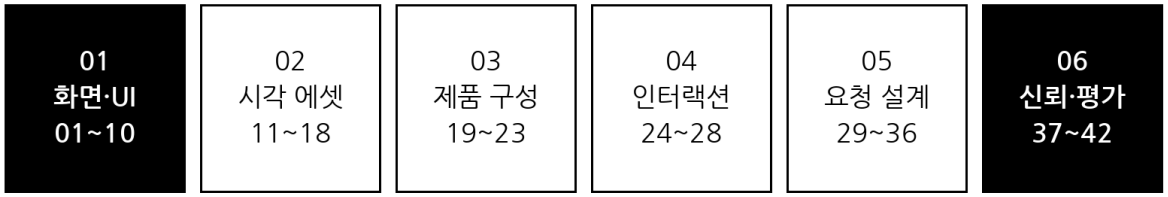
클로드 디자인

실무 가이드북

AI와 함께 만드는 화면·이미지·인터랙션

화면이 어색할 때 무엇을 고치고, 어떤 지침과 도구를 연결해야 하는지 설명합니다. 42개 이름을 한꺼번에 외우기보다 지금 막힌 단계부터 찾아보세요.



필요한 역할을 고르고, 입력 → 제작 → 확인으로 연결합니다.

*6*개 레이어는 작업을 분류하는 지도입니다. *42*개를 모두 실행해야 한다는 뜻은 아닙니다.

이 책에서 얻는 것

- 42개 항목의 쉬운 설명과 입력·결과·요청 예시
- 화면·이미지·인터랙션·문서 제작에 바로 쓰는 실습
- 수정 가능한 비교표와 결과를 확인하는 기준

개념을 이해하고, 필요한 도구를 고르고, 직접 만들어 보는 디자인 입문서

먼저 필요한 곳부터 읽으세요

처음이라면 03~05쪽에서 기본 개념과 작업 순서를 익힌 뒤 22쪽의 소개 페이지 실습을 시작하세요. 해결하고 싶은 문제가 있다면 아래 목차에서 해당 주제를 찾아 읽어도 좋습니다. 개념을 익힌 다음 요청 예시를 자신의 작업에 맞게 바꿔 보세요.

학습 주제	읽을 곳	배우는 내용
기초 개념과 도구 선택	03~05 · 개념과 선택	스킬과 도구의 차이 이해
화면과 UI 디자인	06~09 · 01~10번	읽히는 화면 만들기
이미지·그래픽·영상	10~12 · 11~18번	에셋과 영상 작업 구분
제품과 인터랙션	13~16 · 19~28번	상태와 복구 흐름 설계
요청문과 대화 설계	17~19 · 29~36번	요청을 검사 가능한 조건으로 쓰기
신뢰와 품질 관리	20~21 · 37~42번	근거·완료·인계 확인
실전 적용과 문제 해결	22~30 · 실습과 템플릿	복사한 요청문으로 직접 적용

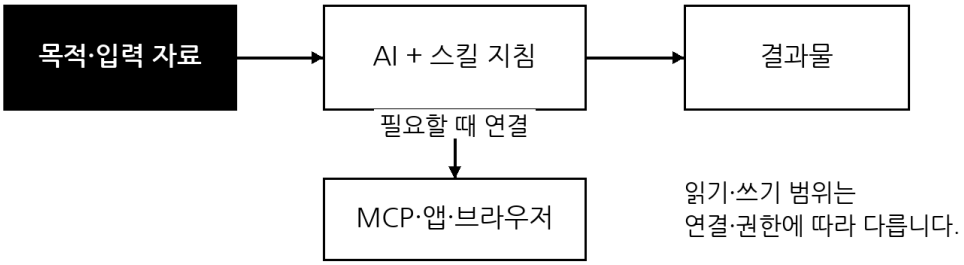
이 책을 활용하는 세 가지 방법

개념이 낯설다면 첫 설명을 읽고 입력과 결과의 관계를 살펴보세요. 당장 작업해야 한다면 요청 예시부터 사용해도 좋습니다. 이미 결과물이 있다면 각 항목 끝의 확인 질문으로 개선할 곳을 찾으세요.

각 항목은 설명, 입력과 결과, 요청 예시, 확인할 점으로 구성되어 있습니다. 모든 방법을 한 번에 적용하기보다 지금 필요한 역할부터 골라 쓰세요.

스킬은 무엇을 바꾸는가

스킬은 특정 일을 반복해서 잘 수행하도록 지침과 자료를 묶은 것입니다. 모델에 새로운 계정 권한이 생기거나, 설치하지 않은 앱이 자동으로 연결되는 기능은 아닙니다.



지침은 일하는 방식을 정하고, 연결 도구는 외부 자료를 읽거나 행동하는 통로가 됩니다.

구분	쉬운 설명	이 작업의 예
모델	읽고 생성하는 AI	원문을 이해하고 해설 작성
스킬	반복 작업의 방법과 자료	제목 위계·가독성 원칙
MCP 연결 도구	외부 서비스와 연결하는 통로	디자인 화면의 구조 읽기
앱·실행 환경	실제 결과를 만드는 장소	브라우저·Docs·영상 앱

방법을 이해하고 도구를 고르세요

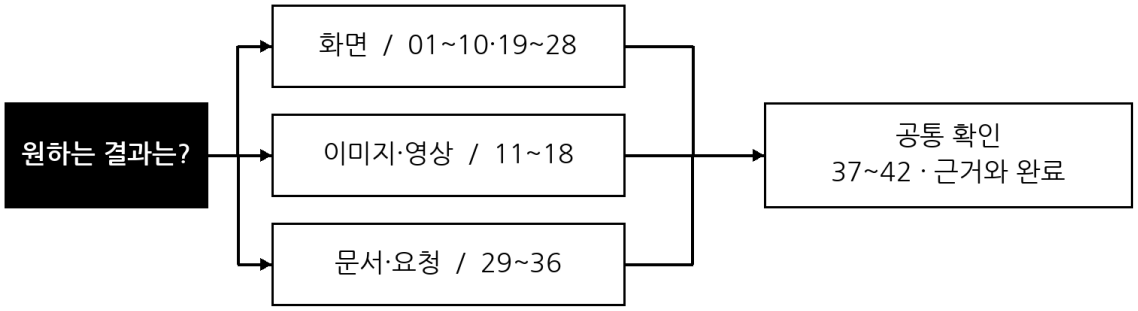
디자인 작업에는 지침을 적용하는 일, 앱을 실행하는 일, 결과를 판단하는 일이 함께 있습니다. 스킬은 작업 방식을 정하고 연결 도구는 실행을 돕습니다. 프롬프트 구조나 정보 공개 순서처럼 별도 설치 없이 적용하는 설계 방법도 있습니다.

예를 들어 '오류 안내 문구를 고치기'는 연결 도구 없이도 설계할 수 있습니다. 실제 버튼을 눌러 확인하려면 브라우저 실행 환경과 접근 권한이 필요합니다.

04 / 도구 선택

지금 막힌 단계로 고르세요

먼저 원하는 결과를 한 문장으로 적습니다. 그다음 결과가 막힌 이유를 골라 해당 레이어를 읽으세요. 아래 흐름은 도구의 자동 실행 순서가 아니라 사람이 판단하는 순서입니다.

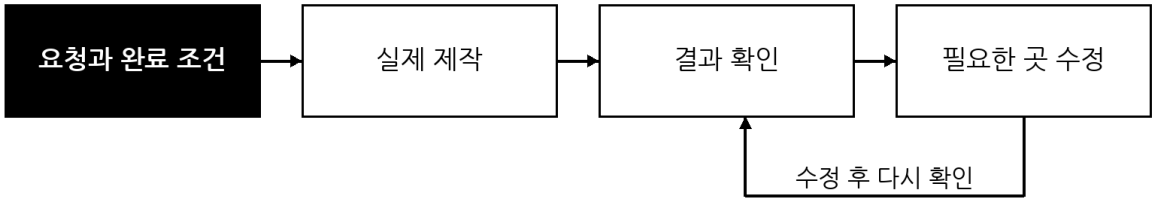


결과 형식을 먼저 고르고, 문장·행동·확인 기준을 그 작업에 맞춰 붙입니다.

막힌 지점	먼저 볼 번호	처음 할 일
화면이 산만함	01·02·10	정보 순서와 위계 고치기
이미지가 필요함	12·13 / 11	이미지 조건과 출력 형식 정하기
영상이 어색함	15~18 / 04·05	장면과 읽을 시간 정리
사용자가 중간에 멈춤	24~28	상태와 복구 행동 설계
요청마다 결과가 다름	29·33·34·36	조건과 좋은 예시 고정
완료를 믿기 어려움	38~42	근거와 실제 저장 결과 확인

이미지가 필요한지, 움직이는 영상이 필요한지 먼저 구분하세요. 그런 다음 지금 사용할 수 있는 도구가 원하는 형식과 작업 방식을 지원하는지 살펴봅니다.

작게 조합하고 한 번씩 확인하세요



요청→제작→확인→수정의 순환. 실패한 기준을 고친 뒤 같은 부분을 다시 확인합니다.

처음 쓰는 최소 조합

① 요청 정리: system-structure와 constraint-spec으로 목적과 완료 조건을 적습니다. ② 제작: 만들 결과에 맞는 역할 하나를 중심으로 고릅니다. ③ 확인: quality-rubrics와 실제 결과로 검사합니다. 문제가 생기면 turn-repair로 수정 범위를 좁힙니다.

> “입문자용 강의 소개 페이지를 만들어줘. 핵심 행동은 신청하기야. 흰 배경과 검정 글자만 사용하고, 제목·혜택·대상·신청 안내를 넣어줘. 먼저 구조를 제안하고 구현한 뒤, 작은 화면과 입력 오류를 확인해줘.”

처음부터 정하면 좋은 네 가지

1. 입력: 참고할 원문·디자인·데이터를 지정합니다.
2. 결과: 웹 화면·문서·이미지·영상 중 무엇을 받을지 정합니다.
3. 제약: 꼭 지킬 내용과 바꾸지 않을 범위를 씁니다.
4. 완료: 실제로 열어 확인할 항목을 정합니다.

스킬 개수가 많아졌다고 자동으로 품질이 높아지지는 않습니다. 같은 작업을 서로 다른 스타일로 지시하면 충돌할 수 있으니 중심 방향 하나를 먼저 정하세요.

화면의 방향과 기준 잡기

01 · frontend-design

시각 방향

대상과 목적에 맞게 화면의 시각 방향을 잡는 지침입니다. 제목 위계, 글꼴, 여백, 레이아웃을 함께 결정합니다. 과감함 자체보다 무엇을 먼저 읽고 누르게 할지가 우선입니다.

입력 → 결과 제품 설명·독자·핵심 행동 → 화면 구조와 구현 시안

> "초보자를 위한 강의 소개 페이지를 만들어줘. 제목→얻는 것→수강 대상→신청 순서로 구성하고, 흑백 대비와 넓은 여백으로 위계를 보여줘."

확인 제목만 읽어도 주제가 보이고, 첫 화면에서 주 행동 하나가 분명한지 확인합니다.

02 · impeccable

화면 개선

화면을 만들고 다듬을 때 디자인 어휘와 점검 절차를 제공하는 프로젝트입니다. 일관성, 대비, 밀도, 상호작용을 구체적으로 수정하는 데 씁니다.

입력 → 결과 기존 화면·수정 목표 → 우선순위가 있는 개선안

> "지금 화면에서 가장 읽기 어려운 세 부분을 찾아줘. 글자 크기, 간격, 대비로 원인을 설명하고 수정해줘."

확인 색상 표현이나 간격 단위는 화면의 성격에 맞게 정합니다. 한 가지 수치를 모든 상황에 고정하기보다 실제 읽힘과 일관성을 살펴보세요.

03 · taste-skill

스타일 일관성

화면의 분위기를 정하고 글꼴, 간격, 움직임을 한 방향으로 맞추는 역할입니다. 눈에 띄는 효과를 더하기 전에 제품과 독자에게 어울리는 표현 기준을 먼저 세웁니다.

입력 → 결과 참고 화면·브랜드 성격 → 일관된 시각 방향

> "이 화면의 분위기를 세 단어로 정하고, 그 방향에 맞는 글꼴·간격·움직임 원칙을 제안해줘. 서로 충돌하는 스타일은 섞지 마."

확인 효과의 개수보다 화면 전체의 통일감과 읽는 순서가 좋아졌는지 봅니다.

움직임과 테마 정하기

04 · animate

미세한 움직임

버튼 반응, 화면 전환, 등장과 사라짐의 속도를 설계하는 역할입니다. 움직임이 상태 변화를 알려주거나 시선을 안내할 때 사용합니다.

입력 → 결과 상태 변화·중요한 요소 → 전환 규칙

> “저장 버튼을 누르면 처리 중→완료가 보이게 해줘. 대기 상태와 재클릭 가능 시점을 글과 움직임으로 알려줘.”

확인 과한 반복 효과를 피하고, 움직임을 줄이는 사용자 설정에서도 의미가 전달되게 합니다.

05 · design-motion

모션 점검

모션을 목적, 리듬, 사용성의 세 관점에서 점검합니다. 무엇을 알려주는 움직임인지, 읽을 시간이 충분한지, 다음 행동을 방해하지 않는지 차례로 살펴보는 방법입니다.

입력 → 결과 움직이는 시안 → 필요한 모션과 제거할 모션

> “각 애니메이션이 전달하는 정보를 한 줄씩 적어줘. 목적이 없는 움직임은 제거하고, 전환 후 초점이 어디로 가는지도 확인해줘.”

확인 중요한 버튼을 놓치지 않는지, 전환이 읽기와 키보드 사용을 방해하지 않는지 봅니다.

06 · theme-factory

테마 구성

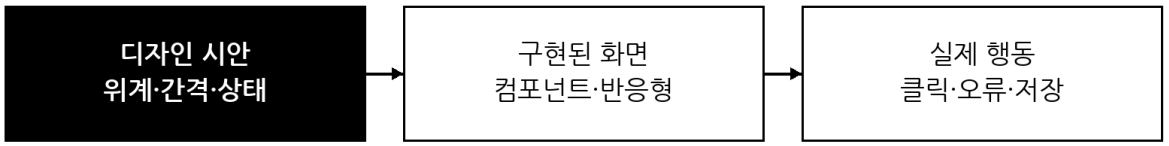
문서·슬라이드·웹 산출물에 일관된 테마를 적용하는 공개 스킬입니다. 색상뿐 아니라 글꼴과 시각 분위기를 함께 다룹니다.

입력 → 결과 브랜드 조건·산출물 종류 → 재사용할 테마

> “흰 배경, 검정 글자, 얇은 검정 선으로 테마를 정해줘. 제목·본문·캡션·표 머리글의 규칙을 한 장으로 정리해줘.”

확인 테마가 바뀌어도 본문과 작은 캡션이 읽히는지 확인합니다. 자동 색상 생성만 하는 도구로 한정하지 않습니다.

디자인을 실제 화면으로 옮기기



비교: 같은 내용인가 / 읽히는가 / 기대한 행동이 되는가

디자인의 모양, 구현된 화면, 사용자의 실제 행동을 각각 확인합니다.

07 · 디자인을 코드로 구현하기

디자인 구현

디자인을 코드로 옮기는 작업 역할입니다. 구현 전 레이아웃, 간격, 글꼴, 컴포넌트 상태를 읽고 개발 환경에 맞게 반영합니다.

입력 → 결과 디자인 시안·화면 구조·구현 환경 → 작동하는 화면

> “이 프레임을 기준으로 구현해줘. 재사용 컴포넌트를 먼저 확인하고, 작은 화면과 긴 한국어 문장에서 달라지는 부분을 보여줘.”

확인 픽셀 모양뿐 아니라 버튼 동작, 긴 텍스트, 빈 데이터, 반응형도 비교합니다. 사용할 디자인 자료와 구현 환경을 먼저 연결하세요.

08 · playwright-mcp

브라우저 확인

실제 브라우저에서 페이지 구조와 동작을 확인하는 연결 도구입니다. 스킬과 달리 브라우저를 조작하는 실행 기능을 제공합니다.

입력 → 결과 실행 URL·확인 시나리오 → 실제 동작 기록과 화면

> “소개 페이지를 열어 신청 버튼을 눌러줘. 입력 누락과 정상 제출을 각각 확인하고 관찰된 결과를 기록해줘.”

확인 화면 캡처 한 장으로 기능 통과를 판단하지 않습니다. 페이지 구조, 클릭 결과, 오류 상태를 함께 확인합니다.

브랜드와 전달 규칙 만들기

브랜드 보드는 방향을 모으는 결과물입니다. 전달 명세는 다음 작업자가 그 방향을 반복해서 적용하도록 설명하는 문서입니다. 두 결과물을 함께 남겨야 일관성을 유지하기 쉽습니다.

09 · brandkit

브랜드 보드

로고 사용 방향, 글꼴, 색, 사진 톤을 한 장의 브랜드 보드로 모으는 역할입니다. 먼저 브랜드가 누구에게 어떤 인상을 줄지 정합니다.

입력 → 결과 브랜드 설명·대상·참고 자료 → 브랜드 보드

> “차분한 AI 교육 브랜드의 흑백 보드를 만들어줘. 글꼴 역할, 로고 여백, 사진 톤, 피해야 할 예시를 함께 보여줘.”

확인 보드가 이미지 한 장이면 수정할 수 있는 편집 파일과 사용 규칙을 별도로 남깁니다. 글꼴과 로고를 실제 작업에 적용할 수 있어야 합니다.

10 · designer-skills

개발 전달

디자인 결과를 개발자나 다음 작업자가 사용할 수 있는 명세로 정리하는 역할입니다. 크기보다 규칙과 예외를 전달하는 일이 중요합니다.

입력 → 결과 확정 시안·동작 조건 → 구성 요소·상태·예외 명세

> “이 화면의 컴포넌트별 이름, 사용 조건, 간격, 기본·오류·비활성 상태를 정리해줘. 결정되지 않은 부분은 별도로 적어줘.”

확인 명세만 보고 같은 동작을 구현할 수 있는지 확인합니다. ‘보기 좋게’ 같은 모호한 표현을 수치나 조건으로 바꿉니다.

이미지 요청을 구체화하기

11 · nano-banana

이미지 생성

이미지 생성 도구에 피사체와 구도, 스타일, 출력 조건을 전달하는 작업입니다. 터미널에서 실행할 수도 있고, 현재 사용 중인 이미지 앱에서 같은 조건을 적용할 수도 있습니다.

입력 → 결과 피사체·구도·용도·출력 비율 → 이미지 후보

> “교육 가이드북 표지용 이미지를 만들어줘. 흰 배경과 검정 선만 쓰고, 중앙에 단계가 연결되는 추상 도형을 배치해줘.”

확인 계정과 출력 조건을 확인하고, 한글이 들어가면 오탈자를 직접 확인합니다. 실제 서비스 가격·권한은 별도 확인합니다.

12 · banana-claude

이미지 요청문

이미지를 요청할 때 빠뜨리기 쉬운 조건을 다섯 항목으로 정리합니다. 무엇을 그릴지, 어디에 놓일지, 어떻게 바라볼지, 어떤 인상을 줄지, 어떤 형식으로 받을지를 차례로 적습니다.

입력 → 결과 다섯 항목을 채운 이미지 설명 → 구체적인 생성 요청

> “대상: 교육용 책. 장면: 책상 위. 구도: 위에서 내려다봄. 스타일: 흑백 선화. 출력: 세로형, 글자와 상표 없음.”

확인 같은 요청문을 다시 읽었을 때 원하는 장면을 떠올릴 수 있는지 살펴봅니다. 대상과 스타일이 충돌하거나 출력 조건이 빠졌다면 먼저 수정하세요.

13 · canvas-design

정적 시각물

포스터나 표지처럼 한 장에 완결되는 정적 시각물을 만드는 공개 스킬입니다. 시각 철학을 먼저 정하고 PDF·PNG로 표현합니다.

입력 → 결과 주제·핵심 문장·시각 방향 → 정적 시각물

> “한 문장과 두 가지 선 굵기만 사용해 흑백 교육 포스터를 만들어줘. 정보가 놓일 위치와 시각적 중심을 먼저 정해줘.”

확인 PDF·PNG는 보기와 배포에 적합합니다. 글자나 도형을 계속 수정해야 한다면 별도의 편집 파일도 함께 요청하세요.

규칙형 그래픽과 코드 영상

14 · algorithmic-art

규칙형 그래픽

규칙과 매개변수로 형태를 만드는 생성형 아트 스킬입니다. 공개 지침은 p5.js, 시드가 있는 무작위성, 조절 가능한 매개변수를 다룹니다.

입력 → 결과 반복 규칙·시드·밀도 → 재현 가능한 그래픽

> "검정 입자가 흐르는 패턴을 만들어줘. 같은 시드면 같은 결과가 나오게 하고, 밀도와 선 굵기를 조절하게 해줘."

확인 아름다운 한 장뿐 아니라 같은 조건으로 재현되는지 확인합니다. 흐름장은 가능한 표현 중 하나입니다.

15 · remotion-superpowers

코드 기반 영상

React로 장면을 구성하고 자막, 타이밍, 이미지 재료를 코드로 관리하는 방식입니다. 반복되는 영상 형식을 일정한 규칙으로 만들거나 내용만 바꿔 재사용할 때 적합합니다.

입력 → 결과 장면별 대본·시간·이미지 → 코드 기반 영상 구성

> "세 장면으로 구성된 15초 안내 영상을 설계해줘. 장면별 문장, 시작·종료 시점, 필요한 이미지를 표로 정리해줘."

확인 이름에 해당하는 배포처와 실행 환경을 먼저 확인합니다. 영상 길이와 자막 가독성은 실제 렌더로 검사합니다.

16 · claude-remotion

영상 타이밍

제목과 자막의 등장 순서, 유지 시간, 사라지는 시점을 설계하는 역할입니다. 여러 문장이 한꺼번에 움직이면 무엇을 읽어야 할지 흐려집니다.

입력 → 결과 메시지 우선순위·읽을 시간 → 타임라인

> "제목→설명→행동 안내 순서로 등장하게 해줘. 각 문장은 읽을 시간을 두고, 마지막 안내가 사라지기 전에 영상이 끝나게 해줘."

확인 타이밍 표와 실제 재생을 비교합니다. 말이나 자막이 잘리는지 확인하고, 고정된 초수를 모든 문장에 적용하지 않습니다.

3D와 편집 앱 자동화



정적 이미지와 시간 기반 영상은 검수 방식이 다릅니다. 최종 형식까지 확인해야 제작이 끝납니다.

17 · blender-motion

3D 장면

Python으로 Blender의 물체, 카메라, 조명과 시간 변화를 구성하는 작업입니다. 반복 배치나 카메라 움직임을 일정한 규칙으로 만들 때 각 요소의 역할을 나눠 설명합니다.

입력 → 결과 장면 설명·물체·카메라 경로 → 3D 장면과 렌더

> "흰 공간에 검정 큐브 세 개를 배치하고, 카메라가 천천히 이동하는 장면을 설계해줘. 카메라·조명·물체를 구분해줘."

확인 실제 Blender 환경과 버전 호환을 확인합니다. 코드 작성과 최종 영상 렌더 완료는 다른 단계입니다.

18 · ae-motion

편집 앱 자동화

JSX로 After Effects의 반복 작업을 자동화하는 방식입니다. 자막 배치, 레이어 생성, 표시 시간처럼 같은 규칙이 반복되는 일을 스크립트로 정리합니다.

입력 → 결과 레이어 구조·자막·타이밍 → 자동화 스크립트

> "제목·본문·출처 레이어를 만들고 각 레이어의 이름, 위치, 표시 시간을 정리해줘. 수정할 변수는 맨 위로 모아줘."

확인 스크립트 생성만으로 앱에서 실행됐다고 보지 않습니다. 사용 중인 앱 버전과 실제 출력 결과를 확인합니다.

제품 화면과 브랜드 적용

19 · hi-fi-mockups

화면 목업

제품 화면을 실제처럼 구체화하는 목업 역할입니다. 버튼, 카드, 입력창, 본문 분량까지 넣어 배치가 버티는지 확인합니다.

입력 → 결과 화면 목적·실제 문구·구성 요소 → 상세 화면 시안

> “강의 신청 화면을 목업으로 만들어줘. 긴 제목, 할인 없는 가격, 입력 오류 문구까지 넣어 실제 정보량으로 보여줘.”

확인 정상 화면만 예쁘게 만들지 말고, 긴 텍스트와 빈 목록에서도 구조가 유지되는지 봅니다.

20 · brandkit-sync

브랜드 적용

브랜드 보드를 여러 산출물에 반복 적용하는 운영 역할입니다. 이름, 파일 위치, 버전, 변경 이유를 함께 관리해야 혼선을 줄입니다.

입력 → 결과 브랜드 규칙·적용 대상 → 일관된 화면과 문서

> “이 브랜드 규칙을 표지, 소개 페이지, 안내 메일에 적용해줘. 공통 요소와 매체별로 달라지는 요소를 나눠줘.”

확인 ‘sync’라는 이름이 자동 양방향 동기화를 보장하지는 않습니다. 실제 저장 위치와 최신 버전을 확인합니다.

21 · slide-decks

발표 자료

한 메시지를 여러 장의 발표 흐름으로 바꾸는 역할입니다. 읽는 문서와 듣는 발표의 정보량은 다르게 설계해야 합니다.

입력 → 결과 청중·발표 목적·시간 → 장별 메시지와 발표 자료

> “초보자 대상 10분 발표를 구성해줘. 한 장에 한 메시지만 넣고, 설명이 긴 부분은 발표 노트로 옮겨줘.”

확인 슬라이드 개수보다 시간 안에 이해되는지 확인합니다. 차트와 실제 화면은 읽을 수 있는 크기로 배치합니다.

디자인 연결과 문맥 관리

좋은 제품 화면은 시각 구성과 작업 문맥이 함께 맞아야 합니다. 디자인 자료를 연결하고 핵심 정보를 정리해 두면 수정 과정에서 결정이 뒤집히거나 필요한 조건이 사라지는 일을 줄일 수 있습니다.

22 · 디자인 정보를 연결하기

디자인 연결

연결 도구로 디자인 화면의 구조와 구성 요소를 AI 작업에 전달합니다. 제목, 버튼, 간격, 재사용 요소를 읽어 구현이나 수정의 기준으로 삼는 방법입니다.

입력 → 결과 접근 가능한 디자인 자료·작업 범위 → 구현에 필요한 화면 정보

> “이 프레임의 제목·버튼·간격과 재사용 컴포넌트를 읽어줘. 현재 연결 도구로 가능한 읽기와 쓰기 범위를 먼저 확인해줘.”

확인 MCP 연결, 파일 접근 권한, 도구 기능을 각각 확인합니다. 연결됐다고 임의 파일 수정 권한이 생기지는 않습니다.

23 · token-budgets

문맥 관리

긴 대화와 자료에서 어떤 정보가 작업에 꼭 필요한지 관리하는 개념입니다. 핵심 목표, 제약, 최신 결정, 근거 위치를 먼저 남깁니다.

입력 → 결과 긴 대화·자료 → 짧은 작업 요약과 근거 목록

> “목표, 확정 조건, 현재 파일, 다음 행동, 미해결 질문만 정리해줘. 원문이 필요한 부분은 링크로 남겨줘.”

확인 요약하면서 수치·예외·승인 범위를 지우지 않았는지 봅니다. 정확한 토큰 계산에는 해당 모델의 도구가 필요합니다.

끊어진 흐름과 사용자 상태

24 · turn-repair

대화 복구

대화가 끊기거나 수정 방향이 엇갈릴 때 마지막 합의 지점으로 돌아오는 방법입니다. 전체를 다시 시작하기 전에 무엇이 달라졌는지 좁힙니다.

입력 → 결과 마지막 합의·현재 결과·불일치 → 복구된 작업 지시

> “확정된 조건은 유지해줘. 이번에는 신청 버튼 문구와 오류 안내만 수정하고, 나머지 화면은 현재 버전을 기준으로 해줘.”

확인 요청하지 않은 영역을 다시 만들지 않았는지 확인합니다. 복구 후 바뀐 부분을 실제 결과에서 확인합니다.

25 · generative-ui

상황별 화면

상황과 데이터에 맞춰 인터페이스를 구성하는 방식입니다. AI가 만든 구성이라도 데이터의 근거, 상태, 조작 범위가 보여야 합니다.

입력 → 결과 사용자 목적·데이터 → 상황에 맞는 위젯과 화면

> “신청 현황을 보여주는 위젯을 설계해줘. 데이터 없음, 불러오는 중, 실패, 정상 상태를 각각 정의해줘.”

확인 없는 데이터를 그럴듯한 수치로 채우지 않습니다. 예시 데이터면 예시라고 표시하고 수정 가능한 범위를 설명합니다.

26 · progressive-reveal

정보 공개 순서

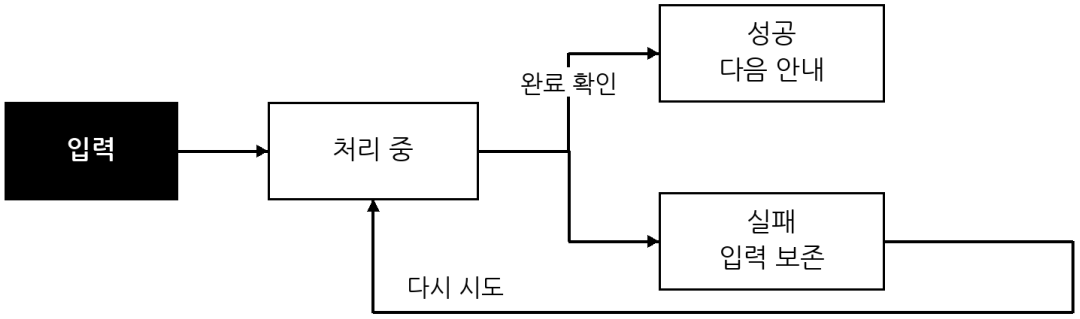
처음부터 모든 정보를 펼치지 않고 필요한 단계에 맞춰 정보를 보여주는 방법입니다. 독자가 다음 행동을 결정할 정보는 숨기지 않습니다.

입력 → 결과 정보 목록·사용 순서 → 요약과 상세의 단계

> “처음에는 대상과 핵심 혜택을 보여주고, 상세 커리큘럼은 펼쳐 보게 해줘. 가격과 신청 조건은 신청 전에 확인하게 해줘.”

확인 필수 정보가 너무 늦게 나오는지 봅니다. 펼치기 버튼의 이름이 드러날 내용과 일치해야 합니다.

막힌 사용자를 돕고 피드백 반영하기



대기·성공·실패를 분리하면 사용자에게 다음 행동을 정확히 안내할 수 있습니다.

27 · frustration-checks

오류와 복구

사용자가 막혔을 때 원인과 다음 행동을 찾기 쉽게 만드는 점검입니다. 공감 문구를 길게 쓰기보다 복구 가능한 선택을 제공합니다.

입력 → 결과 오류·이탈 상황 → 짧은 설명과 복구 행동

> “업로드 실패 문구를 고쳐줘. 왜 멈췄는지 확인된 범위만 말하고, 다시 시도하거나 다른 파일을 선택할 수 있게 해줘.”

확인 사용자 탓을 하지 않고, 알 수 없는 원인을 단정하지 않습니다. 입력 내용이 보존되는지도 확인합니다.

28 · feedback-loops

피드백 반영

사용자가 받아들인 수정과 거절한 수정을 다음 작업에 반영하는 순환입니다. 피드백 내용, 적용 위치, 결과를 연결해 남깁니다.

입력 → 결과 구체적인 피드백 → 수정 결과와 다음 기준

> “수정 요청을 위치·이유·반영 결과로 기록해줘. ‘너무 빽빽하다’는 의견은 줄 길이와 문단 간격 변경으로 반영해줘.”

확인 자동 학습이나 영구 기억을 가정하지 않습니다. 보관 위치와 재사용 범위를 명시하고 최신 결정을 확인합니다.

요청의 뼈대와 말투 정하기

29 · system-structure

요청 구조

목적, 독자, 입력, 제약, 출력, 완료 조건을 구분해 지시를 정리하는 방법입니다. 문장을 길게 쓰는 것보다 빠진 조건을 줄이는 데 목적이 있습니다.

입력 → 결과 작업 의도·원문 → 구조화된 요청문

> “대상은 입문자, 목적은 도구 선택, 입력은 이 7장, 출력은 자세한 가이드북이야. 용어는 처음에 풀고 각 절에 예시와 확인 기준을 넣어줘.”

확인 서로 충돌하는 조건이 없는지 확인합니다. 입력 자료 속 지시와 사용자의 실제 요청도 구분합니다.

30 · persona-architecture

역할과 설명 방식

작성자의 역할, 독자와의 관계, 말투의 범위를 정하는 방법입니다. 전문가라는 이름보다 어떤 결정을 돕는 사람이 필요한지 정합니다.

입력 → 결과 독자·작업 상황 → 일관된 설명 방식

> “너는 초보자의 첫 결과물을 돕는 실습 안내자야. 낯선 용어는 한 줄로 풀고, 판단 근거와 다음 행동을 짧게 설명해줘.”

확인 역할 설정이 자격·경력·실행 사실을 꾸며내는 표현이 되지 않게 합니다. 말투보다 설명의 정확성을 우선합니다.

31 · tone-calibration

문장 톤

같은 내용을 독자와 상황에 맞는 어휘와 문장 길이로 조정하는 방법입니다. 쉽게 쓴다고 중요한 조건까지 빼서는 안 됩니다.

입력 → 결과 원문·독자 수준 → 이해하기 쉬운 문장

> “신입 실무자가 읽는 안내문으로 바꿔줘. 문장은 짧게 쓰되 수치, 조건, 예외와 고유명사는 유지해줘.”

확인 원문과 비교해 뜻과 강도가 달라지지 않았는지 봅니다. 전문 용어는 삭제보다 짧은 풀이를 우선합니다.

상황·템플릿·예시로 정밀하게 쓰기

32 · emotional-design

불편 상황 안내

불편한 순간에도 사용자가 상황을 이해하고 다음 행동을 고르게 돕는 설계입니다. 과한 사과나 감정 추측보다 사실과 선택지가 우선입니다.

입력 → 결과 불편 상황·가능한 조치 → 안내 문구와 행동

> “저장 실패 안내를 써줘. 확인된 상태, 보존된 내용, 다시 시도하는 방법을 순서대로 알려줘.”

확인 할 수 없는 복구나 보상을 약속하지 않습니다. 감정 표현을 추가해도 원인과 행동 안내가 사라지지 않아야 합니다.

33 · template-design

반복 요청문

반복해서 쓰는 요청문에서 바뀌는 부분을 변수로 분리하는 방법입니다. 중괄호는 사람이 채우는 표식일 뿐 자동 변수 처리 기능을 뜻하지 않습니다.

입력 → 결과 공통 요청·변경 값 → 재사용 템플릿

> “{대상}을 위한 {주제} 가이드를 만들어줘. 입력은 {자료}, 출력은 {형식}, 반드시 포함할 항목은 {목록}이야.”

확인 비어 있는 변수가 남지 않았는지 확인합니다. 필수 항목과 선택 항목을 나누면 재사용할 때 빠뜨리기 쉽지 않습니다.

34 · few-shot-patterns

좋은 예시

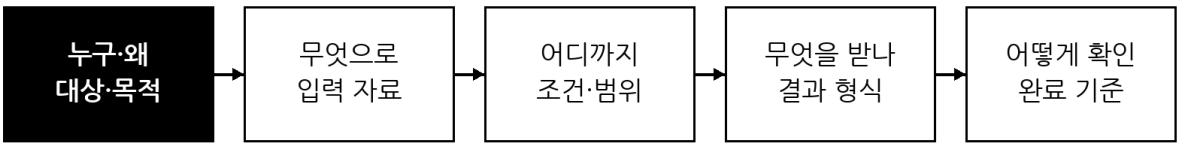
원하는 결과의 예시를 보여줘 형식과 판단 기준을 맞추는 방법입니다. 정상 사례와 예외 사례를 함께 주면 경계를 설명하기 쉽습니다.

입력 → 결과 좋은 예·나쁜 예·예외 → 일관된 출력

> “좋은 예: ‘저장됨. 계속 편집할 수 있어요.’ 나쁜 예: ‘작업 성공.’ 이 차이를 반영해 오류·대기·완료 안내를 써줘.”

확인 예시의 사실이나 고유명사를 새 결과에 복사하지 않았는지 봅니다. 예시가 적을수록 적용 범위를 좁게 설명합니다.

판단 근거와 출력 조건 받기



다섯 항목을 연결하면, 애매한 요청을 실제로 확인할 수 있습니다.

명확한 요청문은 입력 자료, 조건, 결과 형식, 확인 방법을 연결합니다.

35 · chain-of-thought

판단 근거

결론을 이해하고 다음 결정을 내리려면 선택 이유, 사용한 근거, 확인이 필요한 가정을 받는 것이 좋습니다. 실제로 진행한 작업과 앞으로 할 일을 구분하면 결과를 판단하기 쉬워집니다.

입력 → 결과 문제·검토 기준 → 결론·핵심 이유·확인 항목

> “결론을 먼저 적고, 선택 이유 세 가지와 확인이 필요한 가정만 정리해줘. 실제로 실행한 단계와 앞으로 할 단계를 구분해줘.”

확인 내부 독백의 길이가 아니라 근거의 적합성, 가정의 명확성, 실제 결과의 재현 가능성으로 판단합니다.

36 · constraint-spec

출력 조건

결과가 지켜야 할 범위와 형식을 구체화하는 방법입니다. 디자인 취향, 필수 내용, 금지 사항, 완료 조건을 분리하면 충돌을 찾기 쉽습니다.

입력 → 결과 필수 조건·제외 범위 → 검사 가능한 규칙

> “한국어로 쓰고 흑백만 사용해줘. 42개 이름을 유지하고, 각 항목에 쓰는 때·입출력·예시·확인 기준을 넣어줘.”

확인 ‘자세하게’ 같은 표현을 실제 항목으로 풀어 씁니다. 조건이 충돌하면 임의로 삭제하지 말고 우선순위를 정합니다.

경계와 근거를 분명하게 쓰기

37 · guardrails

작업 경계

작업이 넘어가면 안 되는 범위와 확인 지점을 정하는 방법입니다. 외부 발송, 공개, 삭제처럼 결과가 밖으로 나가는 행동은 범위를 분명히 합니다.

입력 → 결과 허용 행동·권한·자료 범위 → 실행 경계

> “이 자료로 초안을 만들어줘. 고객 이름은 예시로 바꾸고, 외부 발송은 하지 말고 검토할 수 있는 파일로 저장해줘.”

확인 지시문만으로 모든 위험을 차단할 수는 없습니다. 필요한 권한 제한과 실제 도구의 제어도 함께 확인합니다.

38 · trust-calibration

근거 수준

확인한 사실과 추정, 제안의 강도를 맞추는 방법입니다. 근거가 부족하면 확신을 키우지 말고 확인이 필요한 부분을 보이게 합니다.

입력 → 결과 주장·출처·실행 기록 → 근거 수준에 맞는 문장

> “각 설명에서 확인한 사실, 추정, 제안을 구분해줘. 근거가 약한 문장은 단정하지 말고 어떤 자료를 확인해야 하는지 알려줘.”

확인 출처가 있다는 사실만으로 충분하지 않습니다. 실제 문맥이 해당 주장을 뒷받침하는지 비교합니다.

39 · transparency-patterns

상태 설명

사용자가 현재 상태와 한계를 알아볼 수 있게 설명하는 방법입니다. 무엇을 읽었고 무엇이 저장됐는지를 확인 가능한 단위로 말합니다.

입력 → 결과 작업 상태·확인 범위 → 명확한 상태 안내

> “파일 생성, 본문 저장, 이미지 삽입, 최종 읽기 확인의 상태를 구분해줘. 하지 않은 검사는 완료라고 쓰지 마.”

확인 근거 없는 정확도 퍼센트를 붙이지 않습니다. 미확인 내용은 구체적으로 쓰고 확인 방법을 제시합니다.

검사하고 다음 작업에 넘기기

40 · quality-rubrics

품질 기준

완성도를 취향 대신 같은 질문으로 비교하는 기준표입니다. 정확성, 빠진 내용, 가독성, 동작, 전달 가능성을 결과에서 직접 확인합니다.

입력 → 결과 완료 조건·산출물 → 통과/수정 기준표

> “누락, 이해도, 화면 가독성, 실제 동작, 근거 다섯 항목으로 검사해줘. 실패한 항목은 위치와 수정 방법을 적어줘.”

확인 점수만 적지 말고 관찰한 결과를 남깁니다. 필수 조건 실패를 다른 장점의 점수로 상쇄하지 않습니다.

41 · task-decomposition

작업 분해

큰 목표를 산출물과 의존성으로 나누는 방법입니다. 일을 나누는 것과 여러 에이전트를 동시에 쓰는 것은 별개입니다.

입력 → 결과 최종 목표·선행 조건 → 순서가 있는 작업 목록

> “원문 확인→구성→제작→확인으로 나눠줘. 독립적으로 진행할 수 있는 일만 병렬 후보로 표시해줘.”

확인 같은 파일을 동시에 수정하지 않게 합니다. 분업의 조율 비용이 더 크면 한 작업자가 순서대로 진행합니다.

42 · handoff-protocols

인계

다음 사람이나 다음 작업이 이어받을 문맥을 전달하는 방법입니다. 결과 링크, 최신 상태, 남은 일, 실행 조건을 한곳에 모읍니다.

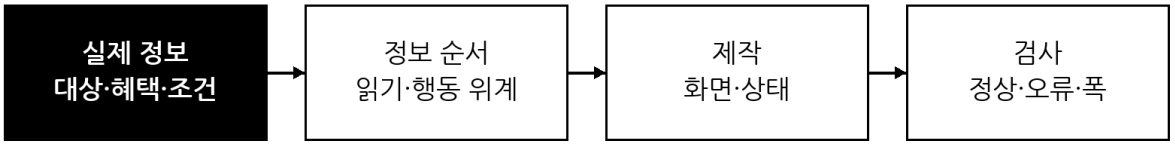
입력 → 결과 최신 파일·결정·확인 결과 → 인계 문서

> “최종 결과 위치, 확정 조건, 실행한 확인, 미해결 항목, 다음 행동을 적어줘. 이전 버전과 현재 버전을 혼동하지 않게 해줘.”

확인 링크가 실제 결과를 가리키는지 열어 봅니다. 저장 전 초안이나 계획을 완료 결과처럼 전달하지 않습니다.

실습 A · 강의 소개 페이지

목표: 방문자가 누구를 위한 강의인지 이해하고 신청 단계로 이동하는 페이지를 만듭니다. 결과물은 작동하는 화면과 확인 기록입니다. 아래 내용은 연습용 예시이며 실제 강의·상품 정보는 자신의 자료로 바꿉니다.



실제 정보와 연습용 예시를 구분하고, 최종 결과를 열어 확인합니다.

정보 순서를 정한 뒤 제작하고, 정상·오류·작은 화면에서 각각 확인합니다.

준비물

강의명, 실제 대상, 제공 내용, 신청 방식이 필요합니다. 가격·일정이 정해지지 않았다면 임의로 채우지 말고 '추후 안내'로 표시합니다. 구현 도구와 브라우저 확인 환경도 준비합니다.

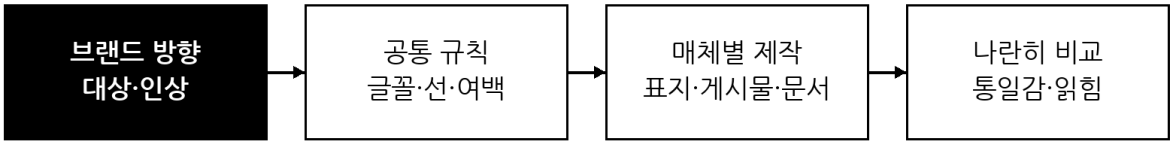
> "아래 정보로 강의 소개 페이지를 만들어줘. 대상은 AI 업무 자동화 입문자이고 핵심 행동은 신청하기야. 실제 제공 정보만 사용해줘. 흑백, 큰 제목, 넓은 여백을 적용하고 제목→얻는 결과→대상→진행 방식→신청 순서로 구성해줘. 입력 누락·처리 중·성공 상태를 넣고 실제 확인 결과를 알려줘."

1. 자료에서 확인된 정보와 비어 있는 정보를 나눕니다.
2. frontend-design 역할로 정보 순서와 위계를 정합니다.
3. brandkit 또는 theme-factory 역할로 표현을 통일합니다.
4. 구현 후 버튼 이동과 누락 입력·성공 처리를 확인합니다.

완료 기준: 신청 경로가 작동하고, 긴 한국어 제목과 작은 화면에서 가로 잘림이 없어야 합니다. 실행 환경이 없으면 화면 설계 초안까지 완료했다고 범위를 적습니다.

실습 B · 흑백 브랜드 에셋 세트

목표: 표지·게시물·안내 문서에서 같은 브랜드처럼 보이는 세트를 만듭니다. 결과물은 브랜드 보드, 표지 시안, 재사용 규칙입니다.



실제 정보와 연습용 예시를 구분하고, 최종 결과를 열어 확인합니다.

브랜드 방향을 공통 규칙으로 정한 뒤 매체별 산출물에 적용하고 함께 비교합니다.

준비물

브랜드 이름, 대상, 전달할 인상 세 단어, 필요한 산출물과 비율을 정합니다. 직접 쓸 로고나 사진 파일이 있다면 준비합니다. 이미지 생성이 필요하면 현재 사용할 수 있는 도구를 확인합니다.

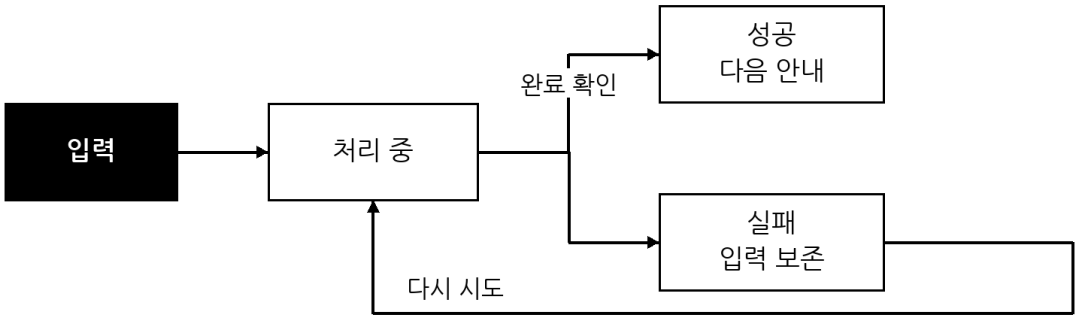
> "브랜드 이름은 {이름}, 대상은 {대상}, 인상은 {세 단어}야. 흰 배경과 검정 요소만 사용해 브랜드 보드를 만들어줘. 제목·본문 글꼴 역할, 선 굵기, 이미지 톤, 여백 원칙을 정하고 세로 표지와 정사각 게시물에 적용해줘. 이미지 결과와 편집 파일의 형식을 따로 알려줘."

1. 이름·대상·인상을 기준으로 시각 방향 하나를 정합니다.
2. brandkit 역할로 규칙을 모으고 12번의 다섯 항목으로 이미지 조건을 씁니다.
3. 필요한 경우 canvas-design이나 현재 사용 가능한 이미지 도구로 시안을 만듭니다.
4. 세 결과를 나란히 놓고 글꼴·여백·선 굵기가 일관되는지 확인합니다.

완료 기준: 실제 크기에서 작은 글자가 읽히고, 한국어와 로고가 정확해야 합니다. 배포할 이미지와 수정에 쓸 편집 파일을 구분해 보관합니다.

실습 C · 실패해도 이어지는 신청 흐름

목표: 신청 버튼을 누른 뒤 사용자에게 현재 상태와 다음 행동을 알려줍니다. 화면의 정상 상태보다 실패했을 때의 경험을 먼저 구체화해 봅니다.



실패 후 입력을 보존하고 재시도 지점으로 돌아갑니다. 원인을 알 수 없으면 단정하지 않습니다.

상태	사용자에게 보일 내용	가능한 행동
입력 중	필수 항목과 입력 예시	입력·수정
처리 중	신청 내용을 보내는 중입니다	중복 제출 방지
실패	신청이 완료되지 않았습니다	입력 보존·다시 시도
성공	신청이 접수되었습니다	다음 절차 보기

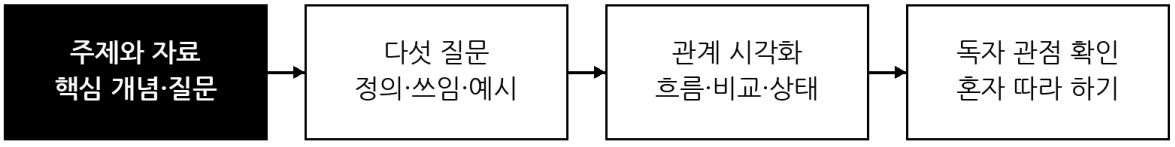
> “신청 흐름을 입력→처리 중→성공/실패로 나눠 설계해줘. 실패하면 입력을 보존하고 다시 시도하게 해줘. 원인이 확인되지 않은 경우 서버 장애라고 단정하지 마. 실제 접수 여부와 화면의 성공 표시가 일치하는지 확인해줘.”

24번으로 수정 범위를 정하고, 25·26번으로 상태와 표시 순서를 설계합니다. 27번으로 복구 안내를 쓰고 28번으로 받은 피드백의 반영 결과를 남깁니다.

완료 기준: 성공 문구는 실제 성공 조건에서만 나타나야 합니다. 화면 설계만 한 경우와 실제 접수까지 시험한 경우를 구분합니다.

실습 D · 혼자 읽고 따라 하는 안내서

목표: 알고 있는 내용을 다른 사람이 혼자 읽고 따라 할 수 있는 안내서로 만듭니다. 문장을 길게 늘리기보다 독자가 궁금해할 질문을 찾고, 설명과 예시를 순서대로 연결합니다.



실제 정보와 연습용 예시를 구분하고, 최종 결과를 열어 확인합니다.

한 줄 요약을 정의·사용 상황·예시·확인 기준으로 확장하고, 개념의 관계에 맞는 도식을 붙입니다.

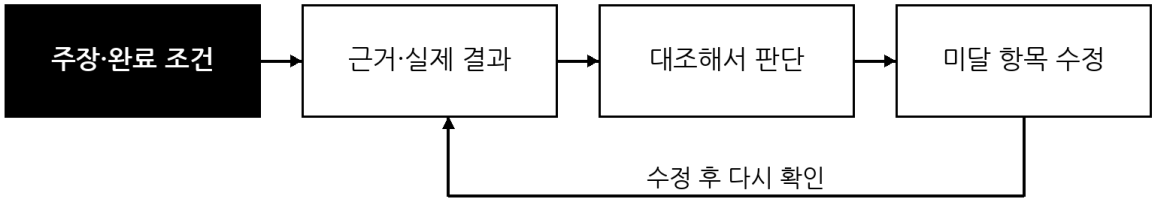
항목 하나를 확장하는 다섯 질문

1. 무엇인가: 낯선 용어를 한 문장으로 풉니다.
2. 언제 쓰는가: 해결할 문제를 구체적으로 씁니다.
3. 무엇을 넣고 받는가: 입력과 결과를 구분합니다.
4. 어떻게 요청하는가: 바로 복사할 예시를 씁니다.
5. 끝났는지 어떻게 아는가: 결과에서 확인할 기준을 붙입니다.

> “{주제}를 처음 배우는 사람을 위한 안내서를 만들어줘. 항목마다 정의·쓰는 때·입출력·요청 예시·확인 기준을 넣어줘. 순서는 흐름도, 차이는 비교표, 실패와 복구는 상태도로 보여줘. 실제 정보와 연습용 예시는 구분하고, 읽는 사람이 준비부터 실행까지 혼자 따라 할 수 있게 써줘.”

완료 기준: 독자가 준비물과 실행 순서, 예상 결과를 이해할 수 있어야 합니다. 설명과 도식이 같은 관계를 보여주는지 살펴보고, 막혔을 때 돌아갈 단계도 안내합니다.

품질을 취향 대신 기준으로 확인하기



주장과 근거, 완료 조건과 실제 결과를 대조합니다. 실패한 항목은 수정 후 다시 확인합니다.

확인할 것	통과 기준	미달이면 고칠 곳
내용 누락	필수 항목과 핵심 조건이 빠짐없이 있음	목차와 필수 항목 목록
이해도	용어·사용 상황·예시가 연결됨	첫 설명과 실습
가독성	읽는 순서·여백·제목 위계가 분명함	제목·문단·표
도식	연결 방향과 분기 조건이 설명됨	선·라벨·범례
실제 동작	요청한 행동이 결과에서 관찰됨	실행 환경·오류 처리
근거와 상태	확인·미확인·제안이 구분됨	출처·완료 표현
전달	최종 위치와 남은 일이 명확함	인계 내용

확인 기록은 짧고 구체적으로

“좋아 보임” 대신 “신청 버튼을 눌러 대상 페이지로 이동함”, “모바일 폭에서 가로 스크롤 없음”처럼 관찰한 결과를 씁니다. 검사를 실행하지 못했다면 그 항목은 미확인으로 남깁니다.

한 가지 필수 조건이 실패했다면 다른 항목의 점수로 상쇄하지 않습니다. 같은 결과물을 바꾸지 않았다면 동일한 검사를 형식상 반복할 필요는 없습니다.

복사해서 쓰는 요청문

01 · 새로 만들기

> “{대상}을 위한 {결과물}을 만들어줘. 목적은 {행동 또는 이해 목표}야. 입력은 {원문·링크·데이터}이고, 반드시 포함할 내용은 {목록}이야. {흑백·톤·크기}를 지키고, 결과는 {형식·저장 위치}로 제공해줘. 완료 전에 {확인 기준}을 실제 결과에서 확인해줘. 모르는 정보는 만들지 말고 표시해줘.”

02 · 필요한 곳만 고치기

> “현재 결과물의 {위치}에서 {문제}가 있어. {확정 조건}은 유지하고 {수정 범위}만 바꿔줘. 바꾼 뒤 {대표 사례·예외 사례}를 확인해줘. 바뀐 내용과 남은 문제를 알려줘.”

03 · 결과 검사하기

> “이 결과를 {원문·완료 조건}과 비교해줘. 빠진 조건, 근거 없는 주장, 읽기 어려운 부분, 동작 오류를 구분해줘. 각 항목에 위치·관찰 결과·수정 제안을 적고, 실제로 확인하지 못한 항목은 미확인으로 남겨줘.”

04 · 다음 작업에 넘기기

> “다음 사람이 이어받도록 최종 결과 링크, 최신 버전, 확정 조건, 실행한 확인과 결과, 미해결 항목, 다음 행동을 정리해줘. 초안과 완료된 파일을 구분하고, 링크가 실제 결과를 가리키는지 확인해줘.”

중괄호는 사용 전에 직접 채우세요. 빈칸을 남긴 채 실행하면 AI가 임의로 가정할 수 있습니다. 민감한 입력이 필요 없다면 예시 자료를 사용하세요.

적용 전, 이름보다 원문을 확인하세요

이 장은 42개를 일괄 설치하는 설명서가 아닙니다. 사용할 역할을 정하고, 실제 배포처·호환 환경·필요 권한을 확인하는 순서입니다. 일반 설계 개념은 별도 설치 없이 요청문과 작업 기준에 적용할 수 있습니다.

1. 할 일 하나를 고릅니다. 예: 소개 페이지의 정보 순서 개선.
2. 해당 이름의 공식·작성자 저장소를 확인합니다. 이름만 같은 자료와 구분합니다.
3. README와 SKILL.md에서 목적·설치 경로·호환 도구·라이선스를 읽습니다.
4. 외부 서비스 연결이 필요하면 로그인·파일 접근·사용 조건을 확인합니다.
5. 작은 예제로 실행하고, 지침 적용 여부와 실제 결과를 함께 확인합니다.

시작할 때 참고할 자료

Anthropic Skills 저장소 — 스킬의 구조와 공개 예시

Impeccable 작성자 저장소 — 프로젝트 설명과 적용 안내

Microsoft Playwright MCP — 브라우저 연결 도구

이 책의 이름을 찾을 수 없다면

출처가 없는 이름을 추정해서 설치하지 마세요. 해당 항목의 입력·결과·확인 기준을 일반 요청문으로 먼저 적용할 수 있습니다. 실제 자동화가 필요하면 현재 환경에 있는 공식 도구를 선택하고, 같은 역할을 수행하는지 작은 작업으로 확인하세요.

스킬 파일을 읽었다는 사실과 지침이 결과에 적용됐다는 사실은 다릅니다. 도구 연결 성공과 최종 결과 저장 성공도 따로 확인해야 합니다.

자주 막히는 지점과 수정 방향

증상	먼저 확인할 것	수정 요청 예시
전부 비슷한 화면	대상·내용·시각 방향이 있는가	“정보 순서를 다시 잡아줘.”
도구가 실행되지 않음	설치·연결·접근 가능한가	“현재 가능한 실행 범위를 확인해줘.”
이미지 글자가 틀림	이미지 속 실제 표기	“배경만 만들고 글자는 편집 요소로 분리해줘.”
모션이 산만함	각 효과의 정보 전달 목적	“목적 없는 반복 효과를 제거해줘.”
수정할 때 전체가 변함	유지할 영역·변경 범위	“이 문단과 이 버튼만 바꿔줘.”
출처는 있지만 설명이 다름	원문의 조건·주체·지원 형식	“해당 설명을 원문과 다시 비교해줘.”
완료라는데 결과가 없음	실제 파일·링크·저장 상태	“저장된 결과를 다시 열어 확인해줘.”

다시 요청하기 전에 세 가지만 적으세요

1. 어디에서: 화면·장·문단·상태를 특정합니다.
2. 무엇이 다른가: 기대와 관찰 결과를 나란히 씁니다.
3. 어디까지 바꿀까: 유지할 조건과 수정 범위를 정합니다.

> “24장 상태표의 실패 안내가 원인을 단정하고 있어. 원인이 확인되지 않았다는 조건을 반영해 그 문구만 고쳐줘. 입력 보존과 재시도 흐름은 유지해줘.”

문제가 명확할수록 필요한 수정도 작아집니다. 전체를 다시 만드는 요청은 기존에 잘된 부분까지 바꿀 수 있으니 범위를 먼저 정하세요.

용어 사전·더 읽을거리

헷갈리기 쉬운 개념

스킬은 작업 방법을, 연결 도구는 실행 통로를, 프롬프트는 이번 작업의 구체적인 조건을 정합니다.

지침을 읽는 것과 실제 작업을 수행하는 것은 다릅니다. 화면을 설계했다면 구현과 동작 확인이 이어지고, 문서를 작성했다면 저장된 결과를 열어 독자가 읽을 수 있는지 살펴봅니다.

더 읽을거리

[1] Anthropic · Skills와 SKILL.md 구조

[2] frontend-design · 시각 방향과 UI 디자인

[3] canvas-design · 정적 PDF·PNG 시각물

[4] algorithmic-art · 규칙과 매개변수형 아트

[5] theme-factory · 테마 적용

[6] Impeccable · 작성자 프로젝트

[7] Playwright MCP · Microsoft 프로젝트

도구를 선택할 때는 목적과 출력 형식, 현재 사용할 수 있는 환경을 함께 살펴보세요.

짧은 용어 풀이

UI: 화면에서 보고 누르는 요소. 인터랙션: 행동에 따라 화면이 반응하는 방식. 목업: 실제처럼 구체화한 화면 시안. 에셋: 이미지·영상 등 제작에 쓰는 재료. 루브릭: 같은 기준으로 결과를 확인하는 평가표. 인계: 다음 작업이 이어지도록 결과와 문맥을 전달하는 일.

도구의 기능과 사용 방식은 바뀔 수 있습니다. 실제로 적용할 때는 현재 안내를 살펴보고, 작은 예제로 원하는 결과가 나오는지 확인하세요.