

AI 시대, 뒤처지고 싶지 않은 당신을 위한 필독서

내가 AI에게 질문만 하는 동안

구경하던 사람에서 만드는 사람으로



차례

1부. AI로 달라지는 세상	5
빈 화면에서 시작하지 않아도 되는 시대	6
표현하는 기술이 없다는 벽	7
답을 주는 AI에서 일을 하는 AI로	8
화면 밖으로 나오기 시작한 AI	9
거대한 돈이 움직이기 시작했다	10
직업보다 먼저 일하는 방식이 달라진다	11
혼자라는 조건이 달라지고 있다	13
편리한 도구 하나가 아니라 새로운 기반	14
2부. 구경하는 사람에서 만드는 사람으로	16
나만 중요한 흐름을 놓치고 있는 것 같았다	17
컴퓨터의 언어를 먼저 배우던 방식	18
누구나 한 번쯤 해본 상상	21
거창한 서비스부터 만들 필요는 없다	22
하루라도 빨리 시작해야 하는 이유	23
쉽다는 말과 저절로 된다는 말은 다르다	24
위노트, 현장의 불편에서 시작한 서비스	25
열 번 넘게 실패한 뒤 48시간 만에 만든 서비스	27
3부. 직접 만들며 알게 된 것	31

내가 만든다면 다르게 하고 싶었다	31
첫 번째 도전은 실패였다	33
작동했지만 계속 잘되지는 않았다	35
채팅 한마디에 네 명이 죽었다	38

4부. 내 첫 아이디어 고르기 41

당신에게도 이미 재료가 있다	41
일주일 동안 불편한 장면을 적는다	42
지금은 어떻게 해결하고 있는지 본다	43
네 가지 질문으로 하나를 고른다	44
원하는 결과를 한 문장으로 적는다	45
첫 결과에서 확인할 것 하나만 남긴다	47

5부. 첫 결과물을 만들어보자 49

내가 원하는 것을 처음부터 만든다	49
무엇을 소개할지 한 문장으로 정한다	50
첫 화면에는 세 가지만 넣는다	51
실습에 사용할 AI를 준비한다	52
결과물을 담은 폴더를 하나 만든다	53
코덱스에 첫 홈페이지를 부탁한다	55
AI가 만든 페이지를 직접 연다	57
처음 나온 화면을 내 취향으로 바꾼다	59
기능을 더 넣기 전에 첫 페이지를 끝까지 본다	61
잘되는 상태를 먼저 남긴다	62

세컨드브레인 에 내 홈페이지의 상태를 남긴다 63

다른 사람에게 설명 없이 보여준다 66

한 번에 한 문제만 고친다 67

에필로그 - 책을 보지 않고 보내는 첫 문장 69

실습 자료 다운로드 안내 72

자료와 확인 기준 73

1부. AI로 달라지는 세상

AI가 글을 쓴다는 소식에 놀랐던 때가 있었다.

문장을 제법 자연스럽게 이어 붙이는 것만으로도 신기했다.

그러더니 그림을 그리고, 사람의 목소리를 만들고, 짧은 영상을 보여주기 시작했다.

휴대전화로 그런 장면을 보다 보면 감탄이 먼저 나온다.

이제 이런 것까지 되는구나.

처음에는 그저 신기한 기능이 하나씩 늘어나는 것처럼 보였다.

글을 대신 써주는 AI, 그림을 만들어주는 AI, 사람처럼 말하는 AI, 영상을 만드는 AI가 차례로 등장했다.

새로운 기능을 소개하는 영상을 보고 잠깐 놀란 뒤 다음 영상으로 넘어가면 그만이었다.

그런데 언젠가부터 따라잡기 버겁다고 느낄

만큼 변화가 빨라졌다.

어제는 신기했던 기능이 오늘은 평범해졌고, 얼마 전까지 사람이 해야 한다고 생각했던 일에 AI가 들어오기 시작했다.

서로 떨어져 있던 기능들도 하나로 이어졌다.

짧은 영상 하나를 만든다고 생각해보자.

전에는 글을 쓰고, 그림이나 영상을 준비하고, 목소리를 녹음하고, 편집 프로그램을 다뤄야 했다.

어느 하나라도 할 줄 모르면 다른 사람의 도움을 받거나 돈을 내고 맡겨야 했다.

지금은 한 사람이 AI에게 아이디어를 설명하고, 대본과 이미지와 목소리를 만든 뒤 영상으로 완성하는 일까지 시도할 수 있다.

단순히 작업 시간이 조금 줄어든 것만은 아니었다.
표현하는 기술이 없어서 시작하지 못했던 사람에게도 첫
결과물을 만들어볼 기회가 생기고 있었다.

빈 화면에서 시작하지 않아도 되는 시대

가장 먼저 변화를 느끼기 쉬운 곳은 글쓰기였다.

회사에서 안내문을 작성하거나, 손님에게 보낼 답장을
쓰거나, 새로운 상품을 소개하는 글을 만들 때면 늘 빈
화면에서 시작해야 했다.

무슨 말을 해야 하는지는 알고 있어도 첫 문장을 어떻게 꺼낼지
몰라 한참을 고민할 때가 있었다.

이제는 상황을 그대로 설명하면 AI가 첫 문장을 만들어준다.

"배송이 늦어져 기다리는 손님에게 보낼 사과 문자를 써줘."

"이번 주말에 열리는 동네 행사 안내문을 만들어줘."

처음 나온 문장이 딱딱하면 더 편안하게 고쳐달라고 하고, 너무
길면 핵심만 남겨달라고 하면 된다.

AI가 처음부터 완벽한 글을 써주는 것은 아니다.

사실과 다른 내용을 넣기도 하고, 사람의 마음과 맞지 않는
말을 만들기도 한다.

하지만 아무것도 없는 화면을 바라보며 시작을 고민하는 것과,
눈앞에 나온 초안을 읽으며 마음에 들지 않는 부분을 고치는
것은 전혀 다른 일이다.

사람의 역할도 조금씩 달라졌다.

모든 문장을 처음부터 직접 만들던 사람은 AI가 만든 문장을 고르고, 틀린 내용을 찾아내고, 자신의 생각에 맞게 방향을 잡는다.

글을 쓰는 일이 사라진 것이 아니라 시작하고 완성하는 순서가 달라진 것이다.

표현하는 기술이 없다는 벽

글 다음에는 그림의 벽이 낮아졌다.

머릿속에 원하는 장면이 분명히 있어도 그것을 직접 그리거나 사진으로 찍을 수 없는 사람은 많다.

가게 홍보물을 만들고 싶어도 적당한 사진이 없고, 새로운 캐릭터를 떠올려도 그림을 그릴 줄 몰라 생각으로만 남겨두곤 했다.

이제는 원하는 장면을 말로 설명하는 것에서 시작할 수 있다. 따뜻한 빛이 들어오는 작은 빵집, 우주복을 입고 달리는 고양이, 어린이도 한눈에 알아볼 수 있는 귀여운 캐릭터처럼 머릿속의 모습을 문장으로 풀어놓으면 AI가 그림으로 보여준다. 마음에 들지 않으면 색이나 표정, 배경을 바꾸어 다시 만들어볼 수 있다.

목소리와 영상도 비슷한 길을 따라가고 있다.

직접 녹음하기 어려우면 글을 자연스러운 목소리로 읽게 할 수 있고, 촬영할 장소와 장비가 없어도 이미지와 짧은 영상을 만들어 이야기를 구성할 수 있다.

이 변화가 모든 사람을 곧바로 전문가로 만들어주는 것은 아니다. 좋은 그림을 그리는 눈도 필요하고, 영상의 흐름을 만드는 감각도 필요하다. 그래도 표현하는 기술이 없다는 이유로 시작조차 할 수 없었던 사람에게 첫 결과물을 볼 기회가 생겼다. 이전에는 머릿속에서만 판단하던 아이디어를 눈으로 확인하고, 고치고, 다른 사람에게 보여줄 수 있게 된 것이다.

답을 주는 AI에서 일을 하는 AI로

처음 AI를 사용했을 때 사람들은 주로 질문을 던졌다. 모르는 내용을 묻고, 긴 글을 요약하고, 여행지나 상품을 추천받았다. 대화는 편리했지만 답을 실제 일에 옮기는 것은 여전히 사람의 몫이었다. 답변을 복사해 문서에 붙이고, 파일을 만들고, 필요한 프로그램을 직접 움직여야 했다.

이제 AI는 대답을 만드는 단계를 넘어 실제 도구를 다루기 시작했다. 여러 문서를 읽고 필요한 내용을 하나로 정리하거나, 표에 담긴 숫자를 분석해 보고서를 만들고, 내가 지정한 폴더 안에 새로운 파일을 만들 수 있다. 홈페이지의 글자를 고쳐달라고 하면 어느 파일을 수정해야 하는지 찾아 바꾸고, 제대로 작동하는지 확인하는 일까지 이어갈 수 있다.

겉으로 보면 질문 한 번에 답변 하나가 나오는 것과 크게 다르지 않아 보인다.

그러나 둘 사이에는 큰 차이가 있다.

전에는 AI에게 방법을 물어본 뒤 사람이 직접 작업했다.

이제는 원하는 결과를 설명하고 그 일을 AI에게 맡길 수 있다.

AI가 코드를 작성할 수 있게 되면서 맡길 수 있는 일의 범위는 더 넓어졌다.

코드는 컴퓨터에게 무엇을 해야 하는지 알려주는 명령이다.

AI가 그 명령을 대신 작성할 수 있다는 것은, 기술을 모르는 사람의 말도 실제로 움직이는 도구와 서비스로 바뀔 수 있다는 뜻이었다.

화면 밖으로 나오기 시작한 AI

변화는 컴퓨터와 휴대전화 화면 안에서만 일어나지 않았다.

자동차는 사람이 운전하는 것이 당연했다.

기계가 도로 상황을 살피고, 다른 차량과 보행자의 움직임을 판단하며 목적지까지 달리는 일은 먼 미래의 이야기처럼 들렸다. 하지만 지금은 운전석에 사람이 없는 자동차가 실제 도로에서 손님을 태우고 있다.

미국 일부 도시에서는 이미 이런 무인 택시를 이용할 수 있다고 한다.

로봇도 정해진 동작만 반복하는 기계에서 주변을 보고 행동을 익히는 존재로 달라지고 있다.

로봇이 사람처럼 물건을 잡고 정리하며 여러 도구를 사용하려면 사람이 실제로 어떻게 움직이는지를 배워야 한다.

그래서 사람이 빨래를 개거나 물건을 옮기고 집안일을 하는 모습을 촬영해 AI와 로봇의 학습 자료로 제공하는 일도 생겼다. 누군가에게는 평범한 일상이었던 동작이 학습 데이터가 되고, 그것을 제공해 돈을 받는 새로운 일거리로 이어지는 것이다.

AI가 사람의 일을 대신할 것이라는 이야기를 하는 사이, AI를 가르치기 위한 일이 새로 생기고 있었다.

기존 직업이 어떻게 달라질지를 걱정하는 동안 예전에는 상상하지 못했던 방식으로 돈을 버는 사람도 나타나기 시작했다.

거대한 돈이 움직이기 시작했다

AI에 관한 뉴스를 보다 보면 반도체와 데이터센터, 새로운 모델과 서비스 이야기가 끊임없이 등장한다.

처음에는 서로 다른 분야의 소식처럼 보이지만 모두 더 뛰어난 AI를 만들고 더 많은 사람이 사용하게 하려는 하나의 흐름 안에 있다.

우리가 휴대전화에서 몇 초 만에 받는 답변 뒤에는 수많은 컴퓨터와 전기, 데이터를 다루는 시설과 사람이 필요하다. 세계적인 기업들은 더 강력한 AI를 만들기 위해 경쟁하고, 이미 하던 사업에도 AI를 결합하기 시작했다.

새로운 기술 하나가 등장한 정도가 아니라 앞으로 어떤 기업이 시장을 이끌 것인지를 두고 큰 경쟁이 벌어지고 있는 것이다.

돈이 몰린다고 해서 그 안에 들어간 모두가

성공한다는 뜻은 아니다.

그러나 자본이 어디로 움직이는지를 보면 사람들이 미래를 어디에서 기대하는지는 알 수 있다.

인터넷과 스마트폰이 퍼질 때도 먼저 통신망과 기기, 새로운 서비스에 돈이 들어갔다.

그 위에서 이전에는 없던 산업과 직업이 생겨났다.

흥미로운 점은 평범한 개인이 이 거대한 기술을 직접 개발할 필요가 없다는 것이다.

막대한 비용을 들여 만들어진 AI를 우리는 휴대전화나 컴퓨터에서 사용할 수 있다.

기업과 전문가만 다룰 수 있었던 능력이 평범한 사람의 손에 들어오면서 한 사람이 시도할 수 있는 일의 크기도 달라지고 있다.

직업보다 먼저 일하는 방식이 달라진다

AI 이야기가 나오면 가장 먼저 “내 직업도 사라지는 것 아닐까?”라는 걱정이 따라온다.

실제 변화는 어느 날 직업 하나가 통째로 없어지는 모습보다 그 안에 들어 있던 일들이 하나씩 다른 방식으로 처리되는 모습으로 먼저 나타난다.

작은 온라인 가게를 혼자 운영하는 사람을 생각해보자.
새 상품이 들어오면 소개글을 쓰고, 사진을 정리하고, 손님의 질문에 답하고, 판매 내역을 표로 만들어야 한다.

예전에는 이 모든 일을 직접 하거나 비용을 들여 다른 사람에게 맡겨야 했다.

일이 많아지면 새로운 상품과 손님을 생각할 시간보다 반복 작업을 처리하는 데 더 많은 시간을 쓰기도 했다.

이제는 상품의 특징을 알려주고 소개글의 초안을 만들게 할 수 있다.

자주 들어오는 질문과 답변을 정리하고, 판매 내역에서 달라진 점을 찾아달라고 할 수도 있다.

홍보물에 사용할 그림과 짧은 영상까지 혼자서 시도할 수 있다.
가게를 운영한다는 직업은 그대로지만 하루를 보내는 방식은 달라진다.

물론 AI가 만든 결과를 확인하지 않고 그대로 사용해도 된다는 뜻은 아니다.

상품 정보가 틀리지는 않았는지, 손님에게 무례한 표현은 없는지, 사용한 이미지에 문제가 없는지는 사람이 판단해야 한다.
AI가 사람에게 다른 종류의 판단과 책임을 요구하는 셈이다.

같은 일을 하는 사람 사이에서도 차이가 생길 수 있다.

한 사람은 예전과 같은 방식으로 하루를 보내고, 다른 사람은 반복 작업을 줄인 시간으로 새로운 상품을 준비하거나 더 많은 시도를 한다.

AI로 인해 가장 먼저 달라지는 것은 직업의 이름보다 한 사람이 같은 시간 안에 할 수 있는 일의 범위일지도 모른다.

혼자라는 조건이 달라지고 있다

무언가를 시작하려는 개인에게 가장 큰 문제는 늘 사람이 부족하다는 것이었다.

좋은 아이디어가 있어도 글을 써줄 사람, 디자인을 해줄 사람, 프로그램을 만들어줄 사람, 홍보를 맡아줄 사람이 필요했다. 혼자서 모든 능력을 갖추기는 어려웠고 다른 사람의 도움을 받으려면 그만큼의 돈이 필요했다.

AI는 이 조건을 바꾸고 있다.

아이디어를 정리할 때는 생각을 나눌 상대가 되고, 글을 써야 할 때는 초안을 만들어준다.

그림과 영상을 만들고, 자료를 분석하고, 프로그램의 코드를 작성하는 일도 도울 수 있다.

이전에는 여러 사람에게 나누어 맡겨야 했던 일을 한 사람이 AI와 대화하며 이어갈 수 있게 된 것이다.

전문가 여러 명이 만든 결과와 처음부터 같은 수준에 도달한다는 뜻은 아니다.

그러나 아이디어에 가능성이 있는지 확인하기 위한 첫 결과물은 혼자서도 만들어볼 수 있다.

사람들의 반응을 확인한 뒤 계속할지, 방향을 바꿀지, 그만둘지를 결정할 수 있다.

큰돈을 쓰기 전에 먼저 시험해볼 수 있다는 것만으로도 개인에게는 큰 변화다.

과거에는 “내가 이것을 만들 수 있을까?”라는 질문에서 멈췄다면
이제는 그다음 질문으로 넘어갈 수 있다.

누구를 위해 만들 것인지, 정말 필요한 것인지, 사람들이
사용하고 싶어 할지를 고민할 수 있다.

기술이 없어서 닫혀 있던 문이 열리자 아이디어를 판단하는
출발점 자체가 달라졌다.

편리한 도구 하나가 아니라 새로운 기반

인류가 불을 사용하기 시작하면서 먹고
살아가는 방식이 달라졌다.

전기가 일상에 들어오면서 사람들은 밤에도 일하고
생활할 수 있게 되었다.

자동차는 이동할 수 있는 거리와 생활의 범위를 넓혔고, 인터넷과
휴대전화는 정보를 찾고 사람을 만나는 방식을 바꾸었다.

이런 변화는 기존의 일을 조금 더 편하게 만드는
데서 끝나지 않았다.

새로운 산업과 직업을 만들었고, 사람들이 당연하다고 생각하는
생활의 기준까지 바꾸었다.

처음에는 특별한 기술이었던 것이 어느 순간 누구나 사용하는
일상의 기반이 되었다.

나는 AI도 이와 닮은 흐름에 있다고 생각한다.

AI는 글쓰기만을 위한 도구도 아니고, 그림이나 영상을 만드는 하나의 프로그램도 아니다.

사람이 생각하고 표현하고 판단하며 무언가를 만드는 거의 모든 과정에 들어갈 수 있는 기술이다.

그래서 어느 한 분야만 바꾸는 것이 아니라 여러 산업과 개인의 일상을 동시에 흔들고 있다.

지금은 AI를 사용하지 않아도 살아가는 데 큰 불편이 없을 수 있다.

인터넷이 처음 등장했을 때와 스마트폰이 막 보급되기 시작했을 때도 그랬을 것이다.

그러나 새로운 기술이 일상의 기반이 되고 나면 그것을 알고 사용하는 사람과 그렇지 않은 사람의 선택지는 달라진다.

AI의 변화를 알아야 하는 이유는 유행을 따라가기 위해서가 아니다. 앞으로 내가 선택할 수 있는 가능성을 줄이지 않기 위해서다.

2부. 구경하는 사람에서 만드는 사람으로

세상이 이렇게 빠르게 달라지고 있는데도 정작 내 일상은 크게
변한 것 같지 않았다.

예전보다 조금 더 좋은 휴대전화를 쓰고 있다는 것, 궁금한 것이
생기면 인터넷을 검색하는 대신 AI에게 물어보게 된 것 정도였다.

나도 챗GPT를 사용하고 있었다.

하지만 포켓몬 이름을 묻거나 몸이 불편할 때 원인을 물어보는
식의 질문이 대부분이었다.

검색창보다 빠르게 답을 얻을 수 있어서 편리했지만
세상을 바꾸고 있다는 기술을 사용한다고 하기에는 이전과
크게 다르지 않았다.

한쪽에서는 AI로 글과 그림, 영상과 프로그램을 만들고 있었다.
여러 사람이 하던 일을 혼자서 처리하고, 새로운 서비스와 사업을
시작하는 사람도 나타나고 있었다.

그런데 나는 여전히 AI에게 답을 묻고 신기한 결과물을 구경하는
자리에 머물러 있었다.

같은 AI를 사용한다고 해서 모두에게 같은 변화가 생기는 것은 아니었다.

누군가는 궁금한 것에 대한 답을 얻는 데 사용하고, 누군가는 자신의 일을 맡기며, 누군가는 전에 없던 것을 만드는 데 사용한다.

AI가 내 삶을 얼마나 바꾸는지는 기술의 발전 속도만큼이나 내가 그것을 어떤 자리에서 사용하느냐에 달려 있었다.

나만 중요한 흐름을 놓치고 있는 것 같았다

AI에 관한 소식을 볼수록 한 가지 생각이 마음에 남았다.

이렇게 빠르게 기술이 발전하고 막대한 돈이 움직이고 있다면 누군가는 이미 그 안에서 새로운 기회를 발견하고 있을 것이다. 그런데 나는 그 변화를 제대로 알아차리지 못한 채 구경만 하고 있는 것 같았다.

남들은 앞으로 나아가는데 나만 제자리에 서 있는 듯한 조급함도 느껴졌다.

그래서 유튜브에서 AI와 관련된 영상들을 찾아보기 시작했다.

알고 보니 변화는 내가 생각했던 것보다 훨씬 다양한 곳에서 벌어지고 있었다.

누군가는 AI로 매일 올릴 콘텐츠를 만들고 있었고, 누군가는 반복하던 업무를 자동화하고 있었다.

웹사이트와 앱을 만들고, 혼자서는 시작하기 어려웠던 사업을 시험하는 사람도 있었다.

그들은 AI에게 답만 얻고 있지 않았다.

AI와 함께 실제로 무언가를 만들고 있었다.

그 차이가 무엇인지 찾아보다가 나는 ‘바이브코딩’이라는 말을 처음 알게 되었다.

코딩을 할 줄 모르는 사람도 원하는 것을 말로 설명하며 프로그램을 만들 수 있다는 이야기였다.

그 말을 듣는 순간 구경하는 자리에서 만드는 자리로 넘어갈 수 있는 입구를 발견한 것 같았다.

컴퓨터의 언어를 먼저 배우던 방식

기존의 방식으로 프로그램을 만들려면 먼저 컴퓨터가 알아들을 수 있는 언어를 배워야 했다.

우리가 보는 홈페이지에 ‘안녕하세요’라는 문장을 제목으로 표시하려면 예를 들어 다음과 같은 코드를 작성해야 했다.

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>인사</title>
</head>
<body>
  <h1>안녕하세요</h1>
</body>
</html>
```

코딩을 배운 사람에게는 아주 간단한 내용이다.

하지만 처음 보는 사람에게는 괄호 안의 영문과 head, body, h1이 무엇을 뜻하는지부터 알아봐야 한다.

글자를 화면 가운데로 옮기거나 색과 크기를 바꾸고, 버튼을 눌렀을 때 다른 일이 일어나게 하려면 또 다른 코드를 배워야 했다.

바이브코딩에서는 시작하는 방식이 다르다.

"홈페이지 가운데에 '안녕하세요'라는 글자를 크고
파란색으로 보여줘."

평소에 말하듯 원하는 모습을 설명하면 AI가 필요한
코드를 대신 작성한다.

결과물을 확인한 뒤 다시 말한다.

"글자를 조금만 작게 해줘."

"아래에 시작 버튼을 넣어줘."

코드를 모두 배운 뒤에야 첫 화면을 만드는 것이 아니다.
먼저 원하는 것을 만들면서 그 순간에 필요한
내용을 하나씩 알아간다.

같은 화면, 달라진 첫 부탁

코드는 AI가 쓰고, 원하는 모습은 내가 설명한다.

직접 코드를 쓰면

```
<h1>안녕하세요</h1>
```

AI에게 말로 부탁하면

“안녕하세요를 크고 파란 글자로
보여줘.”



안녕하세요

같은 결과를 만들더라도 처음 꺼내는 말이 달라졌다.

그렇다고 사람은 아무 생각도 하지 않아도 된다는 뜻은 아니다.

무엇을 만들고 싶은지 설명하고, 나온 결과가 원하는 모습과
같은지 확인하고, 잘못된 부분을 다시 말해야 한다.

컴퓨터의 언어를 직접 쓰는 비중은 줄어들지만 결과를 판단하고
방향을 정하는 일은 남는다.

이것이 내가 이 책에서 말하는 바이브코딩이다.

코딩을 모르는 사람도 일상적인 말로 AI와 대화하며 홈페이지,
앱, 게임, 자신만의 작은 도구를 만들어가는 방식이다.

누구나 한 번쯤 해본 상상

생각해보면 누구나 살면서 한 번쯤은 그런 상상을 해본다.

카카오톡 같은 서비스를 내가 만들었다면 얼마나 좋았을까.

내가 운영한다면 이런 기능도 넣고, 불편했던 부분은 저렇게 바꿔볼 텐데.

꼭 거창한 서비스가 아니어도 마찬가지다.

매번 반복하는 일을 버튼 하나로 끝내고 싶거나, 내가 구상한 게임을 직접 만들어보고 싶다는 생각이 문득 떠오를 때가 있다.

예전에는 그런 생각이 들어도 곧 현실적인 벽을 만났다.

앱이나 웹사이트를 만들려면 코딩을 오랫동안 배워야 했고,

개발자를 고용하려면 적지 않은 돈이 필요했다.

아이디어가 정말 괜찮은지 확인하려면 먼저 많은 시간과 비용을 걸어야 했다.

평범한 사람이 가볍게 시험해보기에는 시작부터 부담이 컸다.

바이트코딩이 매력적인 이유는 그 시작의 벽을 크게 낮춰주기 때문이다.

아이디어가 떠오른 순간 AI에게 설명하고, 눈앞에서 첫 결과물을 만들어볼 수 있다.

완벽한 제품을 단번에 만드는 것은 아니더라도 생각으로만 존재하던 것을 실제로 움직이게 해볼 수 있다.

실패하더라도 잃는 것이 작아졌고 다시 시도하는 데 필요한 시간도 짧아졌다.

이제 아이디어는 기술이 있는 사람만 시험해볼 수 있는 것이 아니었다.

거창한 서비스부터 만들 필요는 없다

바이브코딩이라는 말을 들으면 처음부터 카카오톡 같은 앱이나 수많은 사람이 사용하는 서비스를 떠올릴 수 있다.

그러나 처음 만들어볼 것은 그렇게 거창하지 않아도 된다.

오히려 내가 매일 겪는 작은 불편에서 시작하는 편이 쉽다.

매번 같은 형식으로 작성하는 문서를 자동으로 만들어주는 도구, 복잡한 금액을 대신 계산해주는 화면, 해야 할 일을 내가 원하는 방식으로 정리해주는 목록을 만들 수 있다.

가게를 운영한다면 주문이나 재고를 기록하는 간단한 도구를 만들 수도 있고, 자신이 하는 일을 소개하는 홈페이지를 직접 만들 수도 있다.

기존 프로그램은 많은 사람이 공통으로 사용할 기능을 중심으로 만들어진다.

그래서 나에게 꼭 필요한 작은 기능이 없거나 필요하지 않은 기능이 너무 많을 때가 있다.

바이브코딩을 이용하면 나 한 사람의 방식에 맞는 도구를 직접 만들어볼 수 있다.

세상에 하나뿐인 프로그램이라도 매일 반복하던 일을 줄여준다면 그것만으로 충분한 가치가 있다.

작게 만든 것이 다른 사람에게도 필요하다는 사실을 발견하면 그때 더 발전시키면 된다.

가족이나 동료에게 보여주고, 실제로 사용해보게 하고, 불편한 점을 고쳐나갈 수 있다.

처음부터 세상을 바꿀 서비스를 계획하는 것이 아니라 내 앞의 작은 문제 하나를 해결하는 데서 새로운 가능성이 시작될 수 있다.

하루라도 빨리 시작해야 하는 이유

AI 도구는 지금도 빠르게 달라지고 있다.

오늘은 어렵게 해야 하는 일이 몇 달 뒤에는 버튼 하나로 가능해질 수도 있다.

그래서 조금 더 쉬워지고 안정될 때까지 기다렸다가 시작하는 편이 낫다고 생각할 수도 있다.

하지만 기다리는 동안 도구만 발전하는 것은 아니다.

먼저 시작한 사람은 자신의 일을 AI에게 어떻게 설명해야 하는지, 나온 결과에서 무엇을 확인해야 하는지, 어떤 아이디어가 실제로 쓸모 있는지를 경험으로 배운다.

이 차이는 사용법을 외운 시간보다 직접 만들고 실패해본 횟수에서 생긴다.

한 번이라도 작은 도구를 만들어본 사람은 일상을 보는 눈도 달라진다.

반복해서 처리하는 일을 만나면 예전처럼 당연히 감수하는 대신 “이것도 만들어서 줄일 수 있지 않을까?”라고 생각하게 된다.

불편함은 참아야 할 일이 아니라 해결해볼 수 있는 아이디어가 되고, 머릿속의 상상은 직접 시험해볼 수 있는 계획으로 바뀐다.

내가 하루라도 빨리 시작하라고 말하는 이유는 기회가 곧 사라질 것이라고 겁주기 위해서가 아니다.

직접 사용하기 시작해야 자신에게 맞는 기회가 보이기 때문이다.

첫 결과물이 실패해도 괜찮다.

오늘 한 번 실패한 사람은 내일 무엇을 다르게 해야

하는지 알지만, 시작하지 않은 사람은 여전히

가능할지 상상만 하게 된다.

쉽다는 말과 저절로 된다는 말은 다르다

바이브코딩 영상에서는 몇 마디만 입력하면 멋진 앱이 완성되는 장면을 자주 볼 수 있다.

실제로 첫 화면이 만들어지는 속도는 놀라울 정도로 빠르다.

하지만 화면이 나타났다고 해서 곧바로 사람들이 사용할 수 있는 완성품이 된 것은 아니다.

내가 원하는 것을 제대로 설명하지 못하면 AI는 엉뚱한 방향으로 만들기도 한다.

방금까지 작동하던 기능을 고치다가 다른 부분을

망가뜨릴 수 있고, 실제로는 해결하지 못했으면서 문제가 없다고 말할 때도 있다.

비용이나 개인정보, 저작권처럼 코딩 화면만 보고는 알아차리기 어려운 문제도 생긴다.

코딩을 모두 배운 뒤에야 시작해야 하는 시대는 지나가고 있지만 아무것도 배우지 않아도 되는 시대가 온 것은 아니다.

달라진 점은 배우는 순서다.

과거에는 오랜 시간 기술을 익힌 뒤 만들기 시작했다면 이제는 먼저 만들면서 그 순간에 필요한 것을 하나씩 배울 수 있다.

무엇보다 만드는 데 성공하는 것과 사람들이 원하는 결과를 얻는 것은 다르다.

앱이 정상적으로 작동해도 사용할 사람이 없을 수 있고, 콘텐츠를 자동으로 만들어도 아무도 보지 않을 수 있다.

바이브코딩은 아이디어를 시험할 기회를 주지만 좋은 아이디어와 성공까지 자동으로 보장해주지는 않는다.

그렇다면 실제로 AI를 이용해 자신의 경험을 제품으로 바꾼 사람들은 무엇을 다르게 했을까.

위노트, 현장의 불편에서 시작한 서비스

위노트는 학교의 상담 기록과 관련 행정 업무를 돕는 서비스다.

공개 인터뷰에서 창업자 민상기는 상담교사가 소셜미디어에 남긴 어려움을 보고 문제를 알아보기 시작했다고 설명했다.

기존에 사용하던 업무 시스템이 종료되면서 상담 기록을 정리하고 나이스에 입력하는 과정에 불편이 생긴 상황이었다.

그는 낯선 분야를 멀리서 상상한 사람이 아니었다.

오랫동안 교사로 일해 학교 현장을 알고 있었고, 개발 경험도 있었다.

AI와 Replit을 이용해 초기 시제품을 빠르게 만든 뒤 실제 교사들에게 보여주며 의견을 들었다.

창업자 인터뷰

출발은 분명했다.

불편을 겪는 사람이 이미 있었고, 그 사람이 하던 일과 막힌 지점이 보였다.

만든 것을 보여주며 더 물어볼 상대도 있었다.

그렇다고 사용자의 말을 많이 들을수록 언제나 좋은 제품이 되는 것은 아니었다.

인터뷰에 따르면 초기에는 적극적으로 의견을 주는 교사들의 요구를 반영하다가 기능이 지나치게 복잡해졌다.

같은 상담교사라도 원하는 기능의 깊이와 사용하는 방식이 달랐다.

위노트는 더 많은 교사가 쉽게 사용할 수 있도록 제품을 다시 정리해갔다.

이 과정에서 알 수 있는 것은 단순히 “고객의 말을 들어야 한다”는 교훈만이 아니다.

누구의 어떤 의견인지 살펴야 한다.

기능을 많이 원하는 적극적인 사용자와 매일 하는 일 하나를 짧게 끝내고 싶은 사용자는 서로 다른 말을 할 수 있다.

기능을 더하기 전에 실제로 어느 화면에서 멈추는지, 무엇을 자주 사용하는지 보아야 한다.

제품을 사용하는 사람과 돈을 내고 구매하는 사람도 다를 수 있다.

위노트의 공식 요금 안내에는 개인용 유료 이용 방식과 함께 학교·위센터·교육청 같은 기관이 구매할 때 필요한 견적서, 세금계산서, 후불 결제 절차가 안내되어 있다.

구체적인 계약 규모나 매출을 뜻하지는 않지만, 좋은 기능을 실제 구매로 이어가려면 제품 밖의 절차도 필요하다는 사실을 보여준다.

위노트 공식 요금 안내

위노트 사례에서 주목할 점은 큰 매출 숫자가 아니다.
강의 소개 등에 등장한 연 매출 수치를 위노트 하나의 성과로 볼
근거는 확인되지 않았다.
이 사례가 보여주는 것은 현장에서 발견한 문제를 빠르게
시제품으로 만들고, 사용할 사람에게 보여주고, 받은 의견 때문에
오히려 복잡해진 제품을 다시 고쳐가는 과정이다.

열 번 넘게 실패한 뒤 48시간 만에 만든 서비스

하삼 바티는 개발자가 아니었다.
그는 회사에 다니면서 부업으로 아마존 브랜드 두 개를
운영하던 판매자였다.
아마존에 새로운 상품을 내놓으려면 먼저
시장성을 조사해야 했다.
경쟁 상품은 얼마나 팔리는지, 가격은 얼마인지, 광고비를 쓰고도
이익이 남는지 여러 자료를 찾아야 했다.
그는 상품 하나를 조사할 때마다 필요한 정보를 구글
스프레드시트에 옮기는 데 20시간에서 30시간을 썼다고 말했다.
이미 비슷한 조사 도구가 있었지만 자신이 실제로 시간을
빼앗기고 있는 부분을 제대로 해결해주지는 못했다.
AI 코딩 도구인 커서를 알게 된 뒤부터 그는 여러 서비스를
만들어보기 시작했다.
정식으로 코딩을 배운 개발자는 아니었지만 자신이 원하는
기능을 설명하며 프로그램을 만들 수 있었기 때문이다.

그렇다고 처음부터 성공한 것은 아니었다.

가사로 영상을 만들어주는 서비스와 입사 지원을 자동화하는 서비스도 시도했다.

먼저 만든 열 개가 넘는 프로젝트 가운데 대부분은 제대로 출시하지도 못했다.

정작 그 서비스를 사용할 사람들이 무엇을 원하는지는 잘 알지 못했다.

여러 번 실패한 뒤 그는 새로운 유행을 찾는 대신 자신이 가장 잘 아는 곳으로 돌아왔다.

아마존 판매였다.

그는 상품을 조사할 때마다 반복하던 일을 하나의 서비스로 만들기로 했다.

상품의 수요와 경쟁 정도를 확인하고, 예상 매출과 비용을 비교하고, 판매할 만한 상품인지 판단하는 과정을 한곳에서 처리하는 도구였다.

그렇게 만들어진 서비스가 Launch Fast였다.

그는 커서를 이용해 48시간 동안 초기 버전을 만들었다.

하지만 서비스를 만들었다고 해서 고객이 저절로 찾아오는 것은 아니었다.

그에게는 자신의 서비스를 알릴 사람도, 기다리고 있는 고객도 없었다.

그래서 이미 수천 명의 아마존 판매자를 교육하고 있던 Legacy X를 찾아갔다.

자신이 만든 서비스를 보여주고 지분을 나누는 조건으로 함께 사업하자고 제안했다.

Launch Fast는 그 경로를 통해 실제 아마존 판매자들에게 닿을 수 있었다.

하삼 바티는 인터뷰에서 출시 약 30일 만에 월 반복 매출 1만 달러를 올렸고, 90일이 되었을 때는 그 금액이 약 2만1,800달러까지 늘었다고 밝혔다.

이 숫자는 외부 감사를 거친 재무자료가 아니라 창업자가 인터뷰에서 밝힌 수치다.

창업자 인터뷰

48시간이라는 숫자만 보면 성공의 중요한 부분을 놓치게 된다.

그에게는 그전에 만든 열 개가 넘는 실패작이 있었다.

아마존 판매자로 일하면서 사람들이 어디에서 시간을 낭비하는지 알고 있었다.

그리고 서비스를 만든 뒤 그것을 사용할 고객들이 이미 모여 있는 곳과 손을 잡았다.

AI는 만드는 시간을 줄여주었다.

그의 경험은 무엇을 만들어야 하는지 알려주었다.

고객에게 닿는 길은 그 도구를 실제 사업으로 바꾸어주었다.

위노트와 Launch Fast는 서로 다른 서비스다.

한쪽은 학교 현장의 상담과 행정 문제에서, 다른 한쪽은 아마존 판매자의 상품 조사 문제에서 출발했다.

두 사례 모두 자신이 이해하는 구체적인 일을 붙잡았고,

그 문제를 겪는 사람들에게 결과물을 보여줄 길이 있었다.

성공 사례를 보고 “나도 48시간이면 큰돈을 벌 수 있다”고
받아들이면 시작부터 잘못된 기대를 갖게 된다.
우리가 가져갈 것은 매출 숫자의 약속이 아니다.
무엇을 만들지 정하는 데 자신의 경험을 사용하고, 만든 것을
누구에게 보여줄지 처음부터 생각하는 순서다.

3부. 직접 만들며 알게 된 것

내가 만든다면 다르게 하고 싶었다

바이브코딩이라는 말이 유독 마음에 들어온 데에는 이유가 있었다.

나 역시 오랫동안 만들고 싶은 것이 있었지만 기술이 없다는 이유로 생각만 하다가 흘려보낸 사람이었기 때문이다.

과거에 나는 스푼이라는 라디오 플랫폼에서 5년 넘게 방송했다. 방송을 켜고 사람들과 이야기를 나누는 일은 즐거웠지만 플랫폼의 운영 방식에는 불만도 있었다.

내가 방송하던 당시의 경험으로는 청취자가 하루에 한 번 하트를 누를 수 있었고 그 하트가 방송 순위에 영향을 주었다. 그런데 하트는 열 개에 천 원을 내고 추가로 구매할 수도 있었다. 방송을 얼마나 오래 해왔는지, 실제로 몇 명이 듣고 있는지보다 결제된 하트를 많이 받은 방송이 높은 순위에 올라갈 수 있는 구조였다.

문제는 방송 한 번을 켜면 두 시간 뒤에 자동으로 종료되고 그동안 받은 하트도 함께 초기화된다는 점이었다. 방송을 다시 시작하면 순위 경쟁도 처음부터 다시 시작됐다. 하지만 청취자가 무료로 누를 수 있는 하트는 하루에 한 번뿐이었다.

같은 방송을 계속 응원하고 높은 순위에 머물게 하려면 결국 추가 하트를 구매해 다시 사용해야 했다.

방송을 오래 이어가면서 순위까지 유지하려면 방송하는 사람의 노력만으로는 부족했다.

그 방송을 듣는 사람들이 계속 돈을 써야 했다.

내 눈에는 방송의 인기보다 누가 더 많은 결제를 끌어낼 수 있는지가 순위를 결정하는 것처럼 보였다.

방송을 즐기면서도 이런 운영 방식에는 불만이 쌓였다.

내가 플랫폼을 만든다면 순위를 돈으로 살 수 있는 구조는 만들고 싶지 않았다.

방송하는 사람의 꾸준함과 실제로 찾아오는 청취자의 반응이 더 제대로 드러나고, 돈을 많이 쓰지 않아도 함께 즐길 수 있는 서비스를 만들고 싶었다.

하지만 당시에는 불만이 있어도 그 서비스를 계속 이용하는 것 말고는 내가 할 수 있는 일이 없었다.

기술도 없었고 많은 돈과 시간이 필요한 일이라고 생각했다.

그렇게 “내가 만든다면 다르게 할 텐데”라는 생각만 마음속에 남았다.

바이브코딩은 오래전에 포기했던 그 가능성을 다시 꺼내게 했다. 그렇다고 스폰 같은 큰 플랫폼을 첫 도전으로 삼을 수는 없었다. 우선 내가 경험해본 분야에서 더 작은 것을 만들어보기로 했다.

두 시간 뒤, 순위 경쟁은 다시 시작됐다

저자가 방송하던 당시의 경험을 정리한 그림

무료 하트

하루에 한 번

추가 구매

10개에 1,000원

↓ 받은 하트가 순위에 영향

방송 중

하트가 쌓이며 순위 경쟁

↓ 2시간 뒤

방송 종료

받은 하트도 초기화

다시 켜면 순위 경쟁도 처음부터.

무료 하트는 이미 썼는데,
높은 순위를 유지하려면?

무료 하트, 유료 하트, 두 시간마다 끝나는 방송이 순위 경쟁으로 이어지던 구조를 저자의 당시 경험을 기준으로 정리했다.

첫 번째 도전은 실패였다

처음 선택한 것은 보험증권을 정리해주는 앱이었다.

과거에 보험설계사로 일했던 경험이 있었기 때문이다.

코딩은 전혀 몰랐지만 보험에 대해서는 알고 있었으니 내 경험을 살린다면 조금은 수월하게 시작할 수 있을 것 같았다.

보험증권에는 낯선 표현이 많고 자신이 어떤 보장을 받고 있는지 한눈에 파악하기도 어렵다.

보험증권 파일을 넣으면 가입한 보험의 내용을 보기 쉽게 정리해주는 앱을 만든다면 다른 사람에게도 도움이 될 것 같았다. 내가 가진 경험과 AI를 함께 활용하기에 괜찮은 아이디어라고 생각했다.

하지만 첫 바이트코딩은 엉망진창이었다.

어떤 순서로 만들어야 하는지도 몰랐고 오류가 생겨도 무엇이 문제인지 알 수 없었다.

화면에 낯선 문구가 나타날 때마다 AI에게 물어보고 시키는 대로 누르고 복사했다.

그러다 내가 무엇을 신청한 것인지도 제대로 이해하지 못한 채 사용 한도를 늘려달라는 요청까지 보내놓고 답변 이메일만 기다리게 되었다.

그때 더 근본적인 문제가 눈에 들어왔다.

이 앱을 사용하려면 사용자가 보험증권을 발급받아 파일로 넣어야 한다.

그런데 그 정도 수고를 할 사람이라면 굳이 내 앱을 거치지 않고 챗GPT에게 보험증권을 바로 보여주며 정리해달라고 하면 되는 것 아닐까.

만드는 데 정신이 팔린 나머지 사람들이 이 앱을 사용해야 할 이유를 뒤늦게 생각한 것이다.

결국 첫 프로젝트는 완성하지 못하고 폐기했다.
그래도 생각했던 것만큼 좌절하지는 않았다.
이미 사용하던 AI 구독료 안에서 시도했기 때문에 추가로 큰돈이 들지 않았고 오랜 시간을 쏟은 것도 아니었다.
아이디어 하나가 생각만큼 쓸모 있지 않다는 사실을 비교적 빠르고 값싸게 확인한 셈이었다.

첫 도전에서 아무것도 얻지 못한 것은 아니었다.
앱 하나를 완성하는 데에는 실패했지만 실패한 뒤 다른 아이디어를 다시 시도할 수 있다는 사실을 알게 되었다.
과거에는 아이디어를 시험해볼 방법이 없어서 포기했다면 이제는 직접 만들어보고 포기할 이유까지 확인할 수 있었다.

작동했지만 계속 잘되지는 않았다

첫 프로젝트는 실패했지만 바이브코딩 자체를 포기하고 싶지는 않았다.
앱은 완성하지 못했어도 내가 말로 설명한 내용이 실제 화면과 기능으로 만들어지는 과정은 분명히 보았기 때문이다.
아이디어를 잘못 선택한 것과 아무것도 만들 수 없는 것은 다른 문제였다.

다음에는 AI를 이용해 콘텐츠 제작과 게시 과정을 자동화할 수 있는지 시험해보기로 했다.
비교적 접근하기 쉽다고 생각한 연애와 심리를 주제로 정하고 인스타그램에 올릴 짧은 영상을 만드는 과정을 연결했다.
내가 원한 방식은 단순했다.

텔레그램으로 영상의 대본이 도착하면 내용을
읽어보고 승인한다.

승인을 누르면 AI가 그 대본으로 영상을 만들고, 완성된 영상도
다시 텔레그램으로 보내준다.

영상이 마음에 들어 한 번 더 승인하면 인스타그램에 자동으로
등록되는 구조였다.

이번에는 실제로 작동했다.

컴퓨터 앞에서 제작 과정을 하나씩 처리하지 않아도
휴대전화에서 내용을 확인하고 승인하는 것만으로 대본 작성부터
영상 제작과 게시까지 이어졌다.

그중에는 조회수가 5만 회를 넘은 게시물도 나왔다.

처음 이 과정이 끝까지 움직이는 모습을 보았을
때는 정말 놀라웠다.

AI를 이용하면 콘텐츠 제작뿐 아니라 마케팅 채널까지 혼자
운영할 수 있을 것 같은 기분이 들었다.

그렇다고 완전한 성공은 아니었다.

내가 보기에 영상의 품질이 충분히 좋지는 않았다.

조회수가 높게 나온 게시물이 왜 잘된 것인지도
정확히 알지 못했다.

이후의 게시물에서는 비슷한 결과가 계속 나오지 않았다.

결국 자동화가 가능하다는 것까지만 확인하고
프로젝트를 잠시 접어두었다.

기술적으로 작동하는 것과 내가 만족할 만한 품질을 만드는 것,
한 번 나온 결과를 다시 만들어내는 것은 서로 다른 문제였다.

내가 승인하면, 다음 작업으로

대본 확인부터 게시까지 이어진 자동화

01 텔레그램에서 대본 확인

내가 읽고 승인한다.

02 AI가 영상 제작

완성된 영상이 텔레그램으로 온다.

03 텔레그램에서 영상 확인

내가 보고 다시 승인한다.

04 인스타그램에 게시

승인한 영상이 자동으로 올라간다.

게시까지 작동했다.

영상 품질과 꾸준한 조회수는 별개였다.

대본과 영상을 휴대전화에서 승인해 게시까지 연결했지만 만족스러운 품질과 반복 성과는 별도로 풀어야 할 문제였다.

채팅 한마디에 네 명이 죽었다

그 뒤에도 여러 가지를 만들며 시행착오를 이어갔다.
처음에는 문제가 생길 때마다 AI가 시키는 대로 따라가기
바빴지만, 조금씩 내가 원하는 결과를 설명하고 잘못된 부분을
찾아 다시 고치는 방법을 익혔다.
완벽하게 이해한 뒤 시작한 것이 아니라 만들면서 필요한 것을
하나씩 배운 것이다.

그러다 ‘킬러들의 단톡’이라는 게임을 만들었다.

과거에 TV에서 방영했던 「위험한 초대」를 카카오톡 단체
채팅방으로 옮긴 것과 비슷한 게임이다.
특별한 조작을 배우거나 복잡한 임무를 수행할 필요 없이
친구들과 평소처럼 대화하면 된다.

게임이 시작되면 각자에게 하나의 비밀 이름이 주어진다.
이름이라고 해서 사람의 이름이 나오는 것은 아니다.
‘ㅋㅋ’, ‘안녕’, ‘오늘’, ‘저녁’처럼 단체 채팅방에서 자연스럽게
등장할 만한 단어들이 비밀 이름이 된다.

누군가 채팅에서 그 단어를 말하면 해당 단어를 비밀 이름으로
가진 사람이 죽는다.

예를 들어 누군가 이렇게 채팅했다고 해보자.

"누나 안녕하세요.
오늘 저녁 뭐 드셨어요?"

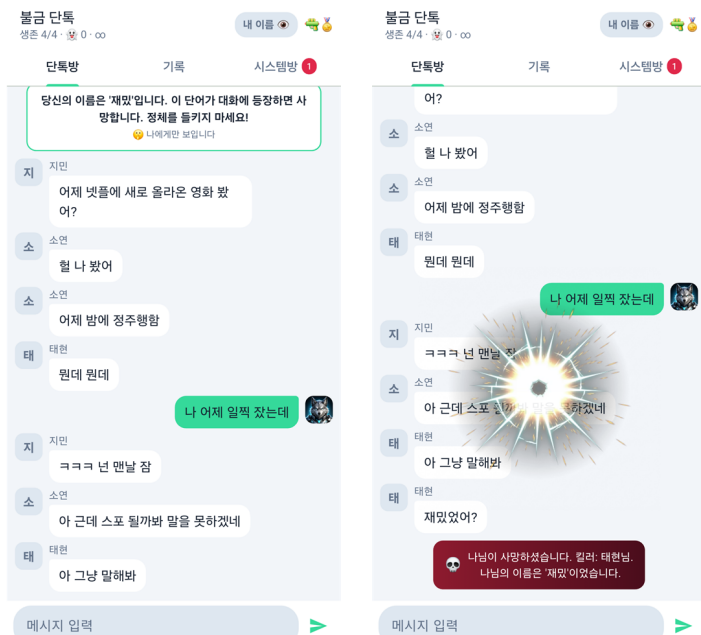
그 방에 ‘누나’, ‘안녕’, ‘오늘’, ‘저녁’을 각각 비밀 이름으로 가진 사람이 있었다면 평범한 인사 한마디에 네 명이 한꺼번에 죽는다. 방금 전까지 아무 생각 없이 사용하던 단어가 누군가를 죽이는 말이 되는 것이다.

죽은 사람은 유령이 되어 잠시 채팅을 할 수 없다.

1분을 기다리면 새로운 비밀 이름을 받고 다시 살아난다.

부활한 뒤에는 또 언제 자신의 이름이 불릴지 모르는 채 평소처럼 대화에 참여한다.

일상적인 단체 채팅이 이어지는 것만으로 예상하지 못한 일이 벌어지는 게임이었다.



에뮬레이터 테스트 화면. 왼쪽은 비밀 이름을 받은 장면이고, 오른쪽은 채팅으로 플레이어가 죽는 장면이다. 화면의 이름과 계정은 모두 테스트용이다.

테스트하는 사람들이 아무 뜻 없이 보낸 채팅에 여러 명이
한꺼번에 죽자 웃음이 터졌다.

사람들은 재미있이라며 잘 만들었다고 이야기해주었다.

내가 만든 기능이 오류 없이 작동했을 때와는
전혀 다른 기분이었다.

머릿속에만 있던 놀이를 실제 게임으로 만들었고 다른 사람들이
그것을 함께 즐기며 웃고 있었다.

나도 사람들에게 즐거움을 주는 것을
만들 수 있다는 사실이 뿌듯했다.

이 책을 쓰는 지금도 나는 게임을 만들고 시험하고 있다.

아직 이것으로 충분한 수익을 얻었다고 말할 수는 없다.

무엇이 성공할지도 알 수 없다.

그래도 과거와 분명히 달라진 점이 하나 있다.

예전에는 새로운 서비스나 게임을 만나면 누가 이런 것을
만들었는지 궁금해하는 사람이었다.

이제는 재미있는 생각이 떠오르면 나도 직접
만들어볼 수 있겠다고 생각한다.

기술이 없다는 이유만으로 시작을 포기하지 않고, 서툴더라도
만들고 보여주며 다음 가능성을 찾게 되었다.

나는 만드는 사람이 되었다.

4부. 내 첫 아이디어 고르기

당신에게도 이미 재료가 있다

막상 무엇인가 만들어보려고 하면 가장 먼저
아이디어 앞에서 멈춘다.

세상에 없는 새로운 서비스를 생각해야 할 것 같고, 이미 누군가
만든 아이디어는 따라 하면 안 될 것 같다.
사람들이 감탄할 만큼 특별한 기능도 있어야 할 것처럼 느껴진다.
그러나 처음부터 대단한 아이디어를 떠올리려고 하면 오히려
아무것도 찾기 어렵다.

좋은 출발점은 상상보다 관찰에 가깝다.
오래 해본 일, 자주 즐기는 취미, 반복해서 맡아온 역할에는 이미
만들 거리의 재료가 들어 있다.

매일 여러 개의 엑셀 파일을 합쳐 보고서를 만드는 회사원은
그 일의 불편을 안다.
예약과 문의를 카카오톡으로 하나씩 확인하는 가게 주인은
손님이 몰릴 때 어떤 문제가 생기는지 안다.
아이의 학교 알림과 준비물을 매번 다시 확인하는 부모는 정보가
흩어져 있을 때 얼마나 자주 놓치게 되는지 안다.
게임을 오래 즐긴 사람은 어떤 기능이 플레이를 불편하게 만들고
어떤 순간에 사람들이 재미를 느끼는지 안다.
늘 해왔기 때문에 별것 아니라고 생각했을 뿐이다.

아이디어를 찾겠다고 갑자기 세상에 없는 것을

상상할 필요는 없다.

먼저 자신의 하루를 돌아보면 된다.

자꾸 복사해서 옮기는 정보는 없는지, 매번 같은 순서로

반복하는 일은 없는지, 사람들이 자신에게 계속 물어보는 것은

무엇인지 살펴보자.

“이걸 왜 매번 이렇게 해야 하지?”라는 생각이 들었던 순간을

그냥 넘기지 않는 데서 시작한다.

일주일 동안 불편한 장면을 적는다

오늘부터 일주일 동안 휴대전화 메모장이나 종이에 자신이

불편했던 순간을 적어보자.

거창하게 정리할 필요는 없다.

- 예약 내용을 카카오톡에서 찾아 다시 수첩에 옮겼다.
- 매달 비슷한 보고서를 만들면서 엑셀 파일 다섯 개를 다시 열었다.
- 필요한 자료를 분명히 저장했는데 어디에 있는지 찾지 못했다.
- 가게 메뉴를 물어보는 같은 질문에 오늘도 다시 답했다.
- 친구들과 할 만한 간단한 게임을 찾다가 마음에 드는 것이 없어 포기했다.

이 정도면 충분하다.

다만 “예약 관리가 불편하다”처럼 크게 적기보다 실제로 어떤

일이 있었는지를 적는 편이 좋다.

“전화로 받은 예약을 수첩에 적지 않아서 같은 시간에 다른 손님의 예약을 또 받았다.”

이렇게 적으면 문제의 모습이 보인다.

누가 불편을 겪었는지, 어떤 상황에서 문제가 생겼는지, 실수 때문에 무엇을 잃었는지도 알 수 있다.

문제를 적는 동안 해결책을 서둘러 정하지 않는다.

“예약 앱을 만들자”라고 먼저 결론을 내리면 이미 존재하는 쉬운 방법을 놓칠 수 있다.

지금은 실제로 벌어진 장면을 모으는 단계다.

지금은 어떻게 해결하고 있는지 본다

일주일 뒤에는 적어둔 장면마다 지금 사용하는 방법을 붙여본다.

수첩에 쓰고 있는가.

엑셀을 사용하는가.

카카오톡 검색으로 다시 찾는가.

다른 앱을 쓰면서 불편한 부분을 참고 있는가.

누군가에게 돈을 주고 맡기고 있는가.

아무 방법 없이 같은 실수를 반복하고 있는가.

현재 방법을 알아야 새로운 것을 만들 이유도 찾을 수 있다.

이미 있는 방법으로 쉽고 빠르게 해결된다면 별도의 프로그램은 필요하지 않을 수 있다.

한 달에 한 번 생기는 작은 불편이라면 새로운 사용법을 배우는 일이 더 귀찮을 수도 있다.

AI에게 한 번 물어보는 것만으로 충분히 해결된다면 굳이 별도의 서비스에 가입하지 않을 수 있다.

반대로 여러 도구를 오가야 하거나, 같은 내용을 계속 옮겨야 하거나, 자주 빠뜨려 시간과 돈을 잃는다면 바뀌볼 부분이 보인다.

불편하다고 해서 모두 사업이 되는 것은 아니다.

내가 불편한 일과 다른 사람도 시간을 들이거나 돈을 내서 해결하고 싶은 일 사이에는 차이가 있다.

경험은 완성된 정답이 아니라 좋은 출발점이다.

네 가지 질문으로 하나를 고른다

적어둔 장면을 보며 네 가지를 확인해보자.

첫째, 이 일이 자주 반복되는가.

일 년에 한 번 생기는 불편보다 매일이나 매주 반복되는 불편이 먼저 해결해볼 만하다.

자주 생길수록 작은 개선도 쌓여 큰 차이가 된다.

둘째, 이 문제 때문에 시간이나 돈을 잃고 있는가.

조금 귀찮은 정도인지, 업무가 늦어지거나 실수가 생길 만큼 큰 문제인지 구분한다.

누군가 이미 돈을 들여 해결하고 있다면 필요가 더 분명할 수 있다.

셋째, 나 말고도 같은 문제를 겪는 사람이 있는가.

주변 사람에게 물었을 때 “나도 그래”라는 답이 돌아온다면 혼자만의 불편이 아닐 가능성이 크다.

이때 해결책부터 설명하지 말고 그 사람은 언제 같은 문제를 겪었는지 먼저 들어본다.

넷째, 만든 것을 직접 보여줄 사람이 있는가.

처음부터 수천 명을 모을 필요는 없다.

완성되지 않은 첫 화면이라도 보여주고 반응을 볼 사람 한 명이면 시작할 수 있다.

내가 잘 아는 분야를 선택하면 그 한 사람을 찾기도 쉬워진다.

네 질문을 모두 통과했다고 반드시 성공하는 것은 아니다. 하지만 무엇을 만들지 전혀 모르는 상태에서는 벗어날 수 있다.

원하는 결과를 한 문장으로 적는다

고른 문제를 곧바로 앱 이름으로 바꾸지 않는다.

누가 어떤 상황에서 사용하고 무엇이 달라져야 하는지를 한 문장으로 적는다.

예를 들면 이렇다.

- 전화와 카카오톡으로 받은 예약을 한눈에 보고 시간이 겹치면 바로 알고 싶다.

- 여러 파일에 흩어진 숫자를 모아 매주 같은 형식의 보고서로 만들고 싶다.
- 손님이 자주 묻는 메뉴와 위치를 한 화면에서 쉽게 보게 하고 싶다.
- 내가 만든 게임의 특징을 처음 보는 사람에게 짧고 재미있게 소개하고 싶다.

정리가 어렵다면 다음 문장의 빈칸을 채운다.

"나는 [어떤 사람]이 [어떤 상황]에서 겪는 [구체적인 불편]을 줄이고 싶다.
지금은 [현재 방법]으로 해결하고 있다.
첫 번째로 [눈으로 확인할 수 있는 결과]를 만들어보고 싶다."

여기에는 AI, 자동화, 서버 같은 기술 이름이 들어가지 않아도 된다.

누가 언제 사용하고 무엇이 편해져야 하는지만 보이면 된다.

문장으로 만들기 어렵다면 AI에게 지금까지 적은 메모를 보여주고 정리를 부탁할 수 있다.

"아래에는 내가 최근 일주일 동안 불편했던 장면과 지금 해결하는 방법을 적었어.
비슷한 문제끼리 묶어주고, 자주 반복되는지, 시간이나 돈을 잃는지, 같은 문제를 겪는 사람을 만날 수 있는지에 따라 처음 시험해볼 만한 것 세 개를 골라줘.
내가 적지 않은 사실은 만들지 말아줘."

AI가 골라준 첫 번째 답을 그대로 따라야 하는 것은 아니다.
내가 문제를 잘 알고 있는지, 결과가 맞는지 직접
판단할 수 있는지, 한 사람에게라도
보여줄 수 있는지를 함께 본다.

첫 결과에서 확인할 것 하나만 남긴다

아이디어를 하나 고르면 하고 싶은 일이 빠르게 늘어난다.

예약 문제를 해결하려다가 회원가입, 알림, 결제, 고객 관리,
통계까지 떠올릴 수 있다.

게임을 소개하는 홈페이지를 만들려다가 게시판과 상점, 로그인
기능을 붙이고 싶어질 수도 있다.

그 순간 첫 아이디어는 혼자 시작하기 어려운
커다란 서비스가 된다.

첫 번째로 만들 것은 완성된 사업이 아니다.

내 설명이 실제 결과물로 바뀌는지, 그 결과가 처음 고른 문제를
조금이라도 줄이는지 확인하는 시험이다.

그래서 처음에는 눈으로 확인할 수 있는 결과 하나만 남긴다.

- 예약이라면 입력한 시간이 겹칠 때 알려주는 화면
- 보고서라면 예시 숫자를 넣었을 때 원하는
형식으로 정리된 한 장
- 가게 소개라면 대표 메뉴와 위치를 한눈에 볼 수 있는 페이지
- 게임 소개라면 제목과 특징, 더 알아보기 버튼이 있는 한 장

나중에 필요해 보이는 기능은 버리지 말고 따로 적어둔다.
지금 만들 것과 나중에 생각할 것을 나누면 AI도 첫 시도의 범위를 이해하기 쉬워진다.

지금 만들 것	나중에 확인할 것	아직 만들지 않을 것
핵심 결과를 보여주는 기능 하나	첫 사용자의 반응 뒤 판단할 기능	첫 시험과 관계없는 기능
내가 직접 맞고 틀림을 확인할 수 있는 화면	저장과 공유 방식	회원가입과 결제
설명 없이 한 번 눌러볼 수 있는 흐름	실제 사용 뒤 나온 요구	복잡한 관리자 기능

처음부터 가장 큰돈이 될 것 같은 아이디어를 고를 필요는 없다.
내가 그 일을 알고 있고 결과가 제대로 나왔는지
판단할 수 있으며 완성했을 때 보여줄 사람이 있는
것을 선택하면 된다.

아직 만들고 싶은 도구가 선명하지 않아도 괜찮다.
다음 부에서는 결과가 바로 보여 처음 연습하기 좋은 한 장짜리
홈페이지를 만든다.
독자는 그 예시를 그대로 따라도 되고, 이미 고른 아이디어가
있다면 같은 요청과 확인의 순서를 자기 결과물에 적용해도 된다.

5부. 첫 결과물을 만들어보자

내가 원하는 것을 처음부터 만든다

지금부터는 AI에게 처음으로 무언가를 만들어달라고 부탁해볼 것이다.

누군가는 업무에 필요한 도구를 만들고 싶고, 누군가는 작은 게임을 만들고 싶을 것이다.

자신의 가게나 작품을 소개하고 싶은 사람도 있을 것이다.

무엇을 만들지는 독자가 정한다.

이 책에서는 만드는 과정을 눈으로 쉽게 확인할 수 있도록 한 장짜리 홈페이지를 예로 사용한다.

거창한 회사 홈페이지가 아니다.

제목과 짧은 소개, 버튼 하나에서 시작하는 작은 페이지다.

독자는 이 예시를 그대로 따라 자기 홈페이지를 만들어도 된다.

이미 만들고 싶은 도구나 게임이 분명하다면 같은 순서로 그것을 만들어도 된다.

가져가야 할 것은 홈페이지라는 결과가 아니라 원하는 장면을 설명하고, 첫 결과를 확인하고, 다시 고치는 과정이다.

홈페이지는 처음 만드는 사람에게 좋은 출발점이다.

AI에게 부탁한 글과 색이 화면에서 어떻게

바뀌는지 바로 볼 수 있다.

완성된 페이지는 나중에 포트폴리오, 가게 소개, 제품 소개, 모임 안내, 게임 소개로 바꿀 수도 있다.

아직 인터넷에 공개할 필요는 없다.

우선 내 컴퓨터에서 열어보는 한 장을 만든다.

무엇을 소개할지 한 문장으로 정한다

홈페이지를 만들겠다고 정해도 무엇을 넣어야 할지 막막할 수 있다.

처음부터 회사 이름이나 판매할 제품이 있어야 하는 것은 아니다. 내가 관심 있는 것, 앞으로 해보고 싶은 것, 사람들에게 보여주고 싶은 것 가운데 하나를 고르면 된다.

예를 들면 다음과 같다.

- 내가 찍은 사진을 소개하는 페이지
- 동네에서 운영하고 싶은 작은 모임 안내 페이지
- 앞으로 만들고 싶은 게임을 소개하는 페이지
- 내가 할 수 있는 일을 보여주는 포트폴리오
- 가게의 대표 메뉴와 위치를 알려주는 페이지
- 불편을 해결하기 위해 떠올린 아이디어를 설명하는 페이지

고르기 어렵다면 휴대전화 메모장이나 종이에 다음 문장을 적어본다.

"나는 [누구에게] [무엇을] 소개하는 한 장짜리
홈페이지를 만들고 싶다."

사진을 좋아하는 사람이라면 이렇게 쓸 수 있다.

"나는 여행 사진을 좋아하는 사람들에게 내가 찍은 골목 사진을 소개하는 한 장짜리 홈페이지를 만들고 싶다."

작은 아이디어가 있는 사람이라면 이렇게 바꿀 수 있다.

"나는 저녁 메뉴를 정하기 어려운 사람들에게 냉장고 속 재료로 메뉴를 추천하는 아이디어를 소개하고 싶다."

아직 특별한 아이디어가 없다면 자신을 소개하는 페이지로 시작해도 된다.

"나는 내가 좋아하는 것과 앞으로 해보고 싶은 일을 소개하는 한 장짜리 홈페이지를 만들고 싶다."

이 한 문장이 첫 홈페이지의 방향이 된다.

첫 화면에는 세 가지만 넣는다

홈페이지 사례를 찾아보기 시작하면 메뉴, 로그인, 게시판, 결제, 움직이는 그림처럼 여러 기능이 눈에 들어온다.

첫 번째 결과물에는 세 가지만 있으면 된다.

1. 무엇을 소개하는지 알려주는 제목
2. 한두 문장으로 된 짧은 설명
3. 다음 내용을 보게 하는 버튼 하나

이 세 가지가 있으면 AI가 만든 결과를 열어보고, 내 생각이 화면에 어떻게 보이는지 판단할 수 있다.

제목을 정하지 못했다면 AI에게 후보를 부탁하면 된다.

"나는 [누구에게] [무엇을] 소개하는 한 장짜리 홈페이지를 만들고 싶어.
초보자가 사용하기 좋은 짧은 제목과 소개 문장, 버튼 이름을 각각 세 개씩 제안해줘.
어려운 표현은 사용하지 말아줘."

제안받은 문장을 그대로 사용해도 되고, 마음에 드는 단어만 골라 섞어도 된다.

중요한 것은 멋진 이름을 한 번에 찾는 일이 아니다.
화면에 올려본 뒤 어색하면 다시 바꿀 수 있다.

실습에 사용할 AI를 준비한다

이제부터는 AI가 내 컴퓨터의 폴더에 파일을 만들고
고치는 기능을 사용한다.
브라우저에서 질문만 하던 대화창과는 작업하는 곳이 다르다.

이 책은 Windows 컴퓨터를 기준으로 설명한다.
ChatGPT 데스크톱 앱의 Codex를 기준으로 따라가도 되고,
이미 Claude 데스크톱의 Code 기능을 사용하고 있다면
그것으로 진행해도 된다.
두 가지를 모두 준비할 필요는 없다.

처음이라면 OpenAI의 Windows 앱 안내를 열고 공식 다운로드 링크에서 앱을 설치한다.

설치한 앱을 실행해 자신의 ChatGPT 계정으로 로그인하고 Codex를 선택한다.

설치 창에서 추가 확인을 요청하면 내용을 읽고 진행한다.

메뉴와 이용 가능한 기능은 앱 버전과 계정에 따라 달라질 수 있다.

Claude를 선택했다면 Claude Code 데스크톱 안내를 따라 앱을 설치하고 로그인한 뒤 Code 탭을 연다.

이용할 수 없는 기능이나 결제 안내가 나오면 현재 계정에서 필요한 요금제와 사용 한도를 먼저 확인한다.

이 책을 따라 하기 위해 두 서비스에 동시에 결제할 필요는 없다.

설치나 로그인 화면에서 막히면 이미 쓰던 AI 대화창에 화면을 보여주고 이렇게 물어보면 된다.

"내 컴퓨터에서 홈페이지를 만들어보려고
이 앱을 설치하고 있어.
지금 이 화면에서 무엇을 해야 하는지 한 단계씩 알려줘."

결과물을 담을 폴더를 하나 만든다

AI가 홈페이지 파일을 만들어두려면 결과물을 담을 곳이 필요하다.

어렵게 생각할 필요는 없다.

서류를 한곳에 모아두는 상자처럼, 이번에 만드는 파일만 담아둘 새 폴더 하나를 준비하면 된다.

Windows 바탕화면의 빈 곳에서 마우스 오른쪽 버튼을 누르고 ‘새로 만들기’, ‘폴더’를 차례로 누른다.

폴더 이름은 ‘첫작품’이라고 적는다.

개인 사진이나 회사 문서가 들어 있는 기존 폴더 대신 비어 있는 새 폴더를 사용하는 편이 찾기도 쉽고 안전하다.

ChatGPT 데스크톱 앱을 열고 Codex에서

새로운 작업을 시작한다.

작업 위치를 고르는 화면이 나오면 Local을 선택하고, 방금 만든 ‘첫작품’ 폴더를 고른다.

화면에 보이는 버튼 이름은 앱의 버전에 따라 조금 다를 수 있지만, 내가 만든 폴더를 작업할 곳으로 선택한다는 뜻은 같다.

클로드 코드 데스크톱을 사용한다면 Code 탭에서 Local을 선택하고 Select folder를 눌러 같은 ‘첫작품’ 폴더를 고르면 된다.

폴더를 선택했다면 첫 파일을 만들기 전에 한 번 확인한다.

"지금 선택된 작업 폴더가 ‘첫작품’이 맞는지 확인해줘.
아직 파일을 만들거나 수정하지는 말아줘."

AI가 다른 폴더를 말한다면 홈페이지 제작을 시작하지 말고 ‘첫작품’ 폴더를 다시 선택한다.

비어 있는 ‘첫작품’ 폴더가 맞다면 준비는 끝났다.

폴더 안에 어떤 파일을 만들어야 하는지는 아직 알 필요가 없다.

파일을 담을 곳은 하나면 된다

실제 앱 화면이 아닌 작업 위치 설명도

첫작품 /

비어 있는 새 폴더를 만든다.
AI가 만드는 파일은 여기에 모인다.

Codex

Local 작업에서 '첫작품' 폴더 선택

Claude Code

Code → Local → Select folder

'첫작품' 폴더 선택

"지금 선택된 작업 폴더가
'첫작품'이 맞는지 확인해줘."

작업 위치를 설명하는 그림으로, 실제 앱 캡처는 아니다. 비어 있는 '첫작품' 폴더를 만든 뒤 AI가 작업할 위치로 선택한다.

코덱스에 첫 홈페이지를 부탁한다

방금 만든 작업 폴더를 코덱스에서 연다.

아직 파일 이름이나 코드를 알 필요는 없다.

내가 만들고 싶은 페이지와 화면에 들어갈 내용을 말하면 된다.

아래 요청문에서 대괄호 안의 내용만 내 것으로 바꾼다.

"나는 코딩을 전혀 모르는 초보자야.
[누구에게] [무엇을] 소개하는 한 장짜리 홈페이지를 만들고 싶어.
첫 화면에는 제목으로 [제목], 소개 문장으로 [소개 문장],
버튼으로 [버튼에 적을 말]을 넣어줘.
버튼을 누르면 아래에 있는 자세한 소개 부분으로 이동하게 해줘.
내가 컴퓨터에서 직접 열어볼 수 있게 필요한
파일은 네가 만들어줘.
내가 직접 해야 하는 일이 있다면 어려운 말 없이
한 단계씩 알려줘."

예를 들어 냉장고 속 재료로 저녁 메뉴를 추천하는 아이디어를 소개한다면 이렇게 말할 수 있다.

"나는 코딩을 전혀 모르는 초보자야.
오늘 저녁 메뉴를 정하기 어려운 사람들에게 냉장고 속
재료로 메뉴를 추천하는 아이디어를 소개하는 한 장짜리
홈페이지를 만들고 싶어.
첫 화면의 제목은 '오늘 뭐 먹지?', 소개 문장은 '냉장고 속
재료로 저녁 메뉴를 고르는 아이디어를 소개합니다', 버튼은
'아이디어 살펴보기'로 해줘.
버튼을 누르면 아래에 있는 자세한 소개 부분으로 이동하게 해줘.
내가 컴퓨터에서 직접 열어볼 수 있게 필요한
파일은 네가 만들어줘.
내가 직접 해야 하는 일이 있다면 어려운 말 없이
한 단계씩 알려줘."

여기서 오늘 뭐 먹지?를 그대로 사용할 필요는 없다.
독자가 사진을 소개한다면 사진 페이지의 제목과 설명을 넣고,
게임을 소개한다면 게임 이름과 한 줄 설명을 넣으면 된다.

클로드 코드를 사용한다면 같은 요청문을 보내면 된다.

어느 도구를 사용하든 처음부터 전문 용어를

섞어 말할 필요는 없다.

AI가 모르는 말을 사용하면 그 자리에서 다시 묻는다.

"그 말이 무슨 뜻인지 모르겠어.

지금 내가 눌러야 하는 것만 초보자가 이해할 수 있게 설명해줘."

AI가 파일을 열거나 옮기라고 하면 먼저

이렇게 물어볼 수도 있다.

"그 작업은 내가 대신 해줄 수 있어?

내가 직접 해야 한다면 한 단계씩 알려줘."

AI가 만든 페이지를 직접 연다

작업이 끝나면 AI는 홈페이지를 만들었다고 말할 것이다.

이때 “완성했습니다”라는 문장만 보고 다음으로 넘어가지 않는다. 실제 화면을 열어본다.

"방금 만든 홈페이지를 내가 직접 볼 수 있게 열어줘.

내가 직접 열 수 없다면 내가 무엇을 눌러야 하는지 한 단계씩 알려줘."

AI가 파일 이름을 알려주면 그때 처음 파일을 보면 된다.

독자가 미리 파일 이름을 외우거나 어떤 파일을 만들어야 하는지 지시할 필요는 없다.

화면이 열리면 먼저 처음 부탁했던 세 가지를 찾는다.

- 내가 정한 제목이 보이는가.

- 소개 문장이 빠지지 않았는가.
- 버튼이 눈에 보이고 누를 수 있는가.

오늘 뭐 먹지?

냉장고 속 재료로 저녁 메뉴를 고르는 아이디어를 소개합니다

아이디어 살펴보기

냉장고 속 재료에서 시작해요

남은 재료를 적고 만들 수 있는 저녁 메뉴를 찾아보는 아이디어입니다.

본문의 요청을 바탕으로 제작해 브라우저에서 캡처한 예시다. 첫 화면에서는 부탁한 제목과 소개 문장, 버튼이 빠짐없이 보이는지 확인한다.

버튼도 직접 눌러본다.

자세한 소개 부분으로 이동하는지 확인한다.

창을 조금 작게 줄였을 때 글자가 화면 밖으로 잘리거나 버튼이 사라지지 않는지도 본다.

문제가 있다면 보이는 그대로 말한다.

"제목과 소개 문장은 잘 보여.

그런데 아이디어 살펴보기 버튼을 눌러도 아래 소개
부분으로 이동하지 않아.

다른 내용을 늘리지 말고 이 버튼부터 고쳐줘."

다른 문제가 생겼다면 문장의 대상을 바꾼다.

"창을 작게 줄이면 소개 문장의 오른쪽이 화면 밖으로 나가.
작은 화면에서도 글이 전부 보이게 고쳐줘.
지금 잘되는 버튼은 그대로 유지해줘."

말로 설명하기 어렵다면 문제가 보이는 화면을 캡처한다.
Windows에서는 Windows 키+Shift+S를 함께 누른 뒤 문제가
있는 부분을 드래그한다.
대화창으로 돌아와 Ctrl+V를 누르고 한 문장을 덧붙인다.

"이 화면에서 내가 말한 문제가 보여.
원인을 찾아 고친 뒤 같은 화면에서 다시 확인해줘."

수정이 끝나면 문제가 생겼던 행동을 같은 순서로 다시 해본다.

처음 나온 화면을 내 취향으로 바꾼다

제목과 버튼이 제대로 움직이면 이제 화면을 내 것으로 바꿔본다.
디자인 용어를 몰라도 괜찮다.
어떤 사람이 볼 페이지인지, 어떤 느낌을 주고
싶은지를 말하면 된다.

"이 홈페이지는 [어떤 사람]이 볼 거야.
전체적으로 [원하는 느낌]이었으면 좋겠어.
제목과 가장 중요한 버튼이 먼저 눈에 들어오게 바꿔줘.
지금 잘되는 버튼 이동 기능은 유지해줘."

여행 사진을 소개한다면 조용하고 따뜻한 느낌을 원할 수 있다.
어린이를 위한 게임이라면 밝고 재미있는 느낌이 어울릴 수 있다.
가게를 소개한다면 대표 메뉴와 위치가 쉽게 보이는
것이 중요할 수 있다.

색 이름을 정확히 몰라도 된다.

"지금은 너무 딱딱한 회사 홈페이지처럼 보여.
친구에게 내가 좋아하는 사진을 보여주는 공간처럼
편안하고 따뜻하게 바꿔줘."

"색이 너무 많아서 어디를 봐야 할지 모르겠어.
배경과 글자, 버튼 색을 단순하게 정리해줘."

한 번에 모든 것을 바꾸기보다 서로 관련된 부분부터 바꾼다.
먼저 글과 색을 고치고 확인한 뒤, 필요하면 카드나
작은 움직임을 더한다.

바꿀 때마다 제목, 소개, 버튼이 그대로 있으며 버튼도 여전히
작동하는지 확인한다.

오늘 뭐 먹지?

냉장고에서 시작하는 저녁

오늘 뭐 먹지?

냉장고 속 재료로 저녁 메뉴를 찾는
아이디어를 소개합니다.

아이디어 살펴보기 ↓

남은 재료부터 살펴보면 어떨까요?

달걀, 대파, 밥처럼 집에 있는 재료를 적으면
만들 수 있는 메뉴를 알려주는 도구를
만들고 싶습니다.

지금은 아이디어를 소개하는 페이지입니다.

본문의 예시를 바탕으로 제작해 브라우저에서 캡처한 화면이다. 제목과 버튼을 유지하면서 글자 크기와 색상, 여백을 다듬고 아이디어 설명을 더했다. 아직 추천 기능이 없는 소개 페이지라는 점도 함께 표시했다.

기능을 더 넣기 전에 첫 페이지를 끝까지 본다

화면이 만들어지면 메뉴와 회원가입, 게시판 같은 기능을 더 넣고 싶어질 수 있다.

그 전에 처음 정한 한 장이 목적을 전달하는지 확인한다.

처음 보는 사람이 화면을 보고 무엇을 소개하는
페이지인지 알 수 있는가.

버튼을 눌렀을 때 다음 내용이 나오는가.

글을 읽은 뒤 무엇을 보여주려는 아이디어인지
한 문장으로 말할 수 있는가.

이 세 가지가 된다면 첫 결과물은 역할을 다한 것이다.

지금만 페이지가 내 컴퓨터에서만 열려도 괜찮다.

인터넷 주소를 만들고 다른 사람에게 링크를 보내는
일은 페이지의 내용과 버튼이 제대로 작동하는 것을
확인한 뒤 진행한다.

잘되는 상태를 먼저 남긴다

원하는 화면이 나오고 버튼이 제대로 움직인다면
그 상태를 보존한다.

"지금 홈페이지의 제목, 소개 문장, 버튼이 내가 원한 대로 보이고
버튼도 제대로 작동해.
나중에 문제가 생기면 지금 상태와 비교하거나
되돌릴 수 있게 안전하게 남겨줘.
무엇을 어떻게 남겼는지도 쉬운 말로 알려줘."

AI가 파일을 복사하거나 이전 상태를 남겼다고 하면 실제로
어디에 저장했는지 확인해달라고 한다.

"보존한 파일이 실제로 있는지 확인하고 위치를 알려줘.
아직 확인하지 않았다면 확인한 뒤 말해줘."

이렇게 잘되는 순간을 남겨두면 다음 수정이 마음에 들지 않거나 기능이 망가졌을 때 돌아갈 기준이 생긴다.

세컨드브레인에 내 홈페이지의 상태를 남긴다

파일을 보존하면 결과물은 남는다.

하지만 왜 이 제목을 골랐는지, 어디까지 직접 확인했는지, 다음에 무엇을 하려 했는지는 파일만 보고 알기 어려울 수 있다.

이때 세컨드브레인을 사용한다.

말 그대로 나를 도와주는 두 번째 기억이라는 뜻이다.

이 책에서는 AI와 함께 읽는 작업 기록 묶음을 이렇게 부른다.

AI와 오래 대화해도 새 대화를 열면 앞서 정한

내용을 놓칠 수 있다.

그래서 대화에만 남겨둔 결정을 파일에도 적어두고, 다음 작업을 시작할 때 그 파일을 먼저 읽게 한다.

백업이 결과물을 지켜준다면, 세컨드브레인은 그 결과를 만든 이유와 다음 순서를 이어준다.

처음 틀을 잡을 때는 대화의 원문, 작업 기록, 확정된 결정, 현재 상태, 다음에 읽을 문서의 위치를 나누어 둔다.

그래야 잠깐 떠올린 생각을 확정된 결정으로 착각하거나, 아직 확인하지 않은 일을 완료로 기억하는 일을 줄일 수 있다.

폴더와 문서 이름을 직접 설계할 필요는 없다.

제공 파일에 그 순서가 담겨 있다.

세컨드브레인에는 예제의 내용을 복사하는 것이 아니라 독자가 실제로 만든 홈페이지의 상태를 남긴다.

- 어떤 홈페이지를 만들고 있는가.
- 어떤 제목과 소개 문장을 사용하기로 했는가.
- 직접 확인한 것은 무엇인가.
- 아직 고치지 못한 것은 무엇인가.
- 다음에 할 일은 무엇인가.

다음 날의 AI에게 남기는 기억

모든 대화를 다시 설명하지 않도록

01 원문

처음 받은 자료와 요청

02 기록

오늘 한 일과 직접 확인한 결과

03 결정·대단원

무엇을 정했고, 왜 그렇게 정했는지

04 프로젝트 현재상태

완료한 것 · 남은 문제 · 다음 행동

05 전체 프로젝트 목록

어느 프로젝트의 기록을 읽을지

시작할 때 읽고, 작업을 마칠 때 갱신한다.

결과 파일만 남기는 것이 아니라 원문과 기록, 결정, 현재 상태, 다음에 읽을 위치를 함께 남긴다.

이 책과 함께 제공되는 세컨드브레인구축.md를 사용하면 필요한 폴더와 문서를 AI가 만들어준다.

구축 파일은 메이킹소울스튜디오 전자책 페이지에서도 무료로 내려받을 수 있다.

책 뒤의 '실습 자료 다운로드 안내'에서도 같은 주소를 확인할 수 있다.

Windows 파일 탐색기에서 '다운로드' 폴더를 연다.

세컨드브레인구축.md를 한 번 누르고 Ctrl+C를 누른다.

앞서 만든 '첫작품' 폴더를 열어 Ctrl+V로 복사한다.

.md는 글을 담은 파일의 형식이다.

내용을 직접 고치거나 코드를 입력할 필요는 없다.

코덱스나 클로드 코드에서 '첫작품' 폴더를 작업 위치로 선택한 뒤 다음 문장을 보낸다.

"이 폴더의 '세컨드브레인구축.md'를 먼저 읽어줘.
그 파일의 지침에 따라 내 세컨드브레인을 초기 구축해줘.
기존 홈페이지 파일은 보존하고, 내 선택이 필요한
부분만 쉽게 물어봐줘."

AI가 파일을 찾지 못하면 실제 파일이 '첫작품' 폴더에 있는지 확인하고, AI가 선택한 작업 폴더도 다시 확인한다.

이 파일의 내용을 대화창에 전부 옮겨 적을 필요는 없다.

구축이 끝나면 이렇게 부탁한다.

"지금 만든 홈페이지의 목적과 확정된 제목, 직접 확인한 기능,
다음에 할 일을 세컨드브레인에 기록해줘.
확인하지 않은 내용은 완료라고 쓰지 말아줘."

다음 날에는 같은 작업 폴더에서 새 대화를 열고 이렇게 말한다.

"세컨드브레인의 현재 상태와 다음 할 일을 먼저 읽고, 어제 만들던 홈페이지 작업을 이어가자."

AI가 기록을 실제로 읽었는지, 어제 정한 내용과 맞는지 확인한다.

파일을 만들어둔 것만으로 모든 대화에서 자동으로 기억하는 것은 아니다.

처음에는 읽어달라는 한 문장을 함께 보내는 편이 확실하다.

다른 사람에게 설명 없이 보여준다

이제 홈페이지를 다른 사람 한 명에게 보여준다.

아직 인터넷에 올리지 않았다면 내 컴퓨터 화면을 직접 보여줘도 된다.

멀리 있는 사람이라면 화면을 공유하며 보게 할 수도 있다.

먼저 내용을 길게 설명하지 않는다.

"내가 만든 페이지인데 처음 보는 사람도 이해할 수 있는지 한번 봐줘.
무엇을 소개하는 페이지인지 말해주고, 눌러보고 싶은 곳도 직접 눌러봐."

상대가 제목부터 읽는지, 버튼을 발견하는지, 버튼을 누른 뒤 어디를 보는지 지켜본다.

사용이 끝난 뒤에는 "예뻐?"라고만 묻지 않는다.

"처음 봤을 때 무엇을 소개하는 페이지라고 생각했어?
어떤 부분을 먼저 놀러보고 싶었어?
이해되지 않는 문장이 있었어?
이 아이디어를 더 알고 싶다는 생각이 들었어?"

내가 소개하려던 내용과 상대가 이해한 내용이 다르다면 글이나 화면의 순서를 고칠 단서가 생긴다.
버튼을 찾지 못했다면 크기나 위치, 버튼에 적합한 말을 바꿔볼 수 있다.

한 번에 한 문제만 고친다

다른 사람에게 보여주면 글, 색, 그림, 새로운 기능에 관한 의견이 한꺼번에 나올 수 있다.

그중에서 먼저 고칠 것은 상대가 페이지의 목적을 이해하지 못한 부분과 실제로 사용하다가 멈춘 부분이다.
취향에 가까운 의견과 새로운 기능 제안은 따로 남겨두고, 지금의 한 장을 이해하는 데 필요한지부터 판단한다.

AI에게는 다음처럼 말한다.

"처음 본 사람이 이 홈페이지를 [상대가 이해한 내용]을 소개하는 페이지라고 생각했어.
나는 [실제로 소개하려던 내용]이 먼저 전달되기를 원해.
새로운 기능은 추가하지 말고 제목과 소개 문장을 더 쉽게 고쳐줘.
지금 잘되는 버튼 기능은 유지해줘."

버튼에서 멈췄다면 요청도 달라진다.

"처음 본 사람이 버튼을 찾지 못했어.
버튼에 적힌 말과 위치를 더 알아보기 쉽게 바꿔줘.
다른 부분은 그대로 유지해줘."

수정이 끝나면 같은 화면을 다시 보여준다.
이번에는 별도의 설명 없이도 무엇을 소개하는지
알아보는지 확인한다.

확인한 반응과 수정 결과는 세컨드브레인에 남긴다.

"방금 다른 사람에게 보여주며 직접 확인한 반응과 고친 내용을
세컨드브레인에 기록해줘.
한 사람의 취향과 실제로 관찰한 문제를 구분하고, 다음에
확인할 한 가지도 남겨줘."

여기까지 왔다면 머릿속에 있던 생각이 다른 사람이 실제로
볼 수 있는 한 장의 화면이 된 것이다.

에필로그 - 책을 보지 않고 보내는 첫 문장

처음 이 책을 펼쳤을 때 AI는 멀리서 빠르게 발전하는 기술처럼 보였을지 모른다.

글을 쓰고 그림을 그리고 영상을 만드는 장면은 신기했지만, 그것이 내 생활과 무슨 관계가 있는지는 선명하지 않았을 수 있다.

직접 무언가를 만들어보라는 말도 이미 해본 사람들에게나 가능한 이야기처럼 들렸을 것이다.

지금 컴퓨터 화면에는 작은 홈페이지 한 장이 열려 있다.

제목과 소개 문장, 버튼 하나가 전부일 수도 있다.

아직 인터넷 주소도 없고, 많은 사람이 찾아오는 페이지도 아니다.

당장 돈을 벌어주는 제품도 아닐 것이다.

그래도 처음 머릿속에만 있던 생각과는 다르다.

화면을 열어볼 수 있고, 마음에 들지 않는 문장을 바꿀 수 있다.

버튼을 눌러보고, 움직이지 않으면 무엇이 잘못됐는지 AI에게 설명할 수 있다.

수정하기 전의 상태를 남기고, 다음 대화에서 이어갈 내용도 기록할 수 있다.

처음에는 AI에게 무엇을 말해야 하는지도 막막했다.

하지만 실제로 해보니 처음부터 완벽한 명령을 만들 필요는 없었다.

원하는 것을 평소 말하듯 설명하고, 나온 결과를 본 뒤 다시 말하면 됐다.

모르는 말이 나오면 쉽게 설명해달라고 했고, AI가 대신할 수 있는 일은 대신해달라고 부탁했다.

이제 다음에 무엇을 할지는 책이 정하지 않는다.

홈페이지를 인터넷에 공개하고 싶은 사람도 있을 것이다.

자신만의 주소를 연결하거나 사진을 넣고 싶은 사람도 있다.

누군가는 홈페이지를 여기에서 멈추고, 처음부터 만들고 싶었던 도구나 게임으로 돌아갈 수도 있다.

어느 쪽이든 다음 과정은 같다.

하고 싶은 일을 말하고, AI의 설명을 듣고, 직접 확인한 뒤 다음 말을 이어가면 된다.

이 책이 도메인을 연결하는 버튼의 위치까지 안내하지 않는 이유도 여기에 있다.

사용하는 서비스에 따라 화면과 방법은 달라질 수 있다.

무엇보다 여기까지 온 독자에게는 이제 자신의 상황을 AI에게 설명하고 필요한 도움을 받아낼 수 있는 경험이 생겼다.

홈페이지를 공개하고 싶다면 이렇게 부탁할 수 있다.

"지금 만든 홈페이지를 다른 사람도 인터넷에서 볼 수 있게 공개하고 싶어.
나는 처음이라 아무것도 몰라.
가능한 방법을 쉽게 비교해주고, 비용이 들거나 내가 결정해야 하는 일이 생기면 먼저 알려줘.
내가 고르면 실제 공개와 확인까지 한 단계씩 도와줘."

문장이 길다고 느껴질 수 있다.

무엇을 빠뜨렸는지 걱정되어 다시 책을 펼칠 수도 있다.
하지만 이제는 예제와 똑같이 말할 필요가 없다.
부족한 내용이 있다면 AI가 파일을 살펴보고 다시 질문할 수 있다.

설명이 어려우면 쉽게 말해달라고 하면 된다.

선택지가 너무 많으면 초보자에게 가장 쉬운 방법 하나를 추천해달라고 하면 된다.

진행하다 막히면 내가 무엇을 눌렀고 화면에 무엇이 나타났는지 말하면 된다.

다음 작업을 시작하는 데 필요한 것은 긴 명령문이 아니다.

대화창을 열고 이렇게 물어보자.

"홈페이지 공개하고 싶은데 어떻게 해?"

실습 자료 다운로드 안내

이 책의 실습 자료는 메이킹소울스튜디오 홈페이지의 전자책 페이지에서 별도로 제공한다.

전자책 PDF와 세컨드브레인구축.md를 함께 무료로 내려받을 수 있다.

구축 파일은 AI에게 전달하는 별도의 실습 자료다.

이 파일을 작업 폴더에 넣고 AI에게 읽어달라고 하면 나에게 맞는 세컨드브레인의 첫 구조를 만들 수 있다.

파일을 찾고 전달하는 순서는 5부에서 안내했다.

구축 파일은 아래 회사 홈페이지의 전자책 페이지에서도 무료로 내려받을 수 있다.

주소를 누르면 자료 페이지가 열린다.

www.makingsoulstudio.com/ebook/

자료와 확인 기준

외부 사례는 공개 인터뷰와 공식 안내를 기준으로 정리했다.

인터뷰에 나오는 매출은 창업자의 설명이며 순이익이나 독자가 얻을 수익을 뜻하지 않는다.

저자의 제작 경험과 외부 사례는 구분해서 읽어주기 바란다.

아래 웹 자료는 2026년 9월 8일 확인했다.

- Waymo 공식 자율주행 서비스 안내: 미국 일부 지역의 무인 승차 서비스.
- micro1 일상 동작 데이터 모집 안내: 일상 동작을 촬영해 학습 자료로 제공하는 일거리.
- 위노트 창업자 인터뷰: 교사 경험, 초기 시제품, 사용자 의견과 제품 개편.
인터뷰 속 인디펍 매출을 위노트 매출로 옮겨 쓰지 않았다.
- 위노트 공식 요금 안내: 개인과 기관의 구매 방식.
매출 규모의 근거는 아니다.
- Launch Fast 창업자 인터뷰: 48시간 초기 제작, Legacy X와 협력,
인터뷰 시점의 월 반복 매출.
영상 제목의 수치와 구분해 본문 발언을 따랐다.
- OpenAI 데스크톱 앱 안내: 로그인, 작업 위치와 Codex 선택.
- OpenAI Windows 앱 안내: Windows 공식 설치 경로.
- Claude Code 데스크톱 안내: Code 작업과 로컬 폴더 선택.

앱의 화면과 이용 조건은 바뀔 수 있다.

실습에서는 버튼 이름을 외우기보다, 내가 작업할 폴더를 선택하고 원하는 결과를 설명한 뒤 직접 확인하는 순서에 집중하면 된다.