

바이브 메이커 · 무료 자료

클로드 디자인 스킬 5
설치 명령 +
바로 쓰는 프롬프트

클로드로 만든 화면이 어딘가 뻔해 보일 때 까는 스킬 다섯. 무엇을 하는지, 어떻게 까는지, 무엇이라고 시키는지.
영상에서 가린 「설치 명령 다섯 줄」과 「만든 화면을 다섯 기준으로 손보는 프롬프트」도 여기 있습니다.

frontend-design · ui-ux-pro-max · hallmark · emil-design-eng · web-design-guidelines

목차

한눈에	화면이 뻔한 건 «기본값» 때문이다	03
까는 법	까는 길은 두 가지, 준비물은 셋	04
스킬 1 · 방향 앤트로픽 공식	frontend-design — 주제부터 정하고 그다음 디자인	05
스킬 2 · 규칙	ui-ux-pro-max — 업종에 맞는 디자인 시스템	06
스킬 3 · 검사 문서 기준	hallmark — AI 티 나는 구성을 거른다	07
스킬 4 · 움직임 문서 기준	emil-design-eng — 곡선과 속도를 잡는다	08
스킬 5 · 점검 Vercel 공식 문...	web-design-guidelines — 코드를 규칙과 대조	09
쓰는 순서	한 화면을 만들 때 다섯을 쓰는 순서	10
영상에서 가린 것 ①	설치 명령 다섯 줄	11
영상에서 가린 것 ②	만든 화면을 다섯 기준으로 손보는 프롬프트	12
막히면	증상 → 할 일	13

한눈에

화면이 뻔한 건 «기본값» 때문이다

클로드에게 화면을 맡기면 주제와 상관없이 비슷한 모양이 나옵니다. 같은 둥근 카드 여러 장, 장식용 그라데이션, 어디서나 쓰는 서체. 앤트로픽의 frontend-design 스킬 문서도 이걸 「AI 가 만든 디자인이 몰리는 특징」으로 따로 적어 둡니다. 아래 다섯은 그 기본값을 각자 다른 지점에서 막습니다.

스킬	막는 곳	만든 곳 · 깃허브 별
frontend-design	방향 — 주제 · 재료에서 디자인을 고른다	앤트로픽 공식 anthropics/skills · 180,097
ui-ux-pro-max	규칙 — 업종에 맞는 스타일 · 색 · 서체를 한 세트로	nextlevelbuilder/ui-ux-pro-max-skill · 133,911 · MIT
hallmark	검사 — AI 티 나는 구성을 관문 57개로 거른다	Nutlope/hallmark · 29,750 · MIT
emil-design-eng	움직임 — 애니메이션 곡선 · 속도	emilkowalski/skills · 44,180 · MIT
web-design-guidelines	점검 — 접근성 · UX 규칙으로 코드를 대조	Vercel 공식 vercel-labs/agent-skills · 32,069

별 수 = 각 저장소에서 2026-10-08 확인. 숫자는 매일 바뀝니다.

이 순서로 보세요

1. 까는 법 두 가지와 준비물 (4쪽)
2. 스킬 하나씩 — 무엇을 하고, 어떻게 까고, 무엇이라고 시키나 (5~9쪽)
3. 한 화면을 만들 때 다섯을 쓰는 순서 (10쪽)
4. 설치 명령 다섯 줄 (11쪽) · 만든 화면을 손보는 프롬프트 (12쪽)

> 제일 빠른 길은 **11쪽의 명령을 그대로 붙여 넣고, 12쪽의 프롬프트를 지금 만들고 있는 화면에 돌리는 것**입니다. 5~9쪽은 각 스킬이 무엇을 하는지 알고 싶을 때 보세요.

까는 법

까는 길은 두 가지, 준비물은 셋

길	어디서 치나	이 자료에서 쓰는 스킬
플러그인 `/plugin install …`	클로드코드 대화창 안	frontend-design · ui-ux-pro-max
skills 명령 `npx skills add …`	터미널(맥) · 파워쉘(윈도우)	hallmark · emil-design-eng · web-design-guidelines

skills 명령이 까는 자리

붙이는 것	깔리는 곳	언제
아무것도 안 붙임	지금 폴더의 `.claude/skills/`	이 프로젝트에서만 쓸 때
`-g`	내 컴퓨터의 `~/.claude/skills/`	모든 프로젝트에서 쓸 때 (이 자료의 기본)
`-a claude-code`	클로드코드에만	다른 AI 도구에는 안 깔고 싶을 때
`--skill 이름`	저장소 안의 그 스킬 하나만	스킬 묶음 저장소에서 하나만 고를 때

출처 = npx skills 공식 README(vercel-labs/skills) 「Options」 · 「Installation Scope」

준비물

- **클로드코드** — 설치 · 부르기 전부 여기서
- **Node.js** — `npx` 가 Node 와 함께 깔린다. 확인: `node -v`
- **Python 3** — ui-ux-pro-max 의 검색 스크립트가 쓴다. 확인: `python3 --version`

복사해서 그대로 쓰세요

```
node -v
python3 --version
npx skills list
```

> 버전 숫자가 나오면 있는 것입니다. 마지막 줄은 지금 깔린 스킬 목록을 보여 줍니다. 깐 뒤 클로드코드를 새로 열어야 새 스킬을 알아봅니다.

스킬 1 · 방향 앤트로픽 공식

frontend-design — 주제부터 정하고 그다음 디자인

스킬 문서는 클로드에게 「뻔한 틀을 이미 거절한 고객을 받은 디자인 스튜디오」처럼 굴라고 시킵니다. 무엇을 만드는지(업종 · 주제 · 재료 · 그 분야의 말투)를 먼저 정하고, 거기서 색 · 서체 · 배치를 고르게 합니다. 문서에 적힌 «AI 디자인이 몰리는 특징»은 이렇습니다.

문서가 꼽은 기본값	대신 하게 하는 것
같은 둥근 카드로 잘게 나눈 구성 · 똑같은 회색 그림자 · 장식용 그라데이션	정보의 무게에 따라 모양을 다르게
어느 프로젝트에나 쓰는 기본 서체	서체를 일부러 고르고, 한두 가족만
제목 위마다 붙는 영문 대문자 라벨 · 링크마다 붙는 →	정보가 있을 때만 구조 장치를 쓴다
섹션마다 아래에서 떠오르는 등장 효과	움직임은 한 번, 사람이 무언가 했을 때

설치 — 클로드코드 대화창에

복사해서 그대로 쓰세요

```
/plugin install frontend-design@claude-plugins-official
```

바로 써 보기

복사해서 그대로 쓰세요

frontend-design 스킬로 우리 동네 꽃집 소개 페이지를 만들어 줘. 만들기 전에 이 가게의 주제 · 손님 · 페이지가 할 일을 한 줄씩 먼저 정하고 나한테 확인받아. 같은 모양 카드 반복, 장식용 그라데이션, 기본 서체는 쓰지 마.

> 스킬은 이름을 말하지 않아도 화면 작업이면 클로드가 알아서 꺼내 씁니다. 확실히 쓰게 하려면 위처럼 이름을 넣으세요.

스킬 2 · 규칙

ui-ux-pro-max — 업종에 맞는 디자인 시스템

색 팔레트 192개 · UI 스타일 79개(지금 쓰이는 것 50개) · 폰트 조합 74개를 데이터로 들고 있습니다. 업종을 말하면 그중에서 페이지 구성 · 스타일 · 색 다섯 · 서체 짝을 한 세트로 골라 줍니다. 페이지마다 색과 서체가 흔들리는 걸 막는 «규칙집»을 먼저 만드는 셈입니다.

README 의 예시 출력 — `search.py "beauty spa wellness" --design-system`

항목	골라 준 값
PATTERN (구성)	Hero-Centric + Social Proof
STYLE	Soft UI Evolution — 부드러운 그림자 · 차분함
COLORS	#E8B4B8 #A8D5BA #D4AF37 #FFF5F5 #2D3436
TYPOGRAPHY	Cormorant Garamond / Montserrat

설치 — 클로드코드 대화창에 두 줄

복사해서 그대로 쓰세요

```
/plugin marketplace add nextlevelbuilder/ui-ux-pro-max-skill
/plugin install ui-ux-pro-max@ui-ux-pro-max-skill
```

터미널이 편하면 `npm install -g ui-ux-pro-max-cli` 다음 `uipro init --ai claude` 도 됩니다 (README).

바로 써 보기

복사해서 그대로 쓰세요

ui-ux-pro-max 로 「동네 꽃집 소개 페이지」에 맞는 디자인 시스템을 하나 뽑아 줘. 페이지 구성 · 스타일 · 색 다섯 · 서체 짝을 표로 보여 주고, 내가 고르면 그 값을 프로젝트 전체 색 · 서체 변수로 정리해 줘.

Python 3 가 있어야 합니다(검색 스크립트 · 표준 라이브러리만 씀). README 에는 「AI 가 대신 설치하지 말고 사람에게 묻게 했다」고 적혀 있어서, 없으면 클로드가 설치 방법을 알려 주고 멈춥니다.

스킬 3 · 검사 문서 기준

hallmark — AI 티 나는 구성을 거른다

페이지를 만들 때 구성 하나를 고르고, 테마 21개 중 하나를 입히고, **슬롭 검사 관문 57개**와 내보내기 전 자기 채점을 거칩니다. 스스로를 「AI 슬롭 반대 디자인 스킬」이라고 소개합니다. 동사 네 개로 부릅니다.

부르는 말	하는 일
(동사 없이)	새 화면을 만든다. 넘기기 전에 슬롭 검사
`hallmark audit 대상`	기존 코드를 채점해 고칠 목록만 낸다. **코드는 안 고친다**
`hallmark redesign 대상`	문구 · 정보 구조 · 브랜드는 두고 구조를 새로
`hallmark study 캡처 · 주소`	마음에 드는 디자인의 «DNA»(구성 · 서체 짝 · 기준 색)를 뽑는다. 픽셀 복제는 거절

관문 몇 개 (SKILL.md 원문)

- **관문 46** 지어낸 수치 — 「+47% 전환」「5만 팀이 사용」 같은 근거 없는 숫자는 실제 값이나 빈칸 표시로
- **관문 38a** 제목 속 기울인 한 단어 — 강조는 굵기 · 색 · 밑줄로
- **관문 47** 그려 넣은 가짜 브라우저 창 · 폰 테두리 — 진짜 캡처를 쓰거나 테두리를 뺀다

설치 — 터미널에

복사해서 그대로 쓰세요

```
npx skills add nutlope/hallmark -g -a claude-code
```

바로 써 보기

복사해서 그대로 쓰세요

hallmark audit src/app/page.tsx
걸린 항목마다 관문 번호와 고칠 방법을 한 줄씩 적어 줘. 코드는 아직 고치지 마.

> 새로 만들 때는 hallmark 가 먼저 「누가 보나 · 어디에 쓰나 · 어떤 분위기인가」 세 가지를 묻습니다. 「알아서 해」라고 답하면 스스로 추정해서 진행합니다(SKILL.md).

스킬 4 · 움직임 문서 기준

emil-design-eng — 곡선과 속도를 잡는다

Vercel · Linear 에서 일한 디자인 엔지니어 에밀 코왈스키가 「에이전트가 자주 하는 작은 실수」를 모아 둔 스킬입니다. 대부분 애니메이션이고 디자인 조언이 조금 섞여 있습니다. 문서의 전후표가 그대로 체크리스트입니다.

전	후	이유 (문서)
`transition: all 300ms`	`transition: transform 200ms ease-out`	바뀌는 속성만 정확히
`transform: scale(0)`	`scale(0.95)` + 투명에서 시작	현실에 없던 게 갑자기 생기지 않는다
드롭다운에 `ease-in`	`ease-out` 계열 곡선	느리게 출발하면 굼떠 보인다
버튼에 눌림 반응 없음	`:active` 에 `scale(0.97)`	누른 게 느껴져야 한다

얼마나 빨라야 하나 (문서 표)

버튼 눌림 반응	툴팁 · 작은 팝오버	드롭다운 · 셀렉트	모달 · 서랍
100~160ms	125~200ms	150~250ms	200~500ms

문서 규칙: UI 애니메이션은 300ms 아래로 · 강한 ease-out = `cubic-bezier(0.23, 1, 0.32, 1)`

설치 — 터미널에 (묶음 중 이 스킬 하나만)

복사해서 그대로 쓰세요

```
npx skills@latest add emilkowalski/skills --skill emil-design-eng -g -a claude-code
```

바로 써 보기

복사해서 그대로 쓰세요

emil-design-eng 기준으로 이 프로젝트의 애니메이션을 검토해 줘. 전 · 후 · 이유 세 칸 표로 보여 주고, 내가 확인하면 고쳐 줘.

같은 저장소에 `review-animations`(엄격 검토) · `improve-animations`(전체 감사 → 고칠 계획)도 있습니다. 필요하면 `--skill` 뒤에 이름을 더 붙이세요.

스킬 5 · 점검 VERCEL 공식 문서 기준

web-design-guidelines — 코드를 규칙과 대조

화면 코드를 웹 인터페이스 가이드라인(규칙 100개 넘게)과 대조합니다. 검사할 때마다 **최신 규칙 원문을 새로 받아 와서** 비교하고, 결과는 `파일:줄 – 무엇` 형식으로 짧게 냅니다. 편집기에서 바로 그 줄로 갈 수 있는 형식입니다.

대조하는 규칙 예 (원문 → 우리말)

규칙	걸리는 코드
아이콘만 있는 버튼엔 `aria-label`	`<button><SearchIcon /></button>`
입력칸엔 `<label>` 이나 `aria-label`, 그리고 `autocomplete`	`<input type="email" />`
`outline: none` 을 쓰면 다른 포커스 표시를	`className="outline-none"`
버튼 이름은 구체적으로(「계속」 말고 「API 키 저장」)	`<button>계속</button>`
클릭 동작은 `<button>`, 이동은 `<a>`	`<div onClick>`

그 밖의 범위 — 포커스 · 폼 · 애니메이션(움직임 줄이기 설정) · 글자 · 이미지 · 성능 · 내비게이션 · 다크 모드 · 터치 · 날짜 · 숫자 형식 (README)

설치 — 터미널에 (묶음 중 이 스킬 하나만)

복사해서 그대로 쓰세요

```
npx skills add vercel-labs/agent-skills --skill web-design-guidelines -g -a claude-code
```

바로 써 보기

복사해서 그대로 쓰세요

web-design-guidelines 로 src/ 아래 화면 코드를 검사해 줘. 파일:줄 형식으로 받은 뒤, 접근성 문제부터 순서대로 고쳐 줘.

> 파일을 안 알려 주면 어떤 파일을 볼지 되묻습니다(SKILL.md). 저장소 묶음엔 리액트 성능 규칙(`react-best-practices`) 같은 다른 스킬도 있습니다.

쓰는 순서

한 화면을 만들 때 다섯을 쓰는 순서

다섯은 겹치지 않고 단계가 다릅니다. 만들기 전 → 만드는 중 → 만든 뒤 순서로 놓으면 서로 하는 일을 덮어쓰지 않습니다.

1. **방향** — frontend-design
   주제 · 손님 · 페이지가 할 일을 먼저 정한다. 여기서 정한 게 뒤의 기준이 된다
2. **규칙** — ui-ux-pro-max
   그 업종에 맞는 스타일 · 색 다섯 · 서체 짝을 고르고 변수로 고정한다
3. **만들기** — 클로드에게 화면을 맡긴다
   1 · 2 에서 정한 것을 지키라고 한 줄 붙인다
4. **검사** — hallmark audit
   AI 티 나는 곳을 관문 번호로 받는다. 지어낸 숫자부터 지운다
5. **움직임** — emil-design-eng
   transition: all · ease-in · 눌림 반응 없음을 찾아 고친다
6. **점검** — web-design-guidelines
   마지막에 접근성 · UX 규칙으로 파일:줄 목록을 받아 정리한다

무엇이 걸릴 때 어느 스킬인가

이럴 때	먼저 꺼낼 스킬
처음부터 새로 만든다 · 방향이 안 잡힌다	frontend-design
페이지마다 색 · 서체가 다르다	ui-ux-pro-max
다 만들었는데 «AI 가 만든 것 같다»는 말을 듣는다	hallmark audit
메뉴 · 모달이 굼뜨거나 툭툭 끊긴다	emil-design-eng
키보드로 안 눌린다 · 입력칸이 불편하다	web-design-guidelines

> 처음이면 **frontend-design 하나만** 깔아도 차이가 큽니다. 앤트로픽 공식이고, 플러그인 한 줄이면 끝납니다.

영상에서 가린 것 ①

설치 명령 다섯 줄

클로드코드 대화창에 (플러그인 둘)

복사해서 그대로 쓰세요

```
/plugin install frontend-design@claude-plugins-official
/plugin marketplace add nextlevelbuilder/ui-ux-pro-max-skill
/plugin install ui-ux-pro-max@ui-ux-pro-max-skill
```

터미널 · 파워쉘에 (skills 명령 셋)

복사해서 그대로 쓰세요

```
npx skills add nutlope/hallmark -g -a claude-code
npx skills@latest add emilkowalski/skills --skill emil-design-eng -g -a claude-code
npx skills add vercel-labs/agent-skills --skill web-design-guidelines -g -a claude-code
```

다 깔았으면 클로드코드를 껐다 켜고, 터미널에서 `npx skills list -g` 로 셋이 보이는지 확인합니다. 플러그인 둘은 클로드코드에서 `/plugin` 을 쳐서 설치된 목록을 봅니다.

명령이 낯설면 — 클로드코드에 이 문장 하나

복사해서 그대로 쓰세요

아래 다섯 개 디자인 스킬을 내 클로드코드에 설치해 줘. 각 저장소 README 의 설치 방법을 그대로 따르고, 모든 프로젝트에서 쓰게 사용자 폴더에 깔아 줘. 시작 전에 node -v 와 python3 --version 으로 준비물을 확인하고, 없는 게 있으면 설치 방법만 알려 주고 멈춰. 다 깔면 각 스킬 이름과 깔린 위치를 표로 보여 줘.
1) https://github.com/anthropics/claude-plugins-official (frontend-design 플러그인)
2) https://github.com/nextlevelbuilder/ui-ux-pro-max-skill
3) https://github.com/Nutlope/hallmark
4) https://github.com/emilkowalski/skills (emil-design-eng 만)
5) https://github.com/vercel-labs/agent-skills (web-design-guidelines 만)

> 설치 중에 클로드가 명령 실행 허락을 물으면, 무엇을 실행하는지 읽고 승인하세요. `/plugin` 으로 시작하는 줄은 클로드가 대신 칠 수 없어서 직접 쳐야 할 수 있습니다.

영상에서 가린 것 ②

만든 화면을 다섯 기준으로 손보는 프롬프트

이미 만든 화면이 있다면 아래를 그대로 붙여 넣으세요. 한 번에 다 고치지 않고, 단계마다 보고하고 멈추게 짜 두었습니다. 마음에 안 드는 단계는 건너뛰라고 하면 됩니다.

복사해서 그대로 쓰세요

지금 이 프로젝트의 메인 화면을 다섯 단계로 손봐 줘. 단계마다 무엇을 바꿨는지 세 줄로 보고하고, 내가 「다음」이라고 하면 넘어가.
1) frontend-design 기준으로 이 화면의 주제 · 손님 · 할 일을 한 줄씩 먼저 적어. 그 주제와 상관없는 기본값(같은 모양 카드 반복, 장식용 그라데이션, 기본 서체, 제목마다 붙은 영문 대문자 라벨)을 찾아 바꿔.
2) ui-ux-pro-max 로 이 업종에 맞는 디자인 시스템을 하나 뽑아서, 화면의 색과 서체를 그 값으로 통일해.
3) hallmark audit 으로 AI 티 나는 곳을 목록으로 받아. 근거 없는 숫자는 실제 값이나 「확인 필요」 표시로 바꾸고 나머지도 고쳐.
4) emil-design-eng 기준으로 transition: all 과 ease-in 을 찾아 고치고, 버튼에 눌림 반응을 넣어. 바꾼 것은 전 · 후 · 이유 표로.
5) web-design-guidelines 로 접근성 · UX 규칙 위반을 파일:줄 목록으로 받아서 전부 고쳐.
끝나면 바뀐 파일 목록과, 가능하면 고치기 전 · 후 화면을 나란히 보여 줘.

새로 만들 때 첫 문장

복사해서 그대로 쓰세요

{만들 것}을 만들 거야. frontend-design 으로 주제 · 손님 · 할 일을 먼저 정해서 나한테 확인받고, ui-ux-pro-max 로 그 업종의 디자인 시스템을 고른 다음 그 규칙대로 화면을 만들어 줘. 다 만들면 hallmark audit 으로 스스로 검사해서 걸린 것까지 고친 뒤 보여 줘.

> 스킬이 안 깔려 있으면 클로드는 그 단계를 일반 지식으로 대신합니다. 결과가 뻔하게 나오면 11쪽 설치부터 확인하세요.

막히면

증상 → 할 일

증상	할 일
`/plugin install frontend-design@claude-plugins-official` 가 마켓을 못 찾는다	먼저 `/plugin marketplace add anthropics/claude-plugins-official` 로 공식 마켓을 등록하고 다시
`npx: command not found`	Node.js 가 없다. nodejs.org 에서 LTS 를 설치하고 터미널을 새로 연다
깔았는데 클로드가 스킬을 안 쓴다	클로드코드를 껐다 켠다. 그래도 안 되면 프롬프트에 스킬 이름을 넣는다(「hallmark audit …」)
`-g` 를 빼고 깔았더니 다른 프로젝트에선 없다	`-g` 없이 깔면 그 폴더의 `.claude/skills/` 에만 깔린다. `-g` 를 붙여 다시
ui-ux-pro-max 가 Python 이 없다며 멈춘다	python.org 에서 Python 3 를 설치(스킬이 대신 설치하지 않게 되어 있다)
emil · vercel 저장소의 스킬이 전부 깔렸다	`--skill 이름` 을 빼먹었다. `npx skills remove 이름` 으로 필요 없는 것만 지운다
hallmark audit 이 코드를 안 고친다	원래 그렇다 — audit 은 목록만 낸다. 「이 목록대로 고쳐 줘」라고 한 번 더
스킬을 최신으로 바꾸고 싶다	`npx skills update` (skills 명령으로 깐 것) · 플러그인은 `/plugin` 메뉴에서

> 이 자료의 사실은 각 저장소의 SKILL.md · README 와 npx skills 공식 README(2026-10-08)에서 옮겼습니다. 스킬은 자주 바뀌니 명령이 안 먹으면 저장소 README 를 먼저 보세요. 같은 자료실(docs.vibemake.kr)에 클로드코드 플러그인 설치 가이드도 있습니다.