내 이메일 톤을 훔치다

제19장. 이메일 수신함 자동 분류(Triage) 및 맞춤형 답장 초안 작성: AI 비서가 내 이메일 톤을 훔치다

초보자를 위한 개념 이해: 나보다 내 말투를 더 잘 아는 AI 이메일 비서 아침에 일어나서 이메일 수신함을 열어보면 쓸데없는 뉴스레터, 스팸, 광고 메일 사이에 고객의 중요한 요청이나 파트너의 미팅 제안이 숨어있습니다. 이 수십 통의 이메일을 읽고 거르는 데만 하루의 시작을 낭비하게 됩니다. AI를 써서 Gmail이나 Outlook 수신함을 스캔하게 만들고, 스팸은 무시하며 오직 '나의 조치가 필요한 이메일'만 뽑아내게 할 수 있습니다. 더 나아가, 내 과거 이메일들을 학습한 AI가 내 특유의 서명과 어투를 완벽하게 모방하여 답장 초안(Draft)까지 작성해 둡니다.

실무 사용 사례: 매일 아침 6시 이메일 분류 및 Slack 알림

1. 환경 준비 에이전트의 'Connector(커넥터)' 설정에서 Gmail 권한을 부여합니다. 그리고 memory.md 파일에 "나의 이메일 작성 원칙(예: 항상 두괄식으로 짧게 쓰고, 끝맺음 인사는 '감사합니다, 랩장 드림'으로 한다)"을 정의해 둡니다.

2. 에이전트에게 지시하기 (예제 프롬프트)

예제 프롬프트 입력: "매일 아침 6시에 실행되는 이메일 분류(Triage) 자동화 스케줄을 설정해 줘.

Gmail 커넥터를 사용해서 지난 24시간 동안 온 내 수신함을 스캔해 줘. 뉴스레터나 스팸, 알림 이메일은 전부 무시해.

오직 내가 직접 행동(Action)을 취해야 하는 중요한 이메일(예: 미팅 요청, 고객의 CS 문의, 결제 이슈 등)만 솎아내어 요약 리스트를 만들어 줘.

솎아낸 각 중요 이메일에 대해서, 내 프로젝트 폴더 내 memory.md에 저장된 '내 이메일 작성 스타일과 어조(Tone of Voice)' 원칙을 엄격하게 적용하여 답장 초안(Draft)을 작성해 줘.

이 모든 결과를 내 Slack으로 보내고, 이메일 정보는

Airtable

데이터베이스에 기록해 줘."

AI의 작업 과정 및 결과 아침 7시에 컴퓨터를 켜고 Slack을 열면 AI가 보낸 메시지가 와 있습니다. "총 28개의 이메일을 스캔했으며, 1개의 새로운 중요 이메일이 발견되었습니다." 해당 메일은 파트너사의 미팅 요청이었고, 그 아래에는 평소 여러분이 이메일을 쓰는 방식과 100% 동일한 친절하면서도 짧은 답장 초안이 작성되어 있습니다. 여러분은 그저 읽어보고 "전송" 버튼만 누르면 됩니다.

# Codex 구체적 활용사례 37 - 제20장. 고객 피드백 서베이 데이터를 분석한 전문 경영진 보고서 제작: 수천 개의 피드백을 단 1분 만에 시각화하기

제20장. 고객 피드백 서베이 데이터를 분석한 전문 경영진 보고서 제작: 수천 개의 피드백을 단 1분 만에 시각화하기

초보자를 위한 개념 이해: 엑셀 속 숨겨진 고객의 목소리(VOC) 꺼내기 워크숍, 행사, 혹은 신제품 출시 후 구글 폼(Google Forms)이나 타입폼(Typeform)을 통해 수집한 고객 피드백은 귀중한 자산입니다. 하지만 응답 데이터가 담긴 CSV 파일을 열어보면, 1부터 5점까지의 숫자와 고객이 주관식으로 길게 적어 놓은 장문의 피드백이 뒤섞여 있어 분석하기가 막막합니다. 경영진이나 파트너사에게 이 엑셀 파일을 그대로 던져주면 아무도 읽지 않죠. AI 에이전트를 쓰면 수천 개의 응답 데이터를 분석하여 긍정 및 부정 요인을 정량화하고, 전문적이고 아름답게 포맷팅된 PDF나 Word 문서 보고서를 순식간에 찍어낼 수 있습니다.

실무 사용 사례: 법률 및 재무 워크숍 후 피드백 보고서(PDF) 자동 생성

1. 환경 준비 Feedback_Reports 폴더에 고객 응답 데이터가 담긴 feedback_survey.csv 파일을 넣습니다. 이 파일에는 고객의 부서, 만족도 평점, 가장 좋았던 점(주관식), 개선점(주관식) 등이 들어있습니다.

2. 에이전트에게 지시하기 (예제 프롬프트)

예제 프롬프트 입력: "현재 폴더에 있는 feedback_survey.csv 데이터를 분석해서, 이해관계자나 경영진이 즉시 읽고 파악할 수 있는 전문적인 피드백 보고서를 작성해 줘.

핵심 데이터 크런치: 참석자들의 전반적인 만족도 평점 평균, AI 도입 의향 비율 등 핵심 지표(High-level stats)를 계산해서 상단에 배치해 줘.

주관식 감성 분석: 고객들이 서술형으로 작성한 '가장 좋았던 점(긍정 요인)'과 '개선점(부정 요인)'을 분석해서, 각각 많이 언급된 상위 3가지 패턴으로 그룹화하고 요약해 줘.

인사이트 및 다음 단계: 이 데이터를 바탕으로 우리가 다음 워크숍에서 어떤 주제를 추가해야 할지(Use cases demanded) 통찰력 있는 '핵심 인사이트'를 제안해 줘.

보고서 생성: 이 분석 내용을 완벽하게 포맷팅된 'Executive Summary(경영진 요약보고서)' 형태의 Word(Docx)나 PDF 문서로 생성하여 폴더에 저장해 줘."

AI의 작업 과정 및 결과 AI는 파이썬(Pandas 등)을 이용해 엑셀 데이터를 연산하고 자연어 처리(NLP)를 통해 감성 분석을 진행합니다. 단 몇 분 뒤, 폴더에 Feedback_Report_AI_Workshop.pdf라는 세련된 파일이 생성됩니다. 파일을 열어보면 "전반적 만족도: 4.8/5.0", "강점: 실무에 즉시 적용 가능한 프롬프트 예시 제공", "개선점: 실습 시간이 조금 더 필요함" 등의 인사이트가 완벽하게 목차별로 잘 정리되어 있습니다. 이 문서를 그대로 경영진 이메일로 전송하면, 여러분은 즉시 탁월한 성과를 인정받게 됩니다.

# Codex 구체적 활용사례 37 - 제21장. 주간 CEO 리뷰 및 비즈니스 핵심 지표 추적 보고서: AI가 기억하는 자동화된 경영진 보고

제21장. 주간 CEO 리뷰 및 비즈니스 핵심 지표 추적 보고서: AI가 기억하는 자동화된 경영진 보고

초보자를 위한 개념 이해: AI에게 '기억력'을 부여하다 매주 월요일 아침, 경영진이나 팀원들에게 지난주의 성과를 요약해서 보고해야 하는 업무가 있습니다. 웹사이트 방문자 수, 신규 가입자 수, 매출액 등을 엑셀에서 뽑아 지난주 수치와 비교(Delta)하여 증감을 계산하고 보고서를 작성하죠. 매주 반복되는 이 지루한 작업을 AI에게 맡기려면 한 가지 문제가 있었습니다. 일반적인 AI 챗봇은 '지난주 데이터'를 기억하지 못하기 때문에 매주 과거 데이터를 다시 업로드해야 한다는 점입니다.

하지만 Claude Co-work의 'Co-work Projects(코워크 프로젝트)' 기능을 사용하면 상황이 180도 달라집니다. 이 프로젝트 공간은 '메모리(기억력)'를 가지고 있어서, 매번 작업을 실행할 때마다 과거의 대화와 데이터를 기억합니다. 따라서 이번 주 데이터만 던져주면 AI가 알아서 "아, 지난주에는 매출이 100만 원이었는데 이번 주는 120만 원이니 20% 증가한 녹색 신호네요!"라고 델타(증감)를 계산해 주는 완벽한 주간 보고서를 자동 생성합니다.

실무 사용 사례 및 환경 설정

프로젝트 생성: Claude Co-work 데스크톱 앱에서 새로운 'Co-work Project'를 생성하고 이름을 Weekly CEO Review라고 짓습니다.

지시 사항(Context) 설정: 프로젝트 설정 파일(예:

agents.md

또는 프로젝트 지시문)에 "너는 우리 회사의 데이터 분석가야. 매주 내가 최신 지표를 주면, 지난주 수치와 비교해서 보고서를 써줘"라고 역할을 부여합니다.

초보자를 위한 예제 프롬프트 (매주 실행하는 명령어):

"Run this week's report(이번 주 보고서를 실행해 줘).

〔이번 주 원시 데이터 복사/붙여넣기 또는 CSV 파일 첨부〕

이 데이터를 바탕으로 다음 양식에 맞춰 '비밀 CEO 주간 리뷰(Secretly CEO Weekly Review)' 보고서를 작성해 줘.

헤드라인(Headline): 이번 주 비즈니스 성과를 한 줄로 요약할 것.

핵심 지표(Key Numbers): 네가 기억하고 있는 '지난주 수치'와 비교하여 증감률(Delta)을 퍼센트로 표시해 줘.

녹색 신호(Green Flags): 전주 대비 크게 성장했거나 긍정적인 지표 3가지.

주의 대상 및 워치리스트(Watch List): 하락했거나 조치가 필요한 위험 지표 2가지.

다음 주 집중 목표(Focus): 이 데이터를 바탕으로 다음 주에 우리가 집중해야 할 액션 아이템."

AI의 작업 과정 및 결과 프롬프트를 실행하면 AI는 단 몇 초 만에 이전 주차의 문맥(Context)을 불러옵니다. 그리고 지난주와 이번 주의 숫자를 교차 분석하여 완벽하게 포맷팅된 경영진 보고서를 화면에 띄워줍니다. 여러분은 이 내용을 그대로 복사해서 슬랙(Slack)이나 이메일로 팀원들에게 전송하기만 하면 됩니다. 매주 2시간씩 걸리던 데이터 취합 및 분석 작업이 단 10초 만에 끝나는 경험을 할 수 있습니다.

# Codex 구체적 활용사례 37 - 제22장. 회의록 비교 분석을 통한 누락된 액션 아이템(Action Item) 추출: 아무것도 놓치지 않는 AI 비서

제22장. 회의록 비교 분석을 통한 누락된 액션 아이템(Action Item) 추출: 아무것도 놓치지 않는 AI 비서

초보자를 위한 개념 이해: 인간의 기록과 AI의 기록을 교차 검증하다 회의가 끝나면 우리는 보통 노션(

Notion

)에 회의록을 정리합니다. 하지만 회의 중 오고 간 수많은 이야기 중, 누가 언제까지 무엇을 하기로 했는지(액션 아이템)를 사람이 손으로 적다 보면 놓치는 부분이 반드시 발생합니다. 반면, 최근 많이 사용하는 AI 음성 기록 앱(Gemini, Gong, Granola 등)은 회의의 모든 대화를 토씨 하나 틀리지 않고 텍스트 트랜스크립트(녹취록)로 남겨둡니다.

이 두 가지를 써 AI에게 "AI가 녹음한 전체 대화 기록과 사람이 노션에 요약한 회의록을 비교해서, 우리가 노션에 적는 것을 깜빡 잊어버린 약속(Commitment)이나 작업이 있는지 찾아내라"고 지시하는 것이 이 장의 핵심입니다.

실무 사용 사례 및 커넥터 연동

커넥터(Connector) 설정: Claude Co-work 설정의 '커넥터(Connectors)' 메뉴로 들어가서 Google Drive(음성 녹취록이 저장되는 곳)와

Notion

(회의록이 있는 곳) 권한을 AI에게 부여합니다.

작업 폴더 구성: AI가 접근할 수 있는 작업 폴더를 지정해 둡니다.

초보자를 위한 예제 프롬프트:

"Google Drive에 저장된 오늘 오전 마케팅 회의의 'Gemini 트랜스크립트(녹취록 전문)' 문서를 읽어와 줘. 그리고

Notion

커넥터를 사용해서 내가 방금 작성한 '마케팅 회의 요약 노트' 페이지를 불러와 줘.

두 문서를 꼼꼼하게 교차 검증(Cross-reference)해서 다음 작업을 수행해 줘:

누락된 작업 추출: 녹취록 대화 중에는 누군가 하겠다고 약속(Commitments)했지만, 내 노션 회의록에는 빠져 있는(Fell through the cracks) 숨겨진 액션 아이템들을 모두 찾아내 줘.

담당자 및 기한 할당: 찾아낸 각 작업이 누구의 담당인지, 언제까지 하기로 대화가 오갔는지 명시해 줘.

노션 업데이트: 누락된 항목들을 내 노션 페이지 하단의 '추가 액션 아이템' 섹션에 자동으로 업데이트해 줘."

AI의 작업 과정 및 결과 AI는 Google Drive에서 수십 장 분량의 회의 녹취록을 순식간에 읽어내고,

Notion

의 회의록을 불러와 두 문서를 대조합니다. 그리고 *"분석 결과: 대화 중 Sarah가 다음 주 수요일까지 랜딩 페이지 시안을 보내주기로 약속했으나,

Notion

회의록에는 이 내용이 누락되어 있습니다"*라고 보고해 줍니다. 깐깐하고 꼼꼼한 관리자처럼, 인간이 놓친 모든 구멍을 AI가 완벽하게 메워주는 강력한 업무 워크플로우입니다.

# Codex 구체적 활용사례 37 - 제23장. 고객 이탈(Churn) 위험 예측 및 업셀링(확장) 기회 모니터링: 통화 녹음에서 돈을 찾아내는 마법

제23장. 고객 이탈(Churn) 위험 예측 및 업셀링(확장) 기회 모니터링: 통화 녹음에서 돈을 찾아내는 마법

초보자를 위한 개념 이해: 고객의 진짜 속마음 읽어내기 구독형 서비스(

SaaS

)나 B2B 비즈니스에서 고객이 서비스를 취소(이탈, Churn)하는 것은 회사의 치명적인 손실입니다. 고객이 갑자기 서비스를 취소하기 전, 영업 사원이나 CS 팀과의 통화에서 반드시 어떤 '불만의 징후'를 보이기 마련입니다. 반대로, "이런 기능도 혹시 있나요?"라며 추가 결제(업셀링)를 암시하는 기회도 통화 속에 숨어 있습니다.

하지만 경영진이나 관리자가 수백 개의 영업 통화(Gong, Fireflies 등 녹취록)를 매일 다 들을 수는 없습니다. 이때

MCP

(Model Context Protocol) 기능을 사용해 AI를

CRM

도구와 연결하면, AI가 매일 수백 개의 통화 스크립트를 스캔하여 고객의 '감정(Sentiment)'과 '의도'를 파악해 대시보드로 요약해 줍니다.

실무 사용 사례 및 환경 설정 Cursor나

Claude Code

같은 도구에 HubSpot(고객 관리

CRM

) 및 Gong(통화 녹음 도구)

MCP

를 플러그인으로 연동합니다. 이렇게 하면 AI가 자연어 명령만으로도

CRM

데이터에 직접 접근할 수 있습니다.

초보자를 위한 예제 프롬프트:

"HubSpot 및 Gong

MCP

커넥터를 사용하여, 이번 주 진행된 기존 고객들과의 모든 영업/CS 통화 스크립트를 분석하는 '고객 건강도 모니터링 시스템'을 가동해 줘.

다음 두 가지 신호를 중점적으로 찾아내서 표 형태로 보고해 줘:

이탈 위험(Churn Risk) 감지: 통화 내용 중 고객이 '좌절감(Frustrated)', '실망(Disappointed)'을 표현했거나, 우리가 '선제적으로 대응하지 못한다(Not proactive enough)'고 지적한 대목을 찾아내 줘. 각 고객사별 '이탈 확률(Churn probability)'을 퍼센트로 계산하고, 우리가 즉시 취해야 할 '즉각적인 조치(Immediate actions)'를 제안해 줘.

업셀링/확장 기회(Expansion): 고객이 만족해하며 다른 마케팅 채널이나 추가 솔루션에 대해 물어보는 등, 예산을 확장할 기회가 보인다면 이를 포착하여 요약해 줘."

AI의 작업 과정 및 결과 AI는 통화 스크립트 속 문맥을 분석합니다. "A고객사의 CMO가 '우리가 뭔가 잘못하고 있나요? 전환율이 떨어지네요'라고 우려를 표명했습니다. 이탈 확률은 75%입니다. 즉각적인 조치: 해당 고객사 계정 매니저가 내일 당장 리뷰 미팅을 잡아야 합니다." 와 같은 섬뜩할 정도로 정확한 인사이트를 추출합니다. 팀원들에게 이 경고를 바로 공유함으로써, 회사는 떠나갈 뻔한 수천만 원의 매출을 방어하고 나아가 추가 매출 기회까지 확보할 수 있습니다.

## Codex 구체적 활용사례 37 - 제24장. 실패한 영업 기회(Lost Deals) 부활을 위한 CRM 데이터 연동: 죽은 계약을 살려내는 영업 사원 AI

제24장. 실패한 영업 기회(Lost Deals) 부활을 위한 CRM 데이터 연동: 죽은 계약을 살려내는 영업 사원 AI

초보자를 위한 개념 이해: 돈이 되는 AI의 끝판왕, Deal Revivor 과거에 영업을 진행하다가 "지금은 예산이 없네요", "타이밍이 안 맞습니다"라는 이유로 성사되지 못하고 휴지통에 들어간 계약(Lost Deals)들이 무수히 많을 것입니다. 사람들은 기억력이 짧아서 1년 전 실패한 고객에게 다시 연락하는 것을 쉽게 잊어버립니다.

이 워크플로우를 'Deal Revivor(계약 부활기)'라고 부릅니다. AI가

CRM

(HubSpot 등)을 스캔하여 1~3년 전 실패했던 엄청난 규모의 계약들을 찾아냅니다. 그런 다음, 해당 기업의 최신 뉴스를 웹에서 검색하여 아주 자연스럽고 소름 돋게 개인화된 맞춤형 후속(Follow-up) 이메일 초안을 작성해 여러분의 슬랙(Slack)으로 보내줍니다.

실무 사용 사례 및 환경 설정

Claude Code

나 특정 AI 대시보드를 Slack 및 HubSpot API와 연결합니다. AI가 "이 사람에게 메일을 보낼까요?"라고 슬랙으로 제안하면, 여러분은 "전송(Send Email)" 버튼만 누르는 구조(Human-in-the-loop)를 만듭니다.

초보자를 위한 예제 프롬프트:

"나를 위한 'Deal Revivor(영업 기회 부활)' 자동화 워크플로우를 구축해 줘.

CRM

데이터 발굴: HubSpot

CRM

을 스캔하여 과거 10개월에서 50개월 사이에 '손실(Lost)' 처리된 1억 원($100K) 이상의 큰 계약 건들을 모두 찾아내 줘.

최신 정보 스크래핑: 찾아낸 회사의 이름과 담당자(예: Jason)를 기반으로 웹 검색을 진행해 그 회사의 최근 뉴스나 성과(예: 투자 유치, 신제품 출시)를 파악해 줘.

맞춤형 콜드 이메일 작성: 이를 바탕으로 다음과 같은 톤으로 팔로업 이메일 초안을 작성해 줘. (예시 어조): '안녕하세요 Jason, 최근 〔검색한 최신 뉴스〕 소식을 보았습니다. 정말 축하드립니다. 우리가 마지막으로 연락한 지 약 〔N〕개월이 지났네요. 회사에 많은 변화가 있었을 텐데, 다시 한번 저희 솔루션에 대해 논의할 타이밍인지 가볍게 통화 가능할까요?'

슬랙(Slack)으로 이 초안과 함께 '이메일 전송(Send Email)' 버튼을 띄워줘."

AI의 작업 과정 및 결과 AI는 즉시 HubSpot을 뒤져 43개월 전에 아깝게 놓친 7억 5천만 원 ($750K) 규모의 계약을 발굴해 냅니다. 그리고 담당자의 승진이나 회사의 최신 뉴스를 엮어 너무나도 완벽한 이메일 초안을 슬랙에 띄워줍니다. 여러분은 커피를 마시며 슬랙 알림을 보고, 초안을 스윽 읽은 뒤 '전송'을 누르기만 하면 됩니다. 코딩을 1도 몰라도 AI를 써 수억 원의 잃어버린 매출을 되찾아올 수 있는 엄청난 시스템입니다.

# Codex 구체적 활용사례 37 - 제25장. 신규 프로덕트 GTM(Go-to-Market) 전략 수립 및 이메일 시퀀스 작성: 15분 만에 끝내는 마케팅 런칭 전략

제25장. 신규 프로덕트 GTM(Go-to-Market) 전략 수립 및 이메일 시퀀스 작성: 15분 만에 끝내는 마케팅 런칭 전략

초보자를 위한 개념 이해: 수천만 원짜리 컨설팅을 AI 한 번의 클릭으로 새로운 제품이나 서비스(예:

SaaS

소프트웨어, 온라인 강의 등)를 시장에 출시(Go-to-Market, GTM)할 때, 마케터들은 치밀한 전략을 세워야 합니다. 고객이 제품을 인지하고 가입한 뒤 유료 결제까지 이어지게 만드는 '이메일 마케팅 시퀀스(연속 이메일 발송)'를 기획하고 작성하는 데는 보통 몇 주가 걸리며 대행사에 수천만 원을 지급하기도 합니다.

하지만

Claude Code

를 쓰면, 단 한 번의 프롬프트와 15분의 시간만으로 목표 고객 분석부터 유저 여정(User Journey)별 촘촘한 맞춤형 이메일 카피까지 완벽한 GTM 전략을 구축할 수 있습니다.

실무 사용 사례 및 환경 설정 프로젝트 폴더 내에 신제품에 대한 설명서(예: readme.md, product_info.txt)를 넣어두어 AI가 우리 제품이 어떤 가치를 주는지 미리 학습하게 합니다.

초보자를 위한 예제 프롬프트:

"이번에 새로 런칭하는 우리 제품(예: ClickFlow)의 GTM(Go-to-Market) 전략과 전체 사용자 온보딩 이메일 시퀀스를 구축해 줘. 폴더 내에 있는 제품 문서를 읽고 우리 제품의 핵심 가치 제안을 파악해 줘.

다음 각 사용자 여정(User Journey) 단계별로 발송될 완벽한 마케팅 이메일 카피(제목, 본문, 콜투액션)를 작성해 줘:

가입 환영(Onboarding/Welcome): 가입 직후 14일 무료 평가판을 시작한 유저에게 제품 시작 방법(예: 웹사이트 연동, 첫 콘텐츠 발행 등)을 안내하는 이메일.

장바구니 유기(Abandoned Signup): 가입하다가 중간에 이탈한 유저를 다시 데려오는 이메일.

평가판 종료 및 취소(Trial Canceled/Churn): 평가판을 끝내고 취소한 유저에게 아쉬움을 표하며 피드백을 묻는 이탈 방지 이메일.

업셀링 및 교차 판매(Cross-sell/Upsell): 제품을 잘 사용 중인 충성 고객에게 상위 요금제나 추가 관리형 서비스(Managed Services)를 권유하는 이메일."

AI의 작업 과정 및 결과 프롬프트를 전송하면 AI는 여러분의 제품 퍼널 아키텍처(Funnel Architecture)를 스스로 설계하고, 각 단계별로 완벽한 심리학적 카피라이팅이 적용된 수십 개의 이메일 텍스트를 쏟아냅니다. "환영합니다! 14일 동안 우리 제품이 맞는지 확인해 보세요. 먼저 Google Search Console을 연동하고 첫 콘텐츠를 발행해 보세요." 와 같은 구체적이고 실용적인 안내가 담겨 있습니다. 여러분은 단 10~15분 동안 이메일 초안을 리뷰하고 5% 정도만 수정한 뒤 마케팅 팀에 넘기면 런칭 준비가 끝납니다. 전문가를 고용하는 비용과 시간을 압도적으로 단축하는 최고의 방법입니다.

# Codex 구체적 활용사례 37 - 제26장. 인터랙티브 슬라이드쇼 및 프레젠테이션 자동 디자인: 브랜드 맞춤형 PPT 10분 만에 찍어내기

제26장. 인터랙티브 슬라이드쇼 및 프레젠테이션 자동 디자인: 브랜드 맞춤형 PPT 10분 만에 찍어내기

〔개념 이해: 초보자를 위한 배경 설명〕 과거에 AI로 프레젠테이션을 만들려고 시도해 본 분들은 보통 실망하곤 합니다. AI에게 "슬라이드 좀 만들어줘"라고 무턱대고 지시하는 이른바 '바이브 디자인(Vibe Designing)' 방식을 사용하면, 엉뚱한 폰트, 촌스러운 색상, 일관성 없는 레이아웃으로 이루어진 결과물을 받게 되기 때문입니다. 이로 인해 끝없는 수정 지시(프롬프트)를 반복하다가 결국 포기하고 직접 PPT를 만들게 됩니다.

하지만 Claude Design이나 Claude Co-work의 강력한 템플릿(Template) 기능과 디자인 시스템(Design System)을 쓰면 이야기가 달라집니다. AI에게 미리 우리 회사의 폰트, 로고, 브랜드 색상, 그리고 20장짜리 슬라이드 레이아웃의 '뼈대'를 학습시켜 두면, AI는 당신이 제공한 텍스트 요약본이나 스크립트만 읽고도 사전에 승인된 레이아웃에 맞춰 복붙하듯 완벽한 슬라이드쇼를 순식간에 자동 생성해 냅니다.

〔사전 준비 및 환경 설정〕

디자인 시스템 구축: Claude Design 시스템 탭에서 회사 이름, 브랜드 색상(Hex 코드), 폰트 등을 설정합니다. 초보자라면 '디자인 시스템 크리에이터 스킬(Design System Creator Skill)'을 써 웹사이트 URL만 입력하면 AI가 알아서 웹사이트의 로고와 브랜드 에셋을 긁어와 시스템을 세팅하게 할 수 있습니다.

템플릿 등록: 과거에 잘 만들었던 20장 이상의 PPT 슬라이드나, 인터넷에서 찾은 예쁜 레이아웃(예: Dribbble, Canva 등)의 스크린샷을 Claude에 올리고 "이 형식들을 슬라이드 템플릿으로 저장해 줘"라고 지시하여 템플릿화합니다.

〔구체적인 예제 프롬프트〕

"나는 다음 주에 진행할 'AI 자동화 트렌드' 유튜브 영상 및 세미나를 위한 시각적 슬라이드쇼를 준비하고 있어. 내 프로젝트 폴더에 저장된 '미리 승인된 28페이지 슬라이드 템플릿'과 '우리 회사의 브랜드 디자인 시스템'을 엄격하게 적용해 줘.

첨부한 워드 문서(세미나 스크립트 요약본)를 읽고 다음 작업을 수행해 줘: 1. 슬라이드 매핑: 스크립트의 각 주제에 잘 어울리는 템플릿 레이아웃(예: 3단 분할 레이아웃, 전체 화면 이미지 레이아웃 등)을 선택해서 할당해 줘. 2. 카피라이팅 작성: 각 슬라이드에 들어갈 간결하고 임팩트 있는 헤드라인과 본문 텍스트를 작성해 줘. 3. 렌더링: 이 모든 것을 결합하여 완벽하게 포맷팅된 인터랙티브 슬라이드쇼 형태로 즉시 생성해 줘."

〔AI의 작업 과정 및 마법 같은 결과〕 이 프롬프트를 전송하면, AI(Visual Layout Skill 등 사용)는 스크립트를 분석하여 "1번 슬라이드에는 14번 템플릿을 적용하고 헤드라인은 이렇게 쓰겠습니다"라며 계획을 세웁니다. 사용자가 승인하면 Claude는 즉시 HTML/CSS 형태나 내부 프레젠테이션 스킬을 사용하여 슬라이드를 렌더링합니다. 결과물을 보면 여러분의 브랜드 컬러가 입혀진 20장의 슬라이

드가 완성되어 있습니다. 만약 텍스트 크기나 문구가 마음에 들지 않는다면, 슬라이드 내의 요소를 직접 클릭하여 손쉽게 텍스트를 수정하거나 픽셀 크기를 조절할 수 있습니다. 5시간 걸리던 PPT 제작이 단 10분 만에 전문가 수준으로 끝나는 워크플로우입니다.

# Codex 구체적 활용사례 37 - 제27장. 데스크톱 앱 및 웹사이트에 대한 전문적인 UX 오딧(Audit) 수행: AI를 우리 회사의 QA 테스터로 고용하기

제27장. 데스크톱 앱 및 웹사이트에 대한 전문적인 UX 오딧(Audit) 수행: AI를 우리 회사의 QA 테스터로 고용하기

〔개념 이해: 초보자를 위한 배경 설명〕 새로운 웹사이트나 앱을 런칭할 때, 사용자가 회원가입부터 결제까지 매끄럽게 진행할 수 있는지 테스트하는 과정(UX Audit 및 QA)은 필수적입니다. 하지만 모든 버튼을 눌러보고 오류를 찾는 것은 엄청난 인내심과 시간을 요구합니다. 최근 Claude와

Codex

에는 '컴퓨터 제어(

Computer Use

)' 및 '브라우저 제어(

Browser Use

)'라는 혁신적인 기능이 탑재되었습니다. 이는 AI가 여러분의 모니터 화면을 직접 눈으로 보고, 마치 사람처럼 마우스를 움직이고 키보드를 타이핑하여 브라우저를 탐색할 수 있는 권한을 부여하는 것입니다. AI를 QA(품질 보증) 테스터로 고용하여, 앱을 샅샅이 뒤지게 하고 마찰 요인(Friction points)을 찾아내 전문 보고서로 작성하게 만들 수 있습니다.

〔사전 준비 및 환경 설정〕

Claude 데스크톱 앱의 '설정(Settings)'에서

Computer Use

권한을 켭니다. Mac 사용자의 경우 시스템 환경설정에서 화면 기록(Screen Recording) 및 접근성(Accessibility) 권한을 허용해야 합니다.

보안을 위해 처음에는 AI가 임의로 클릭하지 못하도록 '행동 전 묻기(Ask before acting)' 모드로 설정하는 것을 권장합니다.

〔구체적인 예제 프롬프트〕

"나의 새로운 데스크톱 웹앱(예: your zone of genius app)에 대해 전문적인 UX 오딧(Audit)을 수행해 줘. 브라우저를 열고 앱의 URL을 입력해서 접속해 줘.

다음과 같이 실제 신규 사용자의 관점으로 전체 유저 플로우(User Flow)를 직접 테스트해 줘: 1. 로그인 플로우: 로그인 페이지에 내 이메일을 입력하고, 내 이메일 수신함으로 가서 일회용 비밀번호(OTP)를 복사한 뒤, 다시 앱으로 돌아와 로그인을 성공시켜 줘. 2. 인터페이스 테스트: 대시보드 내

의 다양한 탭과 버튼을 무작위로 클릭해 보고, 검색창에 텍스트를 입력하거나 필터를 조작해 보며 마찰이 발생하는 단계나 끊어진 링크(Broken links)가 있는지 스트레스 테스트를 진행해 줘. 3. 보고서 작성: 사용자가 혼란스러워할 만한 UI 결함이나 어색한 부분을 찾아내고, 이를 어떻게 수정해야 할지 권고안이 담긴 'UX 오딧 요약 보고서'를 문서로 생성해 줘."

〔AI의 작업 과정 및 마법 같은 결과〕 명령을 내리면 AI는 브라우저를 띄우고 실제로 마우스 커서를 움직입니다. 로그인 창에 이메일을 타이핑하고, 탭을 전환하여 이메일에서 코드를 복사해 로그인을 완수합니다. 이후 앱 곳곳을 돌아다니며 "검색 기능이 너무 융통성 없이 작동합니다", "유튜브 외부 링크를 클릭했는데 새 탭으로 열리지 않습니다", "비어 있는 상태(Empty State) 화면이 너무 휑합니다" 등의 버그와 UI 개선점을 쏙쏙 잡아냅니다. 약 15분 후, AI는 이 모든 테스트 과정을 종합하여 "우선적으로 고쳐야 할 4가지 UX 결함"이 담긴 완벽한 경영진용 감사 보고서를 제출합니다. 값비싼 QA 팀이나 외부 컨설턴트를 고용하지 않아도 최고 수준의 피드백을 얻을 수 있습니다.

# Codex 구체적 활용사례 37 - 제28장. n8n, Make.com 등 자동화 워크플로우(JSON) 코드 변환 및 구축: 시각적 툴과 코딩의 완벽한 융합

제28장. n8n, Make.com 등 자동화 워크플로우(JSON) 코드 변환 및 구축: 시각적 툴과 코딩의 완벽한 융합

〔개념 이해: 초보자를 위한 배경 설명〕

n8n

이나

Make.com

, Zapier 같은 시각적 자동화 툴은 블록을 드래그 앤 드롭하여 앱들을 연결할 수 있어 초보자들에게 인기가 높습니다. 하지만 자동화가 복잡해지면 블록들이 스파게티처럼 엉키고 유지보수가 힘들어집니다. 플랫폼의 유료 실행 횟수(Task 쿼터)를 소진하는 비용 문제도 발생하죠. 이때 AI를 쓰면 시각적으로 만든

n8n

워크플로우를 완벽한 독립형 파이썬(Python) 스크립트로 변환하여 로컬 PC나 무료 서버에서 영구적으로 돌릴 수 있습니다. 반대로, 파이썬 코드를 AI에게 분석시켜

n8n

에 바로 붙여넣을 수 있는 JSON 파일로 만들어 달라고 지시할 수도 있습니다.

〔사전 준비 및 환경 설정〕

기존에 만들어둔

n8n

워크플로우(예: 유튜브 라이브 스트리밍 생성, Luma 이벤트 등록, Zoom 미팅 개설을 자동화하는 흐름) 화면에서 우측 상단의 '다운로드'를 클릭하여 워크플로우를 JSON 파일 형태로 바탕화면 폴더에 저장합니다.

해당 폴더에서

Claude Code

나

Codex

를 실행합니다.

〔구체적인 예제 프롬프트〕

"현재 폴더에 내가 다운로드한

n8n

워크플로우 파일(JSON)이 있어. 이 파일은 내가 매주 라이브 워크숍을 준비할 때 Luma, Zoom,

Notion

을 자동 연동하는 로직을 담고 있어.

1. 로직 문서화: 이 JSON 파일을 주의 깊게 읽고, 이 자동화가 정확히 어떤 트리거로 작동하며 각 단계에서 어떤 일이 일어나는지 초보자도 이해할 수 있도록 마크다운(Markdown) 문서로 설명서를 작성해 줘. 2. Python 코드 변환: 이제 이 시각적 워크플로우를 영구적으로 내 컴퓨터에서 실행할 수 있도록 완벽한 Python 스크립트로 변환해 줘. Zoom, Luma, Notion, YouTube와 통신하는 API 로직은 각각 별도의 파이썬 파일로 깔끔하게 분리하고, 이를 모두 통제하는 메인 실행 스크립트를 작성해 줘. 3. 명령어 생성: 마지막으로 이 코드를 이 채팅창에서 단축어로 쉽게 실행할 수 있도록 커스텀 슬래시 명령어(Slash Command)로 만들어 줘."

〔AI의 작업 과정 및 마법 같은 결과〕 AI는 복잡한 기계어인 JSON 파일을 분석하여 단 1분 만에 "이 워크플로우는 줌 미팅을 생성하고 노션에 데이터를 기록하는 자동화입니다"라는 깔끔한 설명서를 만들어 냅니다. 이어서 AI는 각 앱의 API를 연결하는 파이썬 코드를 스스로 분할 작성하여 패키지로 만들어 냅니다. 이제 여러분은 비싼 월 구독료를 내며 자동화 플랫폼을 쓸 필요 없이, 터미널에서 명령어 한 줄만 입력하면 AI가 파이썬 코드를 구동하여 완벽히 동일한 자동화 작업을 백그라운드에서 무료로 처리하게 만들 수 있습니다.

## Codex 구체적 활용사례 37 - 제29장. 브라우저 제어(Computer Use)를 통한 온라인 쇼핑 및 웹 제어 자동화: AI 개인 쇼핑 비서 두기

제29장. 브라우저 제어(Computer Use)를 통한 온라인 쇼핑 및 웹 제어 자동화: AI 개인 쇼핑 비서 두기

〔개념 이해: 초보자를 위한 배경 설명〕 "퇴근길에 스마트폰 문자로 '커피 원두 다 떨어졌어, 아마존에서 늘 먹던 걸로 주문해 줘'라고 보내면, 집에 있는 컴퓨터가 켜지면서 알아서 장바구니에 담아둔다." 영화 속 아이언맨의 자비스(Jarvis) 같은 이야기지만, Claude Co-work의 '디스패치(Dispatch)' 기능과 '컴퓨터 제어(

Computer Use

)'를 결합하면 오늘 당장 구현할 수 있는 워크플로우입니다. AI에게 모니터를 볼 수 있는 눈과 마우스를 조작할 수 있는 손을 주고, 스마트폰에서 원격으로 지시를 내리는 구조입니다. 쇼핑처럼 일상적인 반복 웹 서핑을 AI에게 완전히 위임할 수 있습니다.

〔사전 준비 및 환경 설정〕

데스크톱에서 Claude Co-work를 켜고, 설정에서

Computer Use

를 활성화합니다. 브라우저 접근 권한을 허용합니다.

보안 팁: AI에게 쇼핑 권한을 줄 때는 해킹이나 프롬프트 인젝션(잘못된 지시)으로 인한 금전 사고를 막기 위해, Privacy.com과 같은 서비스에서 월 한도(예: 50달러)가 설정된 가상 신용카드(Virtual Cards)를 발급받아 아마존 계정에 등록해 두는 것이 강력히 권장됩니다.

〔구체적인 예제 프롬프트 (스마트폰 iMessage 또는 Dispatch 앱에서 전송)〕

"지금 집에 있는 생수가 다 떨어졌어. 내 데스크톱의 브라우저를 열어서 Amazon.com에 접속해 줘.

다음 지시를 따라 쇼핑을 대신 수행해 줘:

검색창에 '평점 좋은 무라벨 생수 2L'를 검색해 줘.

검색 결과 중에서 리뷰가 최소 4.5점 이상이고, 가격이 15달러 이하인 가성비 좋은 상품을 하나 찾아내 줘.

해당 상품의 상세 페이지로 들어가서 '장바구니에 담기(Add to Cart)' 버튼을 마우스로 클릭해 줘.

안전을 위해 최종 결제(Buy now) 버튼은 누르지 말고, 장바구니에 잘 담겼다는 스크린샷을 찍어서 내 스마트폰(iMessage 또는 Dispatch)으로 전송해 줘. 확인 후 내가 직접 결제할게."

〔AI의 작업 과정 및 마법 같은 결과〕 당신이 이동 중일 때, 집에 있는 데스크톱의 AI가 즉시 깨어납니다. AI 커서가 스스로 움직여 크롬을 열고, 아마존에 접속해 검색어를 타이핑합니다. 리뷰 평점을 분

석하여 적합한 생수를 찾은 뒤, 정확히 '장바구니 담기' 버튼을 클릭합니다. 몇 분 뒤 당신의 스마트폰으로 "지시하신 상품을 장바구니에 담았습니다"라는 메시지와 함께 인증 스크린샷이 도착합니다. 당신은 집에 돌아갈 필요 없이 폰으로 승인(결제)만 누르면 일상적인 쇼핑 업무가 완료됩니다.

# Codex 구체적 활용사례 37 - 제30장. 스마트폰/스마트워치 원격 지시(Dispatch)를 통한 데스크톱 백그라운드 작업: 시간과 공간의 제약 허물기

제30장. 스마트폰/스마트워치 원격 지시(Dispatch)를 통한 데스크톱 백그라운드 작업: 시간과 공간의 제약 허물기

〔개념 이해: 초보자를 위한 배경 설명〕 업무를 하다 보면 꼭 컴퓨터 앞에 앉아 있어야만 일이 진행되는 것은 아닙니다. 밥을 먹거나, 헬스장에서 운동 중이거나, 잠들기 직전에 문득 "아, 내일 회의를 위해 경쟁사 데이터를 수집해야 하는데!"라는 생각이 떠오를 때가 있죠. 이럴 때 Claude의 '디스패치(Dispatch)' 기능을 사용하면, 주머니 속 스마트폰이나 애플 워치로 에이전트에게 지시를 던져놓고 나는 내 할 일을 계속할 수 있습니다. 디스패치는 휴대폰을 '리모컨'으로 만들어, 집에 켜져 있는 데스크톱의 AI 에이전트가 무거운 데이터 분석이나 웹 스크래핑 작업을 백그라운드에서 몇 시간 동안 대신 처리하게 만드는 마법 같은 기능입니다.

〔사전 준비 및 환경 설정〕

데스크톱(Mac Mini 등)의 Claude 앱 설정에서 'Dispatch'를 활성화하고 휴대폰의 Claude 앱과 연동합니다.

에이전트가 작업을 수행하는 도중 컴퓨터가 절전 모드로 들어가면 작업이 끊기므로, 설정에서 '화면 켜짐 유지(Keep Awake)' 옵션을 반드시 활성화해 둡니다.

〔구체적인 예제 프롬프트 (커피숍 대기줄에서 스마트폰으로 전송)〕

"내일 아침 마케팅 회의를 위해 유튜브 채널 성과 분석이 필요해. 데스크톱 환경에서 컴퓨터 제어(

Computer Use

) 권한을 사용해 다음 백그라운드 작업을 실행해 줘.

내 브라우저를 열어 YouTube Analytics 페이지에 접속해 줘.

최근 업로드한 10개 영상의 조회수(Views), 클릭률(CTR), 평균 시청 지속 시간(Average View Duration) 데이터를 전부 추출해 줘.

이 데이터를 바탕으로 깔끔하게 포맷팅된 Excel 스프레드시트(.xlsx) 파일을 생성하고, 이번 주 성과가 좋은 이유에 대한 핵심 인사이트를 요약해서 문서 하단에 추가해 줘.

파일 생성과 분석이 완료되면 내 스마트폰 Dispatch 스레드로 결과 요약본을 보내줘. 내일 아침 출근하면 데스크톱에서 확인할게."

〔AI의 작업 과정 및 마법 같은 결과〕 커피를 주문하고 폰을 주머니에 넣는 순간, 원격지에 있는 데스크톱 AI는 브라우저를 띄우고 유튜브 스튜디오에 접속하여 복잡한 데이터를 수집하기 시작합니다. 수천 건의 숫자를 긁어모아 엑셀에 깔끔한 차트와 표를 구성합니다. 당신이 커피를 다 마시고 집에 돌아오거나, 다음 날 아침 데스크톱을 켜보면, 화면 중앙에는 완벽하게 정리된 엑셀 파일과 분석 요약

본이 얌전하게 기다리고 있습니다. 모바일 기기의 기동성과 데스크톱 AI의 강력한 컴퓨팅 파워를 결합하여, 내가 자는 동안에도 24시간 쉬지 않고 나를 위해 일하는 진정한 'AI 비서'를 구축하는 궁극의 워크플로우입니다.

# Codex 구체적 활용사례 37 - 제31장. 에어테이블(Airtable) 데이터를 쓴 맞춤형 인보이스/문서 일괄 생성: 수작업 서류 업무의 완전한 자동화

제31장. 에어테이블(Airtable) 데이터를 쓴 맞춤형 인보이스/문서 일괄 생성: 수작업 서류 업무의 완전한 자동화

〔초보자를 위한 개념 이해: 왜 에어테이블과 AI의 결합인가?〕 프리랜서, 에이전시, 혹은 중소기업의 재무 담당자라면 매월 말일마다 찾아오는 고통스러운 작업이 있습니다. 바로 수십, 수백 명의 고객에게 보낼 '인보이스(청구서)'나 '영수증', '계약서'를 만드는 일입니다. 고객의 이름, 청구 금액, 날짜, 내역을 엑셀에서 확인하고, 워드나 디자인 툴(Canva 등) 템플릿에 일일이 복사하여 붙여넣은 뒤, PDF로 변환해서 다시 저장하는 과정은 엄청난 반복 노동과 오타의 위험을 수반합니다.

지금까지는 Zapier나 Make 같은 자동화 툴을 이용해 이를 해결하려 했지만, 템플릿의 디자인을 섬세하게 맞추거나 복잡한 표를 렌더링하는 데에는 한계가 있었습니다. 하지만

Claude Code

나

Codex

같은 AI 에이전트를 에어테이블(

Airtable

)과 연결하면 이야기가 달라집니다. AI가 에어테이블 데이터베이스를 실시간으로 읽고, 완벽한 브랜드 디자인이 적용된 HTML/CSS 기반의 인보이스를 생성한 뒤, 이를 고품질 PDF로 일괄 변환하여 다시 에어테이블의 '첨부파일' 란에 자동으로 업로드해 줍니다.

〔사전 준비 및 환경 설정〕

에어테이블(

Airtable

) 베이스 세팅: 고객 정보와 청구 내역이 담긴 에어테이블 베이스를 준비합니다. 핵심 테이블은 두 개입니다.

Invoices 테이블: 고객명, 이메일, 총 청구 금액, 진행 상태(Status: '생성 대기', '발송 대기' 등), 그리고 빈 '첨부파일(Attachment)' 컬럼을 만듭니다.

Line Items 테이블: 각 청구서에 들어갈 세부 항목(예: 웹사이트 개발비, 유지보수비)을 기록합니다.

에이전트에게 템플릿 학습시키기: 프로젝트 폴더 내에 sample_invoice.pdf나 디자인 가이드라인 문서를 넣어두어, AI가 우리 회사의 브랜드 컬러, 로고 위치, 폰트 스타일 등을 미리 학습하게 만듭니다.

API 권한 부여: 에이전트(

Claude Code

등)가 에어테이블에 접근할 수 있도록 개인 API 키(Personal Access Token)를 환경 변수(.env)에 입력합니다.

〔초보자를 위한 구체적인 에이전트 지시 프롬프트〕 모든 준비가 끝났다면, 에어테이블에서 상태를 '생성 대기(To Create)'로 바꾼 뒤 에이전트 터미널을 열고 다음 프롬프트를 입력합니다.

"/create_invoices 자동화 스킬을 실행해 줘. 이 작업은 에어테이블과 연동하여 대량의 인보이스 PDF를 생성하는 작업이야.

다음 단계에 따라 완벽하게 자동화해 줘:

데이터 가져오기: 에어테이블 API를 호출해서 Invoices 테이블 중 진행 상태(Status)가 '생성 대기(To create)'로 되어 있는 모든 레코드를 읽어와 줘. 동시에 연관된 Line Items 데이터도 모두 불러와.

HTML 인보이스 렌더링: 폴더에 있는 sample_invoice 스타일 가이드를 바탕으로, 불러온 각 고객의 데이터를 동적으로 매핑하여 아름다운 HTML/CSS 형식의 청구서로 렌더링해 줘.

PDF 변환: 렌더링된 HTML을 Puppeteer나 Playwright 라이브러리를 사용해 고해상도 PDF 파일로 변환해 줘.

임시 호스팅 및 업로드: 에어테이블 API는 URL을 통해서만 첨부파일을 업로드할 수 있어. 생성된 PDF를 임시 호스팅 서버(예: AWS S3, 혹은 imgbb 같은 임시 스토리지)에 잠시 업로드한 뒤, 그 URL을 사용해 에어테이블의 'Attachment(첨부파일)' 컬럼에 PDF를 업로드해 줘.

상태 업데이트: 업로드가 성공하면 해당 레코드의 상태를 '발송 대기(Ready to Send)'로 변경하고 작업을 종료해."

〔AI의 작업 과정 및 마법 같은 결과〕 프롬프트를 전송하면 에이전트는 즉시 파이썬(Python) 또는 Node.js 스크립트를 스스로 작성하고 실행합니다.

1분 경과: 에이전트가 에어테이블에서 50명의 '생성 대기' 고객 목록을 성공적으로 읽어옵니다.

3분 경과: 에이전트가 백그라운드에서 브라우저(Playwright)를 띄워 HTML을 PDF로 변환하는 작업을 병렬로 처리합니다.

5분 경과: 당신이 듀얼 모니터로 에어테이블 화면을 보고 있다면, 마법 같은 일이 벌어집니다. 텅 비어있던 50명의 고객 첨부파일 컬럼에 PDF 아이콘이 1초에 하나씩 '탁, 탁, 탁' 하며 자동으로 생성되어 업로드됩니다. 동시에 상태 컬럼은 '발송 대기'로 초록색 불이 들어옵니다.

생성된 PDF를 클릭해 보면, 회사 로고가 완벽하게 박혀 있고, 고객별 세부 청구 내역 표가 아름답게 계산되어 들어가 있습니다. 과거 재무팀 직원이 일주일 내내 야근하며 만들었던 수백 장의 제안서, 청구서, 계약서를 단 5분 만에, 그것도 에러율 0%로 만들어내는 혁명적인 업무 자동화 워크플로우입니다.

# Codex 구체적 활용사례 37 - 제32장. 주식/가상화폐 포트폴리오 분석 및 매매 자동화 대시보드 구축: 나만의 AI 프라이빗 뱅커 고용하기

제32장. 주식/가상화폐 포트폴리오 분석 및 매매 자동화 대시보드 구축: 나만의 AI 프라이빗 뱅커 고용하기

〔초보자를 위한 개념 이해:

MCP

(Model Context Protocol)로 증권사 API 정복하기〕 투자자들은 보통 MTS(모바일 트레이딩 시스템)나 HTS(홈 트레이딩 시스템)를 보며 자신의 주식이나 가상화폐를 관리합니다. "내가 산 삼성전자가 지금 얼마지?", "내 포트폴리오는 기술주에 너무 치중되어 있나?" 이런 분석을 하려면 엑셀을 열고 숫자를 옮겨 적어 복잡한 차트를 만들어야 합니다.

하지만

MCP

(Model Context Protocol)라는 혁신적인 기술을 사용하면, AI 에이전트에게 증권사 시스템과 직접 대화할 수 있는 '통역기'를 달아줄 수 있습니다. 인도의 거대 증권사인 Zerodha나 한국의 증권사 API, 혹은 가상화폐 거래소(Binance, Upbit 등)의 API를 AI와 연결하면, AI가 실시간으로 내 계좌의 잔고, 수익률, 종목을 들여다보고 경영진 보고서 수준의 완벽한 포트폴리오 분석 대시보드를 시각화해 줍니다. 나아가 조건이 맞으면 직접 매수/매도 주문까지 버튼 하나로 실행하는 나만의 'AI 프라이빗 뱅커'를 구축할 수 있습니다.

〔사전 준비 및 환경 설정〕

API 키 발급: 사용 중인 증권사나 거래소(Zerodha 등)의 개발자 페이지에서 API Key와 Secret Key를 발급받아 환경 변수(.env)에 저장합니다.

MCP

서버 연결: Claude 데스크톱 앱이나 Cursor IDE의 설정에서

MCP

서버를 추가합니다. (예: 증권사 API와 통신하는 간단한 Python 스크립트를

MCP

서버로 등록합니다.) 이를 통해 AI는 안전하게 내 계좌를 조회할 권한을 얻습니다.

〔초보자를 위한 구체적인 에이전트 지시 프롬프트〕

"Zerodha

MCP

서버 커넥터를 사용해서 내 주식 계좌에 완벽하게 동기화되는 '포트폴리오 분석 대시보드'를 구축해 줘.

다음 지시사항을 따라 줘:

계좌 스캔 및 수익률 분석: 현재 내 계좌의 모든 보유 종목(Holdings)과 포지션을 읽어와 줘. 총 투자 원금, 현재 가치, 그리고 미실현 손익(Unrealized P&L)을 계산해서 상단에 큰 숫자로 표시해 줘.

섹터 및 노출도 시각화(Visualizations): 각 종목이 어떤 산업군(예: 기술, 인프라, 금융, 해외 ETF 등)에 속하는지 네가 스스로 리서치해서 분류해 줘. 그리고 내 자산이 특정 섹터에 너무 쏠려 있지 않은지 확인할 수 있는 '파이 차트(Pie Chart)'를 대시보드 중앙에 시각화해 줘.

벤치마크 및 개별 종목 평가: 내 포트폴리오의 큰 수익을 낸 효자 종목(Top winner)과 큰 손실을 낸 종목(Top loser)을 하이라이트 해 줘.

AI 매매 자동화 버튼(주문 실행): 내가 대시보드에서 특정 종목을 클릭하고 'AI 매도(Sell)' 버튼을 누르면, 네가 직접

MCP

주문 API를 호출해서 시장가로 해당 종목을 매도하는 스크립트를 작성해 줘. (단, 실제 체결 전 반드시 나에게 슬랙/터미널로 최종 승인(Yes/No)을 묻는 안전장치를 넣어 줘!)"

〔AI의 작업 과정 및 마법 같은 결과〕

분석 및 시각화: 에이전트가 즉각 API를 호출하여 내 계좌의 14개 종목을 읽어옵니다. 사용자는 본인이 산 'MAFANG'이라는 종목이 무엇인지도 잊고 있었지만, AI는 "MAFANG은 미국 대형 기술주(Tech)에 투자하는 해외 ETF입니다"라고 친절하게 분류합니다.

대시보드 생성: 몇 분 뒤, 로컬 브라우저에 화려한 대시보드가 열립니다. "총 포트폴리오 자산: 2,000만 원, 미실현 수익: +420만 원. 큰 손실 종목: PTM (-38% 하락). 자산의 18%가 인프라 섹터에, 12%가 해외 기술주에 노출되어 있습니다"라는 브리핑과 함께 파이 차트가 그려집니다.

주문 실행 테스트: 하락폭이 큰 종목을 클릭하고 '매도 스크립트'를 실행합니다. AI는 터미널에서 " 현재 잔고를 확인했습니다. 시장가로 매도 주문을 전송할까요?"라고 묻습니다. 'Yes'를 누르면, API를 통해 실제 증권사 서버로 주문이 전송되고 체결 알림이 뜹니다. (주의사항: 초보자의 경우 AI 모델의 환각(Hallucination)으로 인한 잘못된 주문을 막기 위해, 매매 기능은 반드시 소액 테스트와 인간의 수동 승인(Human-in-the-loop) 과정을 거치도록 세팅해야 합니다.)

# Codex 구체적 활용사례 37 - 제33장. Git Worktree를 쓴 다국어(번역) 및 다수 디자인 동시 병렬 개발: 분신술로 일하는 AI 개발팀 만들기

제33장. Git Worktree를 쓴 다국어(번역) 및 다수 디자인 동시 병렬 개발: 분신술로 일하는 AI 개발팀 만들기

〔초보자를 위한 개념 이해: 가지(Branch)와 나무(

Worktree

)의 차이〕 웹사이트나 앱을 만들 때, 한 번에 여러 가지 버전을 실험해 보고 싶을 때가 있습니다. 예를 들어, "이 랜딩 페이지를 한국어, 프랑스어, 독일어로 각각 번역해서 만들어 볼까?" 혹은 "하나는 모던한 스위스 디자인, 하나는 힙한 브루탈리즘(Brutalism), 하나는 귀여운 스타일 등 3가지 레이아웃으로 동시에 코딩해 보면 어떨까?" 하는 상황입니다.

전통적인 Git의 '브랜치(Branch)' 방식에서는 하나의 폴더에서 하나의 브랜치만 열어볼 수 있습니다. 따라서 AI 에이전트 1번이 스위스 디자인을 코딩하는 30분 동안, 나는 다른 디자인을 시도하지 못하고 그냥 기다려야 합니다(병목 현상 발생). 하지만 Git

Worktree

(작업 트리) 기능을 사용하면 완전히 새로운 마법이 펼쳐집니다. 원본 코드베이스의 Git 히스토리를 공유하면서도 물리적으로 완전히 분리된 여러 개의 폴더(복제본)를 순식간에 찍어냅니다. 이를 통해 우리는 3명의 AI 에이전트를 각기 다른

Worktree

폴더에 동시에 투입하여(병렬 처리), 서로 코드 충돌 없이 완벽하게 독립적인 3가지 앱을 동시에 개발하게 만들 수 있습니다.

〔사전 준비 및 환경 설정〕

작업 디렉토리 구성: 바탕화면에 메인 프로젝트 폴더(예: App_Main)를 준비합니다. Git 저장소로 초기화되어 있어야 합니다(git init).

Worktree

관리 폴더 생성: 메인 폴더 옆에

Worktree

복제본들을 모아둘

Worktree

s라는 빈 폴더를 하나 만듭니다.

〔초보자를 위한 구체적인 에이전트 지시 프롬프트〕 터미널에서

Codex

또는

Claude Code

를 메인 폴더에서 실행하고 멀티 에이전트 지시를 내립니다.

"현재 이 메인 프로젝트를 기반으로 3가지 각기 다른 프론트엔드 디자인을 병렬로 테스트해 보고 싶어. Git

Worktree

를 사용해서 작업을 구성해 줘.

1. Worktree 폴더 3개 생성: git worktree add 명령어를 사용해서 옆에 있는 Worktrees 폴더 내에 각각 design-swiss, design-brutalism, design-japanese라는 이름의 분리된 Worktree 폴더를 3개 생성해 줘. 각각 독립된 브랜치와 연결되어야 해.

2. 3명의 병렬 에이전트 투입:

design-swiss 폴더에는 '스위스 인터내셔널 타이포그래피(Swiss International Typographics)' 스타일로 랜딩 페이지를 재설계하는 에이전트를 투입해.

design-brutalism 폴더에는 투박하고 강렬한 '브루탈리즘(Brutalism)' 스타일로 코딩하는 에이전트를 투입해.

design-japanese 폴더에는 여백의 미가 돋보이는 '일본 미니멀리즘(Japanese minimalism)' 스타일로 코딩하는 에이전트를 투입해.

3. 병렬 로컬 서버 구동: 내가 세 가지 디자인을 눈으로 실시간 비교할 수 있도록, 스위스 디자인은 5171번 포트, 브루탈리즘은 5172번 포트, 일본식은 5173번 포트에서 동시에 로컬 서버(npm run dev)를 구동시켜 줘."

〔AI의 작업 과정 및 마법 같은 결과〕

분신술 실행: 메인 에이전트가 순식간에 3개의 물리적 폴더를 찍어내고, 3개의 개별 AI 에이전트 세션(Sub-agents)을 각 폴더에 투입합니다. 화면에는 3개의 에이전트가 동시에 각기 다른 CSS와 UI 코드를 작성하며 막대그래프가 올라가는 장관이 펼쳐집니다.

충돌 없는 독립 개발: 각 에이전트는 서로의 파일을 덮어쓰거나 방해하지 않습니다. 각자의

Worktree

세계에서 코딩하고 테스트를 진행합니다.

실시간 리뷰 및 병합(Merge): 15분 뒤 코딩이 모두 끝났습니다. 당신은 브라우저를 열어 localhost:5171, 5172, 5173 탭을 띄웁니다. 세 가지 완전히 다른 분위기의 디자인 앱이 완벽하게 돌아가고 있습니다. "음, 스위스 디자인(5171)이 제일 마음에 드네!"라고 결정했다면, 에이전트에게 "스위스 디자인 버전 브랜치를 메인 브랜치로 병합(Merge)해 주고, 나머지

Worktree

들은 삭제해 줘"라고 지시합니다.

며칠이 걸릴 다국어 번역 작업이나 A/B 테스트용 다수 UI/UX 디자인 개발 작업을 AI 분신술(

Worktree

)을 통해 단 30분 만에 끝내는 진정한 1인 개발자의 궁극기가 완성됩니다.

## Codex 구체적 활용사례 37 - 제34장. 코드 리뷰, 테스트 주도 개발 (TDD) 및 아키텍처 자동 리팩토링: 내 코드를 뜯어고치는 엄격한 시니어 엔지니어 고용하기

제34장. 코드 리뷰, 테스트 주도 개발(TDD) 및 아키텍처 자동 리팩토링: 내 코드를 뜯어고치는 엄격한 시니어 엔지니어 고용하기

〔초보자를 위한 개념 이해: 바이브 코딩(Vibe Coding)의 함정과 TDD의 마법〕 AI에게 "게시판 기능 만들어 줘"라고 모호하게 지시하는 이른바 '바이브 코딩'을 하다 보면, 당장 눈앞에서는 작동하지만 내부 코드는 엉망진창인 '스파게티 코드(Spaghetti Code)'가 만들어지기 일쑤입니다. 비즈니스 로직과 데이터베이스 연결이 하나의 파일에 뭉쳐 있고(모놀리식), 테스트 코드는 전무하며,

SQL

인젝션 같은 보안 취약점이 가득한 상태가 됩니다. 이런 코드는 추후에 기능 하나만 수정하려 해도 전체 앱이 붕괴되는 대참사를 낳습니다.

이때 필요한 것이 바로 AI를 '깐깐하고 엄격한 시니어 엔지니어'로 탈바꿈시키는 것입니다. AI에게 소프트웨어 오딧(Audit, 감사) 스킬과 TDD(테스트 주도 개발) 스킬을 장착시켜 줍니다. TDD란 코드를 짜기 전에 '실패하는 테스트 코드(Red)'를 먼저 작성하고, 그 테스트를 통과하기 위한 '최소한의 코드만 작성(Green)'한 뒤, 코드를 예쁘게 다듬는 '리팩토링(Refactor)'을 반복하는 최고의 소프트웨어 공학 기법입니다. AI가 이 루프를 자율적으로 돌게 만들면, 에러가 없고 유지보수가 완벽한 엔터프라이즈급 소프트웨어를 얻을 수 있습니다.

〔사전 준비 및 환경 설정〕

커스텀 스킬(Skill.md) 정의: 프로젝트의 루트 폴더에 에이전트가 따를 지침서인 TDD_Skill.md와 Security_Audit.md를 작성해 둡니다. (예: "항상 Red-Green-Refactor 루프를 따를 것", "보안 취약점 스캔 시 OAWSP 기준을 따를 것").

그릴 미(Grill Me) 스킬 장착: AI가 무턱대고 코딩을 시작하지 못하도록, 개발 전 나에게 끊임없이 예리한 질문을 던져 아키텍처를 견고하게 설계하게 만드는 Grill_Me.md 스킬을 설정합니다.

〔초보자를 위한 구체적인 에이전트 지시 프롬프트〕 이미 엉망으로 짜인 auth.js(사용자 로그인) 코드가 있다고 가정해 봅시다.

"현재 내 프로젝트의 전체 코드베이스에 대해 '시니어 엔지니어 아키텍처 오딧(Audit)'과 '보안 취약점 스캔'을 수행해 줘.

1단계. 코드 및 보안 감사 (Audit):

auth.js를 비롯한 전체 파일에서 단일 책임 원칙(SOLID)을 위반하는 갓 오브젝트(God Object, 너무 많은 일을 하는 코드)나 스파게티 코드가 없는지 스캔해 줘.

하드코딩된 비밀번호, API 키 유출, 또는

SQL

인젝션이 가능한 취약점이 있다면 1~10점의 심각도와 함께 완벽한 마크다운 오딧 보고서로 정리해 줘.

2단계. TDD 기반 자동 리팩토링 (Red-Green-Refactor):

네가 찾아낸 심각한 결함(예: DB 연결이 라우터에 뭉쳐 있는 문제)들을 수정하기 위해, TDD_Skill.md 지침에 따라 코드를 리팩토링해 줘.

코드를 고치기 전에 반드시 실패하는 유닛 테스트(Unit Test) 코드를 먼저 작성하고 실행해서 에러(Red)를 확인해.

그다음, 테스트를 통과할 수 있도록 비즈니스 로직과 서비스 계층을 깔끔하게 분리(Green)하여 코드를 작성해 줘.

수정이 끝나면 Playwright를 통해 실제 브라우저 환경에서 작동하는지 E2E(End-to-End) 테스트까지 스스로 구동하여 자체 검증 루프를 완수해 줘."

〔AI의 작업 과정 및 마법 같은 결과〕

1단계: 가차 없는 팩트 폭행(오딧 결과): 에이전트는 단 2분 만에 코드를 샅샅이 뒤지고 보고서를 출력합니다. "심각도 10/10: auth.js 파일 하나에 데이터베이스 쿼리, JWT 토큰 생성, 라우팅이 모두 뭉쳐 있는 거대한 스파게티 코드입니다. 이메일 입력란에

SQL

인젝션 공격이 가능합니다." 초보자가 짠 허술한 코드의 치부를 완벽하게 진단해 냅니다.

2단계: TDD 무한 루프 구동: 에이전트는 결함을 고치기 위해 스스로 행동합니다. 먼저 test_auth.js를 만들고 해커의 공격을 시뮬레이션하는 테스트를 돌려 실패(Red)하는 것을 확인합니다. 그 후, AI는 코드를 3개의 파일(route.js, auth_service.js, db_repository.js)로 예쁘게 쪼개어 계층화(Abstraction)합니다.

3단계: 테스트 통과 및 최종 승인: 쪼개진 코드로 다시 테스트를 돌리자 완벽하게 통과(Green)합니다. 마지막으로 E2E 브라우저 테스트 툴(Playwright)을 백그라운드에서 띄워 가상의 사용자가 아이디/비밀번호를 치고 로그인하는 과정까지 에러 없이 통과시킵니다.

이러한 과정을 통해, 초보자가 주먹구구식으로 만든 '위험한 쓰레기 코드'가 실리콘밸리의 시니어 엔지니어 팀이 몇 주간 심혈을 기울여 만든 듯한 '완벽하고 안전한 엔터프라이즈급 아키텍처'로 단 20~30분 만에 마법처럼 재탄생하게 됩니다.

# Codex 구체적 활용사례 37 - 제35장. 거대한 PDF 및 책 데이터를 챕터별로 자동 분할 및 요약 처리: 400MB 문서의 장벽을 부수는 AI

제35장. 거대한 PDF 및 책 데이터를 챕터별로 자동 분할 및 요약 처리: 400MB 문서의 장벽을 부수는 AI

〔초보자를 위한 개념 이해: 왜 챗GPT 웹사이트에서는 안 될까?〕 대학원생, 연구원, 법률가, 혹은 방대한 매뉴얼을 다루는 직장인이라면 수백 페이지, 용량이 수백 MB에 달하는 거대한 PDF 파일을 마주할 때가 있습니다. 이 파일을 챗GPT나 Claude 웹사이트에 업로드하려고 하면 "파일 용량이 너무 큽니다(제한 30MB)"라는 경고 메시지와 함께 업로드가 거부됩니다. 설령 텍스트를 복사해서 넣으려 해도 AI의 '컨텍스트 윈도우(기억력 한계)'를 초과하여 중간에 멈춰버리죠.

하지만 내 컴퓨터에 설치하여 로컬 파일에 직접 접근할 수 있는 Claude Co-work나

Codex

같은 데스크톱 에이전트를 사용하면 이 한계가 완전히 사라집니다. AI 에이전트는 거대한 400MB짜리 PDF 파일을 스스로 읽어 들이고, 목차(목차 페이지 혹은 헤딩 텍스트)를 논리적으로 파악한 뒤, 이를 챕터별로 쪼개어(Chunking) 각각의 독립된 PDF나 텍스트 파일로 저장해 줍니다. 나아가 쪼개진 각 챕터별로 핵심 요약본까지 만들어 폴더에 예쁘게 정리해 주는 마법 같은 워크플로우입니다.

〔사전 준비 및 환경 설정〕

작업 폴더 생성: 바탕화면에 PDF_Master라는 빈 폴더를 만듭니다. (Claude Co-work에서는 이를 'Playground' 폴더로 지정하면 안전합니다).

거대 파일 준비: 이 폴더 안에 여러분이 분할하고 싶은 거대한 파일, 예를 들어 2026_Global_Economic_Report_400MB.pdf를 넣어둡니다.

에이전트 실행 권한: 에이전트가 로컬 파일에 접근하고 파이썬 스크립트(PyPDF2 라이브러리 등)를 실행할 수 있도록 파일 읽기/쓰기 권한을 허용해 둡니다.

〔초보자를 위한 구체적인 에이전트 지시 프롬프트〕

"현재 폴더에 있는 2026_Global_Economic_Report_400MB.pdf라는 거대한 PDF 파일을 처리해 줘.

다음 단계에 따라 순차적으로 진행해: 1. 논리적 구조 파악: 문서를 스캔해서 목차(Table of Contents)와 자연스러운 챕터 구분선(자연스러운 섹션 브레이크)을 찾아내 줘. 2. 챕터별 파일 분할(Chunking): 파이썬 스크립트를 스스로 작성하고 실행해서, 이 거대한 PDF를 챕터별로 쪼개어 줘. 그리고 폴더 안에 Chapters라는 하위 폴더를 만들어서 '01_Introduction.pdf', '02_Market_Trends.pdf'처럼 직관적이고 설명적인 파일 이름으로 각각 저장해 줘. 3. 개별 요약본 생성: 파일 분할이 완료되면, 쪼개진 각 챕터 파일을 다시 읽고 1장 분량의 핵심 요약(Executive Summary)을 마크다운(.md) 파일로 작성해 줘. 한눈에 찾기 쉽도록 모든 작업이 끝나면 나에게 요약 리스트를 보고해 줘."

〔AI의 작업 과정 및 마법 같은 결과〕 프롬프트를 전송하면 에이전트는 파일의 크기에 압도당하지 않고 침착하게 파이썬 스크립트를 작성하여 문서를 분석하기 시작합니다.

약 1분 뒤: "문서에서 총 12개의 핵심 챕터를 발견했습니다. 파일 분할을 시작합니다."

약 3분 뒤: 바탕화면의 Chapters 폴더에 12개의 가벼운 PDF 파일이 완벽하게 쪼개져서 저장됩니다.

약 5분 뒤: 각 챕터 옆에 01_Introduction_Summary.md라는 요약본이 자동으로 생성됩니다.

이제 여러분은 무거운 PDF 뷰어를 켜고 스크롤을 끝없이 내릴 필요 없이, 에이전트가 완벽하게 분류해 둔 챕터별 요약본만 빠르게 읽어보고 필요한 원문만 쏙쏙 빼서 볼 수 있습니다. 문서 작업의 효율성이 수십 배 상승하는 순간입니다.

# Codex 구체적 활용사례 37 - 제36장. 유튜브 댓글 데이터 감성 분석 및 맞춤형 대시보드 구축: 고객의 목소리를 데이터 앱으로 만들기

제36장. 유튜브 댓글 데이터 감성 분석 및 맞춤형 대시보드 구축: 고객의 목소리를 데이터 앱으로 만들기

〔초보자를 위한 개념 이해: 엑셀을 넘어선 Streamlit 데이터 대시보드〕 유튜브 크리에이터나 마케터들은 영상에 달린 수백 개의 댓글을 읽으며 시청자의 반응을 파악합니다. 하지만 댓글이 200개, 500개가 넘어가면 이를 다 읽고 "이 영상은 긍정적인 반응이 70%고, 주로 오디오 문제에 대한 불만이 많네"라고 정량적으로 분석하는 것은 불가능합니다.

일반적으로 이를 엑셀로 뽑아내는 것까지는 상상할 수 있지만, 한 걸음 더 나아가 Streamlit(스트림릿)이라는 파이썬 라이브러리를 쓰면, 여러분의 컴퓨터에서 즉시 실행되는 '나만의 인터랙티브 웹 대시보드 앱'을 단 몇 분 만에 만들 수 있습니다. 스트림릿은 파이썬 코드 몇 줄만으로 복잡한 파이 차트, 필터 기능, 데이터 테이블을 갖춘 웹사이트를 만들어주는 강력한 도구입니다. AI에게 댓글 수집부터 감성 분석, 그리고 대시보드 코딩까지 한 번에 지시해 보겠습니다.

〔사전 준비 및 환경 설정〕

작업 폴더 생성: Youtube_Dashboard 폴더를 만듭니다.

API 및 환경 준비: 에이전트에게 유튜브 API 키(또는 웹 스크래핑 도구 권한)를 제공하여 댓글을 수집할 수 있도록 환경 변수(.env)에 세팅합니다. 컴퓨터에 파이썬이 설치되어 있어야 합니다.

〔초보자를 위한 구체적인 에이전트 지시 프롬프트〕

"나의 유튜브 채널에서 최근에 올린 영상의 최신 댓글 200개를 자동으로 수집(Scraping)하고, 이를 감성 분석하여 Streamlit 대시보드 데이터 앱으로 구축해 줘.

다음 단계를 완벽하게 코딩하고 실행해: 1. 데이터 수집 및 감성 분석: 파이썬 스크립트를 작성해 유튜브 댓글 200개를 가져와 줘. 그리고 각 댓글을 '피드백(Feedback)', '질문(Question)', '불만 사항(Complaint/Bug)' 3가지 카테고리로 분류하고, 긍정/부정/중립의 감성 분석 점수를 매겨서 comments_data.csv로 로컬에 저장해 줘. 2. Streamlit 대시보드 구축: 이 CSV 데이터를 시각적으로 보여주는 app.py라는 Streamlit 웹 앱 코드를 작성해 줘. 대시보드에는 다음이 포함되어야 해.

상단: 전체 긍정 비율 및 총 댓글 수 요약 카드 (Summary Scorecard)

중앙: 카테고리별(피드백/질문/불만) 분포를 보여주는 파이 차트

하단: 특정 키워드나 부정적인 댓글만 골라볼 수 있는 검색 및 필터 기능이 달린 데이터 테이블. 3. 실행: 코드가 완성되면 터미널에서 streamlit run app.py를 자동으로 실행해서 내 브라우저에 대시보드를 띄워 줘."

〔AI의 작업 과정 및 마법 같은 결과〕

데이터 분석: 에이전트는 즉시 유튜브 API를 호출하여 댓글 200개를 긁어오고, 자연어 처리(NLP)를 통해 "영상 내용이 너무 유익해요"는 긍정/피드백으로, "소리가 너무 작아서 안 들려요"는 부정/불만 사항으로 분류하여 CSV 파일로 저장합니다.

대시보드 구동: 2~3분 뒤 에이전트가 app.py를 완성하고 스스로 터미널에서 streamlit run app.py 명령어를 입력합니다.

결과 확인: 여러분의 크롬 브라우저가 자동으로 열리며, 로컬 웹 호스팅 화면(localhost:8501)에 기가 막힌 데이터 대시보드가 뜹니다. 화면에는 "긍정 반응 65%, 불만 사항 12개"라는 예쁜 요약 카드와 함께 화려한 파이 차트가 그려져 있습니다. 검색창에 "불만"을 체크하면 오디오 문제를 지적하는 댓글 12개만 테이블에 깔끔하게 필터링되어 나타납니다. 개발자와 데이터 분석가를 동시에 고용해야만 얻을 수 있었던 수백만 원짜리 결과물을, 프롬프트 한 번과 단 5분의 대기 시간으로 얻게 되는 경이로운 순간입니다.

# Codex 구체적 활용사례 37 - 제37장. 대규모 에이전트 군단(Agent Swarm)을 쓴 자동 이력서 작성 및 일자리 검색: 취업 준비의 판도를 바꾸다

제37장. 대규모 에이전트 군단(Agent Swarm)을 쓴 자동 이력서 작성 및 일자리 검색: 취업 준비의 판도를 바꾸다

〔초보자를 위한 개념 이해: 1명의 AI를 넘어선 300명의 AI 군단 (

Agent Swarm

)〕 지금까지 우리는 1명의 똑똑한 AI 비서(Agent)에게 일을 시키는 방법을 배웠습니다. 하지만 최신 AI 모델(예: Kimi 2.6 등)과 에이전트 프레임워크는 여기서 한 차원 더 진화한 '에이전트 군단(

Agent Swarm

)' 기술을 제공합니다.

에이전트 군단이란, 여러분이 하나의 지시를 내리면 AI 메인 매니저가 이 지시를 분석한 뒤 수십에서 최대 300명의 하위 에이전트(Sub-agents)를 동시에 생성하여 병렬로 투입하는 기술입니다. 취업 준비를 예로 들어보겠습니다. 이력서를 고치고, 구직 사이트를 검색해서 나와 맞는 회사 100곳을 찾고, 각 회사의 인재상에 맞춰 내 이력서와 커버레터(자기소개서)를 100개 버전으로 수정하는 일은 사람의 손으로는 몇 주, 몇 달이 걸리는 피 말리는 작업입니다. 하지만 에이전트 군단을 쓰면 이 모든 과정을 단 20분 만에 끝낼 수 있습니다.

〔사전 준비 및 환경 설정〕

마스터 파일 준비: Job_Search 폴더를 만들고 여러분의 기본 이력서 파일(base_resume.pdf 또는 .md)과 여러분의 경력, 희망 연봉, 선호하는 업무 형태(원격, 출퇴근 등)가 적힌 preferences.md 파일을 넣어둡니다.

Swarm 지원 플랫폼 사용: 여러 에이전트를 동시에 병렬로 가동할 수 있는 Kimi K2.6,

Codex

다중 워크트리, 혹은 Claude의 서브 에이전트(Sub-agents) 생성 기능을 지원하는 환경을 준비합니다.

〔초보자를 위한 구체적인 에이전트 지시 프롬프트〕

"현재 내 폴더에 있는 base_resume.pdf와 preferences.md(원격 근무, 프론트엔드 개발자, 희망 연봉 8천만 원 등)를 읽고, 나를 위한 '대규모 취업 지원 에이전트 군단(

Agent Swarm

)'을 가동해 줘.

다음 과정을 병렬(Parallel) 에이전트로 처리해: 1. 대규모 리서치: 먼저 10개의 리서치 서브 에이전트를 파견해서 링크드인(LinkedIn)이나 인디드(Indeed) 같은 구직 플랫폼에서 내 조건과 완벽하게 일치하는 열린 채용 공고(Job Openings) 100개를 스크래핑하고 찾아내 줘. 2. 100명의 전담 에이전트 생성: 찾은 100개의 회사 각각에 대해 전담 하위 에이전트 1명씩(총 100명)을 동시에 배정해 줘. 3. 맞춤형 서류 작성(초개인화): 각 에이전트는 자기가 맡은 회사의 채용 공고(JD)와 회사의 최근 뉴스를 딥 리서치한 뒤, 내 기본 이력서의 경험을 그 회사의 인재상에 맞게 재배치하여 초개인화된 맞춤형 이력서와 커버레터(Cover Letter)를 자동으로 작성해 줘. 4. 결과물 정리: 작업이 완료되면 Applications라는 폴더 내에 100개 회사의 이름으로 각각 하위 폴더를 만들고, 그 안에 완성된 맞춤형 이력서와 지원 사이트 링크를 예쁘게 정리해서 저장해 줘."

〔AI의 작업 과정 및 마법 같은 결과〕 지시를 내리는 순간 화면에는 경이로운 광경이 펼쳐집니다.

마스터 에이전트가 "100개의 병렬 작업 에이전트를 배포합니다(Deploying agents)"라는 메시지와 함께 수십 개의 터미널 스레드가 동시에 핑핑 돌아가기 시작합니다.

각 에이전트는 "A회사 리서치 중...", "B회사 채용공고 분석 중...", "C회사 맞춤형 이력서 생성 중..." 이라는 로그를 뿜어내며 완벽한 병렬 처리(Parallel Processing)로 100개의 작업을 동시에 수행합니다.

약 20~25분 후 모든 에이전트 군단이 복귀하여 작업을 종료합니다.

폴더를 열어보면 100개의 회사 이름으로 된 폴더가 생겨 있습니다. A회사 폴더에 들어가 커버레터를 열어보면, "A회사가 최근 런칭한 AI 프로젝트에 제 프론트엔드 경험이 정확히 일치합니다"라는 소름 돋게 구체적인 초개인화 자기소개서가 들어있습니다. 여러분은 그저 정리된 엑셀 대시보드의 입사 지원 링크를 누르고, 에이전트가 만들어준 맞춤형 이력서 첨부 파일을 클릭하여 제출하기만 하면 됩니다. 혼자서 일자리를 구하는 것이 아니라, 수백 명의 HR 전문가와 비서를 한 번에 고용하여 취업 시장을 폭격하는 궁극의 확장(Scaling) 사용법입니다.

이 책을 잘 읽으셨으면 그리고 새로운 가치있는 지식을 얻으셨다고 판단되시면
농협 302-1096-0948-81 (예금주 김경진) 에 자발적 후원 부탁드립니다.