

바이브 메이커 · 무료 자료

아레나 스킬 사용 설명서

클로드 답이 마음에 안 들 때, 다시 묻는 대신 클로드 여러 개를 대진표로 싸우게 하는 무료 스킬. 설치 세 가지, 쓰는 법, 심사 기준, 비용, 그리고 결과를 가르는 과제 파일 쓰는 법까지.
영상에서 가린 「돌리기 전 적어 둘 한 가지」가 여기 있습니다.

설치 세 가지 | 쓰는 법 · 옵션 | 심사 기준 100점 | 비용 표 | 과제 파일 쓰는 법

바이브 메이커 Vibe Maker

docs.vibemake.kr · 무료 배포 · 2026-10-21 기준

목차

한눈에	다시 묻는 대신, 클로드 100개를 싸우게 한다	03
설치 세 가지 문서 기준	셋 중 하나만 하면 된다	04
쓰는 법 문서 기준	/arena 뒤에 할 일을 적는다	05
작동 원리 README 원문	같은 과제, 다른 카드, 한 판씩 탈락	06
심사 기준 rubric.md 원문	100점 만점, 다섯 칸	07
비용과 시간 직접 실행	스킬은 무료, 토큰은 내 것	08
영상에서 가린 것 · 과제 파일	100개가 아는 건 과제 파일 하나뿐	09
영상에서 가린 것 · 쓸 일 고르기	틀리면 곤란한 일에, 고르는 건 내가	10
알아 둘 것 README 원문	100개의 클로드, 정확히 무엇인가	11
막히면	증상 → 할 일	12

한눈에

다시 묻는 대신, 클로드 100개를 싸우게 한다

아레나(Arena)는 클로드 코드에 까는 스킬 하나입니다. 답이 마음에 안 들 때 `/arena` 를 치면, 같은 일을 **글자 하나 다르지 않게** 서브 에이전트 100개에게 나눠 줍니다. 둘씩 짝지어 서로의 답을 공격하고, 각자 고치고, 심사 에이전트가 점수를 매겨 진 쪽을 떨어뜨립니다. 100 → 50 → 25 → 13 → 7 → 4 → 2 → 1. 마지막에 **살아남은 답 하나**가 돌아옵니다.

무엇	클로드 코드 스킬 1개 · 대진표 도구 `bracket.py`(파이썬 표준 라이브러리) · 심사 기준표 `rubric.md` · 전략 카드 목록 `strategies.json`
어디	`github.com/Jakeschincariol/arena-skill`
값	스킬은 무료 · MIT 라이선스 · 가입 · API 키 · 연결할 것 없음. 단 **토큰은 내 것**(8쪽) · 깃허브 별 828 (2026-10-21)
돌려받는 것	이긴 답 전문 · 그 답이 버틴 공격 목록 · 그 답의 카드 · 라운드 수 · 마음에 안 든 옛 답이 있었다면 둘의 비교 점수

이 자료와 아레나는 제휴 관계가 없습니다. 이름 · 명령 · 규칙은 저장소의 README · SKILL.md · rubric.md(커밋 df07b8e) 원문을 옮겼습니다.

답을 합쳐 주는 도구가 아니다

여러 답의 좋은 부분을 모아 하나로 붙여 주는 도구로 소개되기도 하지만, 저장소 문서에는 **답을 합치는 단계가 없습니다.** 답마다 상대의 공격을 받고 자기 답을 스스로 고치며, 진 답은 탈락합니다. 끝까지 남는 것은 그렇게 고쳐진 답 하나입니다(6쪽).

이 자료를 만든 쪽이 직접 해 본 것

- **대진표 계산기**(`bracket.py plan`)를 2026-10-21 직접 실행해 라운드 · 호출 수를 확인했습니다(8쪽) 직접 실행
- 100개짜리 실제 대결은 돌려 보지 않았습니다. 설치 · 작동 · 심사 기준은 문서 기준 입니다

설치 세 가지 문서 기준

셋 중 하나만 하면 된다

길 ① 클로드 코드에 붙여넣기 · 가장 쉬움

클로드 코드 대화창에 아래를 그대로 보냅니다. README 원문 그대로라 깔고 나서 `/arena` 가 되는지까지 확인합니다.

복사해서 그대로 쓰세요

https://github.com/Jakeschincariol/arena-skill

Install this skill, then confirm /arena works.

길 ② 폴더 복사 · `/arena` 로 짧게 부르기

복사해서 그대로 쓰세요

```
git clone https://github.com/Jakeschincariol/arena-skill.git
cp -r arena-skill/skills/arena ~/.claude/skills/
```

한 프로젝트에서만 쓰려면 `~/.claude/skills/` 대신 그 프로젝트의 `.claude/skills/` 에 복사합니다.

길 ③ 플러그인

복사해서 그대로 쓰세요

```
/plugin marketplace add Jakeschincariol/arena-skill
/plugin install arena-skill@arena-skill
```

플러그인으로 깔면 이름 앞에 묶음 이름이 붙어 `/arena-skill:arena` 로 부릅니다. 이 자료의 `/arena` 를 그 이름으로 바꿔 쓰세요.

필요한 것	클로드 코드 · 파이썬 3.8 이상. 따로 설치할 패키지는 없습니다
왜 클로드 코드인가	서브 에이전트를 띄우는 Agent 도구를 쓰기 때문이라고 README 가 적습니다
깔렸는지 확인	클로드 코드를 다시 열고 `/` 를 쳐서 목록에 `arena` 가 뜨는지 봅니다

쓰는 법 문서 기준

/arena 뒤에 할 일을 적는다

README 예시 그대로

복사해서 그대로 쓰세요

```
/arena
/arena --quick write the headline for our pricing page
/arena --agents 32 fix the flaky test in tests/test_api.py
/arena --seed 7 plan my launch week, I have 6 hours a day
```

같은 예시를 우리말로

복사해서 그대로 쓰세요

/arena --quick 우리 가격 페이지 맨 위 한 줄을 써 줘
/arena --agents 32 tests/test_api.py 에서 가끔 실패하는 테스트를 고쳐 줘
/arena --seed 7 출시 주간 계획을 짜 줘. 하루 6시간 쓸 수 있어

옵션	하는 일
`--agents N`	몇 개가 싸울지. 기본 100
`--quick`	16개. README 가 「평소에 쓰는 설정」이라고 부릅니다
`--seed S`	같은 숫자면 같은 카드 · 같은 대진표(같은 답은 아님). 안 주면 무작위로 정하고 기록
`--wave W`	한 번에 띄우는 수. 기본 10. 클로드 코드의 동시 실행 한도를 올렸을 때만 같이 올립니다

할 일을 안 적으면

`/arena` 만 치면 **방금 한 요청**이 과제가 되고, **마음에 안 든 그 답**이 이겨야 할 상대가 됩니다.

저절로 시작하나

슬래시 없이 「별로야, 경쟁시켜 봐」처럼 말해도 클로드가 아레나를 꺼낼 수 있습니다. 이렇게 스스로 꺼냈을 때는 쓰기 전에 한 번 묻도록 SKILL.md 에 적혀 있습니다. 전체(100개 · 호출 595번), `--quick`(16개 · 호출 91번), 그냥 다시 답하기 중에서 고릅니다. 확실히 돌리려면 `/arena` 를 직접 치는 쪽이 빠릅니다.

> 시작 직전에 클로드가 accept-edits 모드(Shift+Tab)를 권합니다. 서브 에이전트가 `.arena/` 에 파일을 쓸 때마다 승인 창이 뜨기 때문입니다(100개면 수백 번). 이 설정은 클로드가 대신 바꾸지 않습니다.

작동 원리 README 원문

같은 과제, 다른 카드, 한 판씩 탈락

1 **띄우기**
N개 모두에게 같은 과제 + 서로 다른 전략 카드 한 장. 카드 = 생각하는 방식 15 × 일하는 순서 12 × 전략 12 = 2,160가지, 겹치지 않게 나눠 줍니다

2 **공격**
둘씩 짝(같은 생각 방식끼리는 피함). 상대 답에서 틀린 것 · 빠뜨린 요구 · 깨지는 입력을 최대 7개, FATAL · MAJOR · MINOR 로 표시

3 **방어**
받은 공격마다 인정하거나 근거로 반박하고, 인정한 것은 고쳐서 자기 답을 다시 씁니다

4 **심사**
따로 띄운 심사 에이전트가 고친 두 답을 읽고 공격을 직접 확인한 뒤 기준표로 채점(7쪽). 점수 계산은 bracket.py. 진 쪽은 탈락

5 **반복**
이긴 쪽이 고친 답을 들고 다음 라운드로. 홀수면 한 명 부전승(같은 에이전트가 두 번 받지 않음)

6 **결과**
하나 남으면 그 답 · 버틴 공격 · 카드 · 라운드 수. 마음에 안 든 옛 답이 있었다면 마지막 심사가 둘을 가린 채 비교하고, **옛 답이 이겨도 그대로 알려 줍니다**

카드에 적힌 것 (strategies.json · 일부)

생각하는 방식 15	처음부터 원리로 · 거꾸로 생각하기 · 비유 · 반대편에서 · 제약 먼저 · 풀이 예시 · 소크라테스식 · 일부러 반대 · 시스템 사고 · 쪼개기 · 끝에서 거꾸로 · 확률로 · 근거 먼저 · 전문가 회의 외
일하는 순서 12	초안 → 비판 → 다시 쓰기 · 개요 먼저 · 테스트 먼저 · 조사한 뒤 종합 · 초안 셋 중 하나 고르기 외
전략 12	되는 것 중 가장 단순하게 · 최대한 엄밀하게 · 사용자 공감 먼저 · 예외 먼저 · 속도 외

> 한 경기 = 공격 2 + 방어 2 + 심사 1 = 서브 에이전트 호출 5번. 100개의 1라운드는 50경기라 250번입니다(8쪽 표). 대진 전체는 `.arena/` 안 JSON 파일 하나에 남아서, 대화가 중간에 압축되거나 「그만」으로 멈춰도 그 파일에서 이어 갑니다.

심사 기준 RUBRIC.MD 원문

100점 만점, 다섯 칸

기준	비중	묻는 것
정확성 Correctness	30	맞는가. 틀린 주장 · 논리 오류 · 버그 · 오해를 부를 것이 없는가
완전성 Completeness	25	과제가 **적은** 요구를 다 채웠나. 심사가 바랐던 것이 아니라 과제 글이 기준
견고성 Robustness	20	이번 경기에서 받은 공격을 버텼나. 고쳤거나, 맞게 반박했거나
구체성 Specificity	15	지금 바로 쓸 수 있나. 정확한 단계 · 값 · 이름 · 코드
명료성 Clarity	10	읽고 쓰기 쉬운가. 과제에 맞는 길이인가

칸마다 0~10점 → (정확성×30 + 완전성×25 + 구체성×15 + 견고성×20 + 명료성×10) ÷ 10 = 0~100점. 「7점은 예의상 주는 기본값이 아니다」고 적혀 있습니다.

점수보다 먼저 보는 규칙

- **치명(fatal)** · 심사가 직접 확인한 결함으로 답이 틀렸거나 못 쓰는 경우(돌지 않는 코드 · 핵심 주장이 거짓 · 꼭 지킬 조건 위반 · 다른 질문에 답함). 치명인 답은 점수가 높아도 치명 아닌 답을 이기지 못합니다. 둘 다 치명이면 점수로
- **무승부 없음** · 같으면 남은 공격이 적은 쪽, 그래도 같으면 정확성이 높은 쪽

심사가 점수를 주지 않는 것

길이	길다고 좋지 않다. 요구를 다 채운 짧은 답이 같은 요구를 채운 긴 답을 이긴다
자신감	「고쳤다」는 말은 증거가 아니다. 고친 답을 직접 본다
방식	심사는 카드를 보지 않고 짐작도 하지 않는다. 결과물만 채점
자기 칭찬	「견고하고 포괄적인 답」이라고 써도 0점. 실제로 그래야 한다

> 심사도 클로드입니다. README 는 결과가 「이 대회에서 찾은 가장 강한 답이지, 맞다는 증명은 아니다」라고 적습니다. 기준표는 저장소에 있어 읽고 고칠 수 있습니다.

비용과 시간 직접 실행

스킬은 무료, 토큰은 내 것

README 의 비용 설명 첫 줄입니다. 「The skill is free. The tokens are yours, and 100 agents is a lot of them.」 호출마다 과제와 답 한두 개를 읽으므로, 과제가 클수록 비용도 커집니다.

몇 개	라운드	서브 에이전트 호출	10개씩 묶음
100 · 기본	7	595	70
64	6	379	49
32	5	187	28
16 · `--quick`	4	91	16
8	3	43	10

README 표 그대로. 마음에 안 든 옛 답이 있으면 마지막 비교 1번이 더해집니다.

돌리기 전에 직접 계산해 보기

스킬이 시작 전에 이 숫자를 먼저 보여 주고, 아래 명령으로는 언제든 볼 수 있습니다(아무것도 쓰지 않음). 길 ②로 깔았을 때의 경로입니다.

복사해서 그대로 쓰세요

```
python3 ~/.claude/skills/arena/bracket.py plan --quick
```

2026-10-21 이 자료를 만든 쪽이 직접 돌린 출력

```
16 agents, 4 rounds, waves of 10

  round  alive  matches  bye  sub-agent calls  waves
  spawn     16        -    -                16      2
      1     16        8    -                40      5
      2      8        4    -                20      3
      3      4        2    -                10      3
      4      2        1    -                 5      3
  total                                     91     16

  alive per round: 16 -> 8 -> 4 -> 2 -> 1
```

시간

클로드 코드는 기본으로 도구를 한 번에 10개까지 돌리므로 아레나도 10개씩 묶어 돌립니다. 100개 전체 = 70묶음, `--quick` = 16묶음. README 는 몇 분인지 숫자 대신 「시간이 좀 걸린다」고만 적고, **평소엔 --quick, 100개는 정말 중요한 답에** 아껴 쓰라고 권합니다.

영상에서 가린 것 · 과제 파일

100개가 아는 건 과제 파일 하나뿐

서브 에이전트는 **내 대화를 못 봅니다.** 스킬이 대화를 읽고 `.arena/task.md` 를 따로 쓰는데, 100개가 아는 것은 그 파일이 전부입니다. README 의 경고 그대로, **파일에 안 들어간 요구는 100개가 다 같이 놓칩니다.** 「결과를 가르는 단계」라고 SKILL.md 가 직접 적은 곳도 여기입니다.

과제 파일에 들어가야 하는 것 · SKILL.md 2단계

1. **요청** · 되도록 내가 쓴 말 그대로
2. **대화 어디서든 말한 요구와 조건**
   대상 · 길이 · 형식 · 말투 · 쓰는 기술 · 마감 · 바꾸면 안 되는 것
3. **처음 보는 사람이 알아야 할 배경**
   관련 파일의 전체 경로 · 붙여 넣은 데이터 · 무슨 제품인지 · 코드 규칙
4. **끝의 모습** · 말했다면
5. **이길 상대가 있으면** · 옛 답의 무엇이 싫었는지, 내 말로

반대로 스킬은 내가 말하지 않은 요구를 지어 넣지 않고, 클로드가 생각하는 정답도 적지 않습니다. 100개를 한 방향으로 밀어 버리기 때문입니다.

돌리기 전 이 한 가지 · 두 번에 나눠 보내기

① 먼저 평소 대화로 과제를 정리시킵니다(아직 아레나를 돌리지 않음)

복사해서 그대로 쓰세요

```
아레나에 넘길 과제를 한 덩어리로 정리해 줘. 아직 돌리지는 마.
- 할 일: {한 줄}
- 대상: {누가 읽거나 쓰는지}
- 꼭 지킬 것: {길이 · 형식 · 말투 · 바꾸면 안 되는 것}
- 참고 자료: {파일 전체 경로 또는 붙여 넣은 내용}
- 끝의 모습: {무엇이 나오면 끝인지}
- 지난 답에서 싫었던 점: {내 말로}
빈 칸이 있으면 채우기 전에 나한테 물어봐.
```

② 정리된 글을 그대로 붙여 돌립니다. 다 돈 뒤 `.arena/{실행 폴더}/task.md` 를 열면 실제로 무엇이 넘어갔는지 볼 수 있습니다

복사해서 그대로 쓰세요

```
/arena --quick {①에서 정리된 과제를 그대로 붙여 넣기}
```

영상에서 가린 것 · 쓸 일 고르기

틀리면 곤란한 일에, 고르는 건 내가

잘 맞는 일	README 예시는 가격 페이지 맨 위 한 줄 · 가끔 실패하는 테스트 고치기 · 출시 주간 계획. 공통점은 **답이 여러 갈래이고, 잘못 고르면 손해가 나는 일**입니다(이 자료의 정리)
안 맞는 일	막연한 과제(README: 「막연한 과제를 넣으면 막연한 답 100가지가 나온다」) · 다시 물어도 금방 되는 가벼운 일(평소엔 `--quick`)
최근 사실이 걸린 일	법 · 가격 · 일정처럼 최근에 바뀐 사실이 답을 가른다면, 그 사실의 출처를 과제 파일에 같이 넣습니다. 100개와 심사가 모두 같은 근거를 보고 겨루게 됩니다

결과가 오면 이 다섯 가지를 본다 · SKILL.md 5단계

1. **이긴 답 전문**
2. **이긴 이유**
   버틴 공격 목록과 카드 한 줄(생각하는 방식 + 일하는 순서 + 전략)
3. **라운드**
   예: 「7라운드, 100개 중 1개」
4. **옛 답과의 비교 점수**
   옛 답이 더 높으면 그렇다고 그대로 말하고 둘 다 보여 준다
5. **전체 기록이 있는 실행 폴더**

내 프로젝트 파일을 바꾸는 답이어도 아레나는 **직접 적용하지 않고** 「적용할까요, 고칠까요?」를 묻습니다. 서브 에이전트는 `.arena/` 안에만 씁니다.

복사해서 그대로 쓰세요

이긴 답이 버틴 공격을 하나씩 보여 주고, 그중 사실 확인이 필요한 주장이 있으면 따로 뽑아 줘. 옛 답과의 비교 점수도 같이 보여 줘. 아직 아무 파일도 바꾸지 마.

알아 둘 것 README 원문

100개의 클로드, 정확히 무엇인가

README Fine print	무슨 뜻인가
"100 versions of Claude" means 100 sub-agents of the model you are running.	서로 다른 모델 100개가 아니라 **지금 쓰는 모델**의 서브 에이전트 100개. 서로 다른 건 카드뿐이고, 144개까지는 세 칸 중 두 칸도 겹치지 않게 나눈다
A competitor is its card plus its solution file.	서브 에이전트는 호출 사이를 기억하지 않는다. 3라운드의 a017 은 a017 의 카드와 최신 답을 받은 새 에이전트. 그래서 100개인데 호출은 595번
"The best answer" means the one that survived every match.	심사도 클로드. 가장 강한 답이지 맞다는 증명이 아니다. 그래서 버틴 공격을 보여 주고, 옛 답이 더 높으면 그대로 말한다
100 agents do not all run at the same moment.	10개씩 묶음. 100개 = 70묶음이라 시간이 좀 걸린다
Expect permission prompts unless you allow edits.	기본 모드면 파일마다 승인. accept-edits(Shift+Tab)로 돌린다
Sub-agents cannot see your chat.	과제 파일이 전부다(9쪽)
It never edits your project.	코드 변경은 diff 나 전체 파일로 답 안에 담겨 온다. 적용은 내가 정한다
The same seed gives the same cards and the same bracket.	같은 답까지는 아니다. 모델은 매번 다르게 답한다
It does not fix a bad task.	막연한 과제를 넣으면 막연한 답 100가지가 나온다

막히면

증상 → 할 일

증상	할 일
`/arena` 가 안 보인다	플러그인으로 깔았으면 `/arena-skill:arena`. 짧게 쓰려면 4쪽 길 ②. 깐 뒤 클로드 코드를 다시 연다
승인 창이 끝없이 뜬다	accept-edits 모드(Shift+Tab)로 바꾸고 다시. 클로드는 이 설정을 대신 바꾸지 않는다
파이썬 오류	`python3 --version` 이 3.8 이상인지 본다. 따로 설치할 패키지는 없다
너무 오래 걸린다	`--quick`(16개 · 16묶음)이나 `--agents 8`(10묶음). 동시 실행 한도를 올렸다면 `--wave` 도 같이 올린다
중간에 멈췄다 · 대화가 압축됐다	대진은 파일에 남아 있다. 처음 돌린 그 폴더에서 「아레나 이어서 해 줘」라고 하면 bracket.py status · next 로 이어 간다
결과가 내 요구를 놓쳤다	`.arena/{실행 폴더}/task.md` 를 열어 본다. 빠진 요구를 과제에 넣고 다시 돌린다(9쪽)
옛 답이 더 높게 나왔다	고장이 아니라 기능이다. 옛 답이 더 나았다는 뜻이니 둘을 같이 보고 고른다
같은 seed 인데 답이 다르다	`--seed` 는 카드와 대진표만 고정한다. 답까지 같게 만들지는 못한다
토큰이 걱정된다	`bracket.py plan --agents N` 으로 호출 수부터 본다(8쪽). 평소엔 `--quick`

> 이 자료의 이름 · 명령 · 규칙은 arena-skill 저장소의 README · SKILL.md · rubric.md · strategies.json(2026-10-21 · 커밋 df07b8e)에서 옮겼고, 계산기 출력은 같은 날 직접 실행한 것입니다. 스킬 내용은 바뀔 수 있으니 안 되면 저장소 README 를 먼저 보세요.