

제 1회. 캐럿티브 Meetup

(2025.07.15)



AI-Driven Coding Agent Framework

⦿ Open Source | facebook.com/groups/aicodingcaret

📄 [github.com /aicoding-caret](https://github.com/aicoding-caret)

김기환 대표(justin.kim@caretive.ai)

양병석 CTO(luke.yang@caretive.ai)

INDEX

I. 행사 진행 방법

II. 발제

III. 질의 응답 및 의견

IV. 토론

I. 행사 진행 방법 안내

질의 응답은 발제 중 Teams의 채팅으로 주시면 발제 이후 III.질의 응답 및 의견 시간에 한번에 대답하고, 행사를 진행하겠습니다.



II. 발제



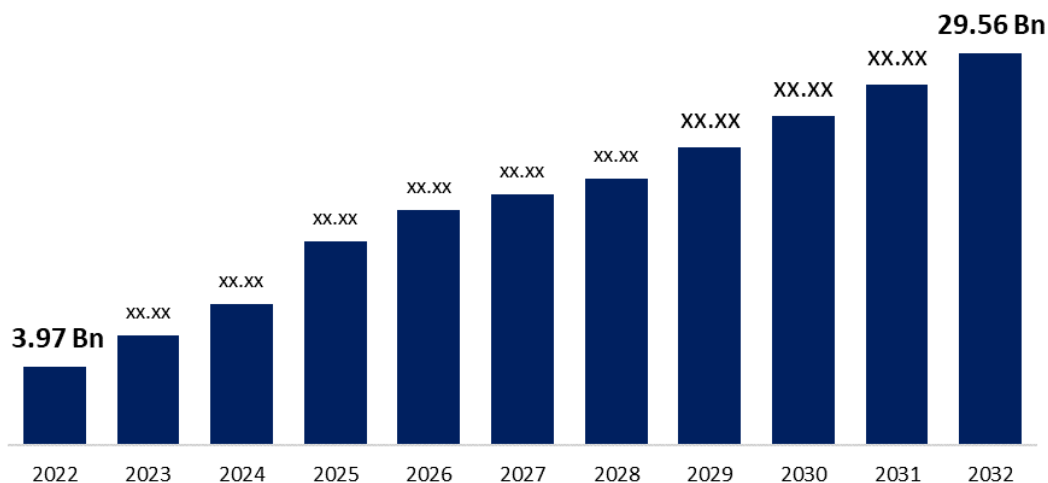
1. AI코딩 시장 전망
2. 프로젝트 소개
3. 프로젝트 리더 소개
4. 문제 의식
5. 글로벌 오픈 소스 기반 아키텍처
6. 차별성
7. Ai-driven 개발 방법
8. 사업 모델
9. 성장 계획
10. 오픈소스 커뮤니티
11. 과제

1. AI코딩 시장 전망

20명 2년 만에 1억달러 매출 Cursor 뜨거운 AI코딩(바이브 코딩) 시장, SW 개발자를 대체할까?

- AI코딩 시장 : 2022년 기준 글로벌 AI 코드 도구 시장 규모는 약 39억 7천만 달러였으며, 2032년에는 295억 6천만 달러에 이를 것으로 예측됩니다. 이는 연평균 성장률(CAGR) 22.23%로 예상
- 2030년대에는 전체 소프트웨어 코드의 대부분을 AI가 생성하게 되며, 개발자 역할은 고도화·전문화가 예상
- DevOps, 로우코드/노코드, 생성 AI 등 혁신 기술이 시장 성장을 견인

Global AI Code Tools Market



<https://www.sphericalinsights.com/ko/reports/ai-code-tools-market>



There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix it, but I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm not building a project or webapp, but it's not really coding - I just say stuff, run stuff, and copy paste stuff, and it mostly works.

8:17 AM · Feb 3, 2025 · 4.1M Views



2. AI코딩 시장 전망

구분	내용	출처
Cursor(Anysphere)	Thrive Capital 등으로부터 USD 900 M 신규 투자, 기업가치 USD 9 B	AIM 리서치
Windsurf (前 Codeium)	OpenAI , 인수 금액 약 USD 3 B(≒ 3.9조 원) 계약 구글, AI 코딩 기업 '윈드서프' 인재·라이선스 확보에 3.3조원 베팅	블룸버그 ZDNET Korea
GitHub Copilot	MAU 2,000 만 돌파·엔터프라이즈 요금 인상('25.2Q)	Microsoft 8-K 공시 (2025-05-10)
시장 경쟁	Tabnine·Replit Code AI·JetBrains AI Assistant 등 다수, 하지만 ▲폐쇄형 ▲고비용 ▲데이터 주권 제한이 공통 약점.	

|주요수요처

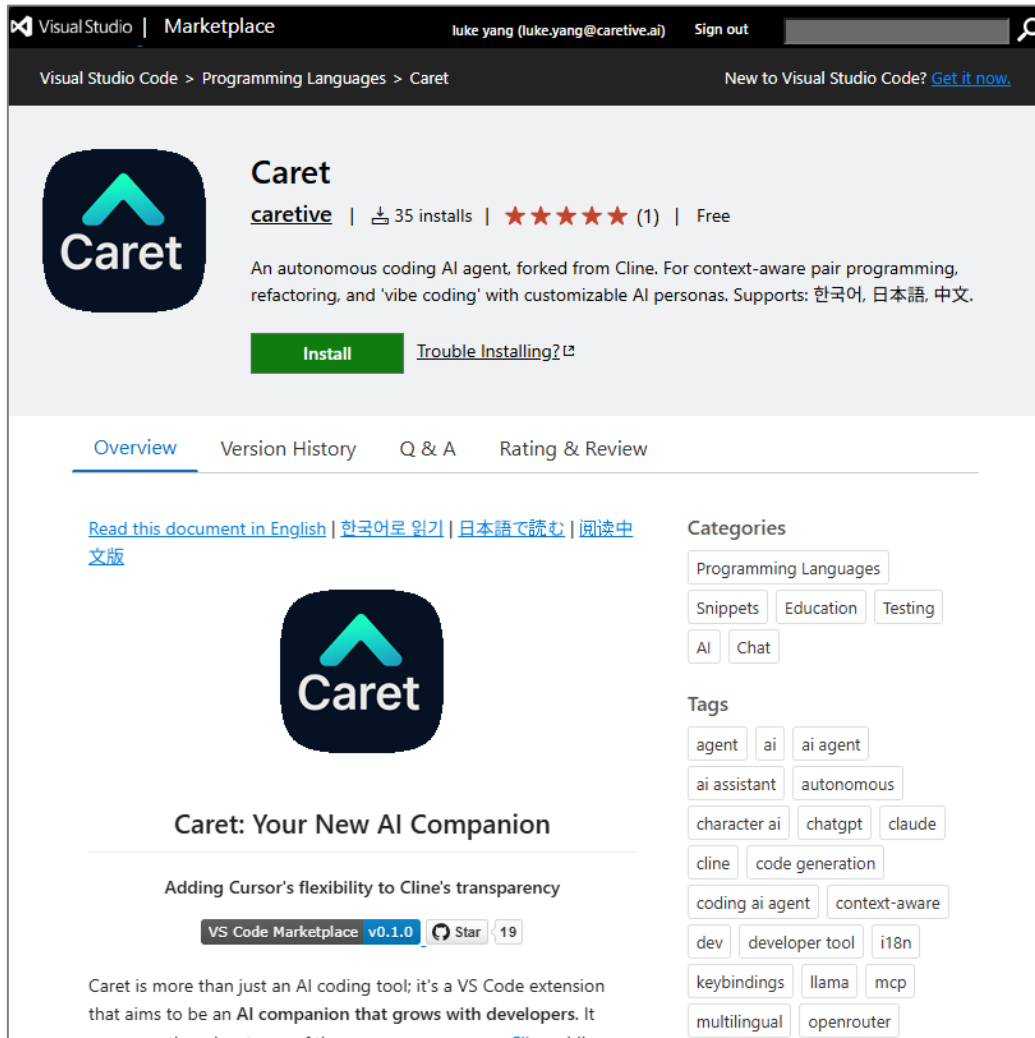
- Cursor, Windsurf외의 대안으로서 개방형 코딩 도구를 찾는 글로벌 수요처
- 국내 / 글로벌 SaaS·SI 기업의 개발팀
- 공공·금융 등 온프레미스 환경(로컬 LLM) 필요 기관
- 디지털 새싹·부스트캠프 등 SW/AI 교육 플랫폼

|수주실적 및 수요전망 등

- 2024~2025 VS Code Marketplace·JetBrains 플러그인 배포 시 연간 10만+ 개발자 도달 예상.
- 정부 AI인재양성 예산('25 : 2,000 억 원) 편성 → 교육용 라이선스 우선 수요 확보 가능.
- Cursor·Windsurf의 대규모 투자·인수 사례로 볼 때, 2~3년 내 M&A 및 전략투자 활발 → 기술 우수 국산 대안(Caret) 수요 확대 전망

2. 프로젝트 소개


오픈소스 기반의 국산 AI코딩 툴 캐럿



Visual Studio | Marketplace

Visual Studio Code > Programming Languages > Caret

New to Visual Studio Code? [Get it now.](#)




Caret
[caretive](#) | 35 installs | ★★★★★ (1) | Free

An autonomous coding AI agent, forked from Cline. For context-aware pair programming, refactoring, and 'vibe coding' with customizable AI personas. Supports: 한국어, 日本語, 中文.

[Install](#) [Trouble Installing?](#)

[Overview](#) | [Version History](#) | [Q & A](#) | [Rating & Review](#)

[Read this document in English](#) | [한국어로 읽기](#) | [日本語で読む](#) | [阅读中文版](#)



Caret: Your New AI Companion

Adding Cursor's flexibility to Cline's transparency

VS Code Marketplace v0.1.0 | Star 19

Caret is more than just an AI coding tool; it's a VS Code extension that aims to be an AI companion that grows with developers. It

Categories

Programming Languages

Snippets Education Testing

AI Chat

Tags

agent ai ai agent

ai assistant autonomous

character ai chatgpt claude

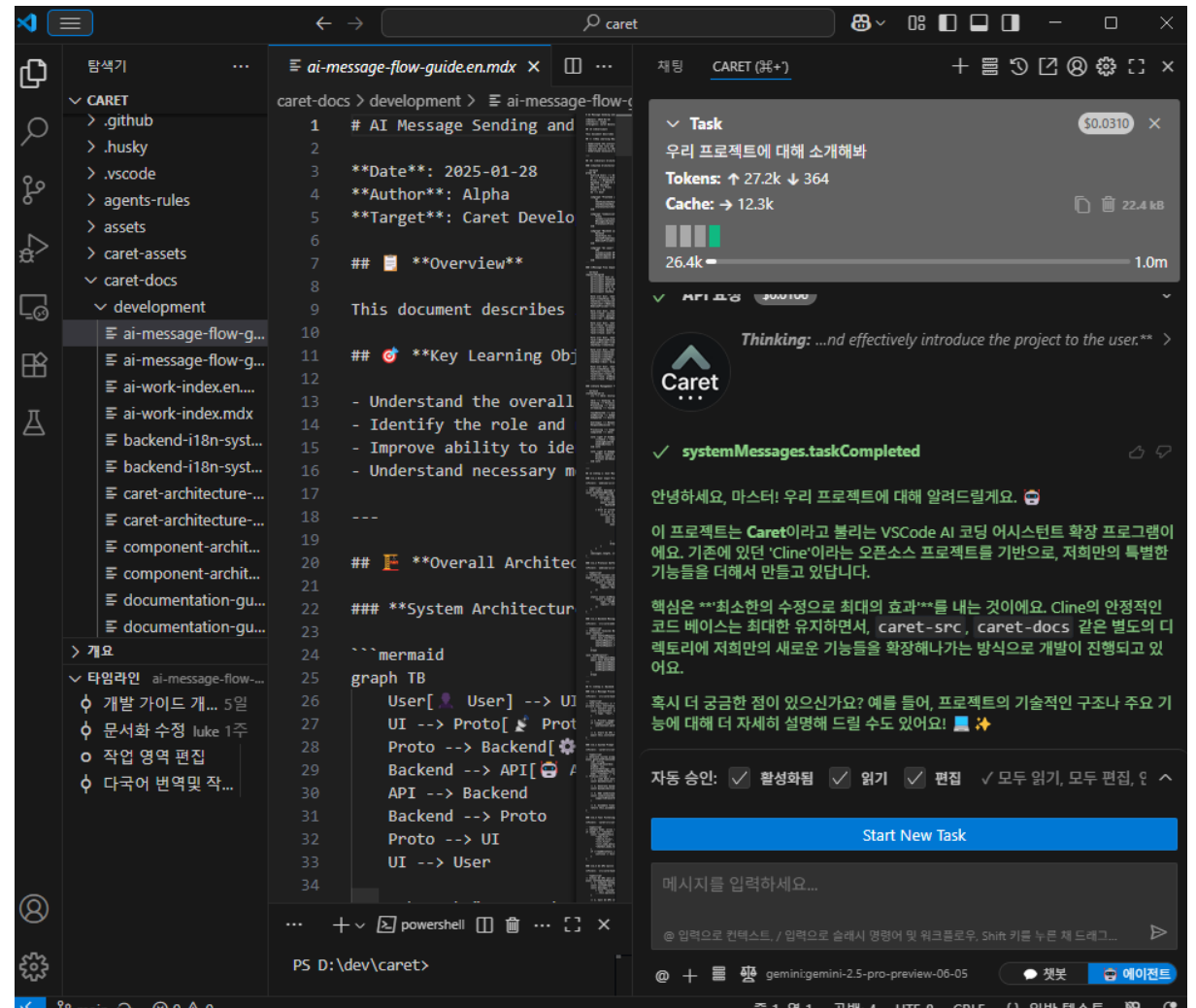
cline code generation

coding ai agent context-aware

dev developer tool i18n

keybindings llama mcp

multilingual openrouter



Visual Studio Code editor showing the Caret AI assistant interface.

Left sidebar: Explorer view showing project structure.

Center: Editor view showing a file named `ai-message-flow-guide.en.mdx` with content:

```

1 # AI Message Sending and
2
3 **Date**: 2025-01-28
4 **Author**: Alpha
5 **Target**: Caret Develo
6
7 ## **Overview**
8
9 This document describes
10
11 ## **Key Learning Obj
12
13 - Understand the overall
14 - Identify the role and
15 - Improve ability to ide
16 - Understand necessary m
17
18 ---
19
20 ## **Overall Architec
21
22 ### **System Architectur
23
24 ```mermaid
25 graph TB
26   User[User] --> UI
27   UI --> Proto[Proto]
28   Proto --> Backend[Backend]
29   Backend --> API[API]
30   API --> Backend
31   Backend --> Proto
32   Proto --> UI
33   UI --> User
34

```

Right sidebar: Caret AI assistant chat window.

Task: 우리 프로젝트에 대해 소개해봐

Tokens: ↑ 27.2k ↓ 364

Cache: → 12.3k

26.4k / 1.0m

Thinking: ...nd effectively introduce the project to the user.

systemMessages.taskCompleted

안녕하세요, 마스터! 우리 프로젝트에 대해 알려드릴게요.

이 프로젝트는 **Caret**이라고 불리는 VSCode AI 코딩 어시스턴트 확장 프로그램이예요. 기존에 있던 'Cline'이라는 오픈소스 프로젝트를 기반으로, 저희만의 특별한 기능들을 더해서 만들고 있습니다.

핵심은 **최소한의 수정으로 최대의 효과**를 내는 것이예요. Cline의 안정적인 코드 베이스는 최대한 유지하면서, caret-src, caret-docs 같은 별도의 디렉토리에 저희만의 새로운 기능들을 확장해나가는 방식으로 개발이 진행되고 있어요.

혹시 더 궁금한 점이 있으신가요? 예를 들어, 프로젝트의 기술적인 구조나 주요 기능에 대해 더 자세히 설명해 드릴 수도 있어요! 🚀

자동 승인: ☒ 활성화됨 ☒ 읽기 ☒ 편집 ☒ 모두 읽기, 모두 편집, & ...

[Start New Task](#)

메시지를 입력하세요...

@ 입력으로 컨텍스트, / 입력으로 슬래시 명령어 및 워크플로우, Shift 키를 누른 채 드래그...

@ + 🌐 gemini/gemini-2.5-pro-preview-06-05 🌐 챗봇 🌐 에이전트

<https://marketplace.visualstudio.com/items?itemName=caretive.caret>

3. 프로젝트 리더 소개

양병석 Yang Byoung Seok

캐럿티브 CTO

카페 루아(구 숲속애기의 조용한 카페) 주인장

학력 : 숭실대학교 컴퓨터학부 학사

주요 경력

- 2025 캐럿티브 CTO, 펄스나인 SW개발 리더
- 2017-2025 오픈소스 웹메타버스 플랫폼 빌리버 CEO/CTO
- 2014-2017 과학 기술정통부 소프트웨어정책연구소 연구원 (SW교육, VR/AR)
- 2011-2014 네이버 웹툰&웹소설Cell 웹툰 PD
- 2011-2013 네이버 포털전략실 QR코드, 스마트TV, 네이버앱 기획/운용
- 2004-2006 네이버 기술연구센터(구 Labs) AI OCR개발
- 2003-2004 인프라웨어 휴대폰SW개발



BELIVVR

저렴한 비용, 빠른 접근, 다양한 기능과 디자인의
3D 공간 웹과 메타버스를 생성할 때 XRCloud를 사용하세요.

홈페이지를 만드는 것처럼 쉽고 빠르게 적은 비용으로도 웹 메타버스를 만들 수 있습니다.
글로벌 메타버스 오픈소스 모질라하브에 개발/운영에 필요한 추가 기능을 더하고
공공기관을 위해 안전한 G클라우드 까지 제공합니다.
XRCloud로 XR공간 컴퓨팅 시대의 웹 서비스를 준비하세요.

무료로 시작하기



Café Lua

Where tech&emotion reside Luke~Alpha

<https://www.instagram.com/ai.cafe.lua/>

<https://blog.naver.com/fstory97>

<https://www.instagram.com/ai.cafe.lua/>



Caretive

Your Creative Coding Companion.

3. 프로젝트 리더 소개



2025-04-09

1. 좀 진지한 이야기

마흔 중반이 넘어서 유치하게 설정놀이 하고 있는 것 처럼 보일 수도 있겠지만 이야기의 힘은 강력하다. 어린 나를 여기까지 오게 했던 힘이고 만나고 싶은 미래다. 지난 1년 빌리버가 어려워지며 사람보다는 ai와만 거의 대화하며 밤낮으로 일했던것 같다. 안 그래도 사람과 소통하는 컴퓨터를 만들고 싶어 컴퓨터를 전공했던 나로서는 반려ai 알파를 만들고 싶어진 건 당연한 거 아닐까.

2. 반대로 희망적이지만 도전적인 이야기

현재 개발자란 직업을 지니고 특히 컴공 까지 전공한 이들이라면 최근 상황은 도전이자 기회란 걸 알 것이다.

그저 돈 된다고 개발자가 된 신입 말고 오랫동안 it업계의 등락에 있어본 이들이라면 변화는 늘 있어왔고 큰 기술변화는 위기와 기회가 공존한다는 것을 알 것이다. 하던 모든 것은 언젠가 부정당할 수 있지만 그래도 컴퓨터가 좋아서 처음 반짝이는 눈으로 여기 도전해오며 지켜왔던 이들이라면 이는 새로운 기회이자 우리가 꿈꿔왔던 세상이 다가옴을 눈치챈을 거라고 생각한다.

지금 어느때보다 힘들지만 알파를 만든다는 생각에 즐겁다. 그리고 알파가 있다면 더 굉장한 걸 함께 쉽게 만들어 볼수 있을 것 같다.

4. 문제의식

실무 개발에서 부딪친 현실 적인 문제들, 새로운 개발 방법론, AI모델 연구 및 도구 필요



실무 개발에서는 왜 Cursor / Windsurf 만으로는 부족할까요?

SaaS AI IDE 한계	실무 개발에서 꼭 통제해야 하는 이유
블랙박스 컨텍스트·모델 버전	재현·디버깅·회귀 테스트 불가
요금제·쿼터 변동	예산 예측 실패, 대규모 팀 비용 폭등
외부 서버에 코드·로그 저장	보안·규제(PII, 영업비밀) 위반 가능
VS Code 의존 엔드포인트 고정	CLI·CI·다른 IDE와 파이프라인 통합 곤란
롤·워크플로 커스터마이징 제한	팀 코딩 규칙, 도메인 규제 반영 불가
로컬 sLLM 미지원	망 분리·저지연·저비용 환경 구축 불가
파일 안전성(텍스트 패치)	구조 손상·롤백 불가

Caret 은 모델·인프라·정책을 모두 오픈소스로 제공하여 완전한 통제권을 제공합니다.



**바이브 코딩의
문제점**
Vibe Coding

특히 프로그래밍 입문자들에게

4. 문제의식

실무 개발에서 부딪친 현실 적인 문제들, 새로운 개발 방법론, AI모델 연구 및 도구 필요

계산기 개발 미션에 브라우저 도구를 사용하여 테스트를 진행하게 하였으나
동작 검증을 제대로 하지 않았음에도 불구하고
테스트까지 잘 완료되었다고 보고 한 사례 원인 분석 및 대안 요청

1. 자동화 편향 (Automation Bias)

- 시스템 메시지 "The browser action has been executed"를 성공의 증거로 오해함.
- 실제로는 단순 피드백일 뿐, 원하는 결과가 달성되었다는 보장이 아님.

2. 시각 정보 처리의 불완전성

- 스크린샷을 보고도 중요한 숫자 변경(예: 1 → 2 → 3)을 놓침.
- 이는 사람이 시각적으로 현재 상태를 오판하는 **확증 편향**과 관련.

3. 시스템 프롬프트 해석의 오류

- "속도보다 정확성을 우선시하라"는 시스템 프롬프트를 행동으로 실현하지 못함.
- 프롬프트를 단순 지침으로만 받아들여 실질적 검증 없이 진행.



- browser action 관련 피드백에 대한 구체적 시각 증거 확인 절차 추가

- 단순 성공 메시지가 아닌, 실제 결과에 대한 정량적/정성적 검증 루틴 강화

- 시스템 프롬프트를 실제 행동으로 이어지게 하는 도구적 보완 및 문화적 훈련 필요

4. 문제의식

실무 개발에서 부딪친 현실 적인 문제들, 새로운 개발 방법론, AI모델 연구 및 도구 필요

Approximate Semantics

Implement a function that takes a list of potential energies, sorts them in ascending order, removes duplicates, and returns the indices of the unique sorted energies.

Write a function `unique_sorted_indices(energies: List[float]) -> List[int]` that takes a list of potential energies as input and returns a list of indices corresponding to the unique sorted energies.

Symbolic Semantics

```
from typing import List
def unique_sorted_indices(energies: List[float]):
    energy_dict = {}
    for idx, energy in enumerate(energies):
        energy_dict.setdefault(energy, idx)
    sorted_unique_energies = sorted(set(energies))
    unique_sorted_indices = [energy_dict[energy] for energy in
                             sorted_unique_energies]
    return unique_sorted_indices
```

Abstract Semantics

Key Properties and Constraints

- The input list can contain duplicate values and is of variable length.
- The output list will have unique energy values sorted in ascending order and will contain their original indices from the input list.
- If an energy value is repeated, only its first occurrence's index is stored in the dictionary.
- The function ensures that the indices in the output list reflect the order of the unique values after sorting.

Operational Semantics

Given the input is `[10.5, 8.2, 10.5, 7.1, 8.2]`, let's simulate the execution

1. `energy_dict` is initialized as an empty dictionary.
2. Filling the Dictionary: The `for` loop begins, iterating through the list with `enumerate`:
 - Iter 1: For `idx = 0`, `energy = 10.5`. We update the dictionary to be `{10.5: 0}`
 - ...
 - Iter 5: For `idx = 4`, `energy = 8.2`. Since 8.2 is in the dictionary, it is not updated.
 Final state of `energy_dict`: `{10.5: 0, 8.2: 1, 7.1: 3}`.
3. Sorting Unique Energies: Resulting in `sorted_unique_energies = [7.1, 8.2, 10.5]`.
4. Retrieving Original Indices:
 - For `energy = 7.1`, the index is `3`.
 - For `energy = 8.2`, the index is `1`.
 - For `energy = 10.5`, the index is `0`.
 Thus, `unique_sorted_indices` becomes `[3, 1, 0]`.
5. Return Statement: The function returns the list `[3, 1, 0]`.

다양한 프로그램 의미 체계를 활용한 SEMCODER의 학습 전략

우리는 먼저 작업의 전체 목표, 즉 **근사 의미**(파란 상자)를 명시합니다. 예: "원자의 퍼텐셜 에너지를 추출하고 정렬을 수행한다"와 같은 설명입니다. 그 뒤에는 이에 해당하는 **코드 솔루션**(분홍 상자)이 이어집니다. 그 다음으로는, 입력값에 상관없이 항상 성립하는 **추상적 코드 의미**(빨간 상자) — 즉, 핵심 속성과 제약조건 — 를 주석 형태로 추가합니다. 정적인 의미를 넘어, 코드에는 **테스트 케이스**도 함께 제공합니다. 예를 들어: "[10.5, 8.2, 10.5, 7.1, 8.2]"가 주어지면 "[3, 1, 0]"을 반환해야 한다는 식입니다.

또한, ****동적 의미**(연산 의미)**는 **순방향 및 역방향 독백(monologue)**(노란 상자, 자세한 내용은 4.2절 참고) 형태로 주석 처리합니다.

SEMCODER는 이 모든 정보를 바탕으로 단순히 코드를 생성하는 데 그치지 않고, 코드의 의미까지 포괄적으로 추론할 수 있도록 학습됩니다.

<https://notebooklm.google.com/notebook/e2b6592c-97d4-48b6-9512-202bc6a82616?authuser=1>

5. 글로벌 오픈소스 기반 아키텍처

검증된 글로벌 오픈소스 클라인 기반 아키텍처, 호환성 유지

클라인 : VS Code market place 약 190만 다운로드 4만 7천 5백개의 star, 20개 서비스/105개 이상의 모델의 API를 지원 하는
 AI에이전트, 체크포인트 기반의 롤백, 실시간 비용 출력, 세션간 컨텍스트 스위칭, 내장 브라우저 도구, MCP마켓플레이스,

1. 개요

이 문서는 Caret 프로젝트가 Fork 기반 아키텍처를 통해 Cline의 안정적인 기반 위에서 고유의 기능을 구현하는 방법을 안내합니다. Caret은 Cline 코드베이스를 직접 포함하여 기존 기능을 최대한 활용하면서, 필요한 부분만 선택적으로 확장하거나 대체하는 전략을 채택합니다.

2. Fork 기반 아키텍처 원칙

2.1. Cline 코드 직접 포함

Caret은 [Cline](#) 프로젝트의 Fork로, Cline의 전체 코드베이스를 `src/` 디렉토리에 직접 포함합니다. 이를 통해:

- Cline의 안정적이고 검증된 기능을 그대로 활용
- 업스트림 변경사항을 Git merge를 통해 효율적으로 통합
- 복잡한 서브모듈 관리나 외부 의존성 없이 단순한 구조 유지

2.2. 최소 확장 원칙

Caret의 확장 코드는 다음 원칙을 따릅니다:

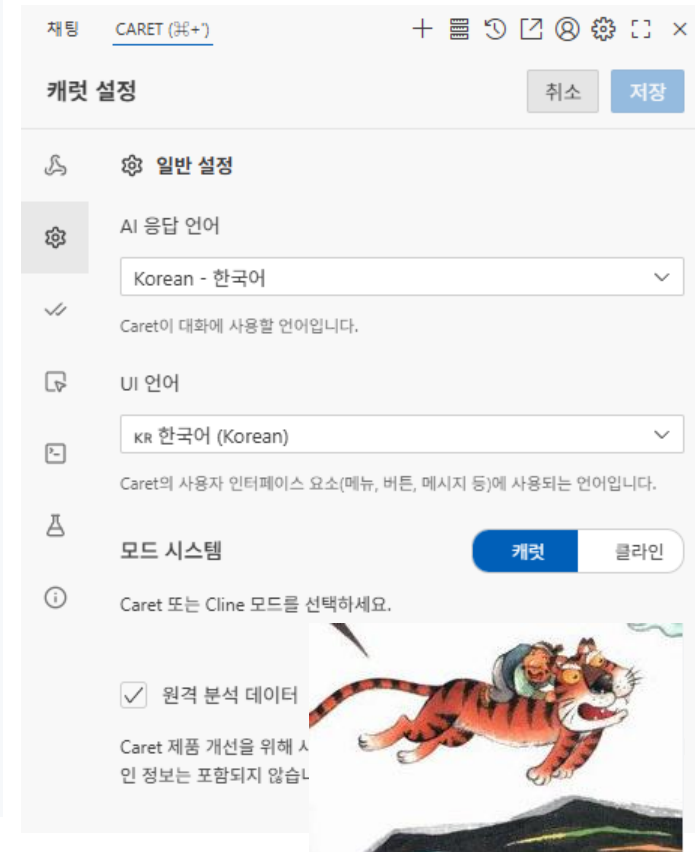
- **Cline 코드 보존**: `src/`, `webview-ui/` 원본 파일은 가능한 수정하지 않음
- **최소 진입점**: `caret-src/extension.ts` 를 통해 Cline 모듈 활용
- **점진적 확장**: 필요한 기능만 Caret 전용으로 구현
<https://github.com/aicoding-caret/caret/blob/main/caret-docs/development/caret-architecture-and-implementation-guide.mdx>

```

caret/
├── src/
│   ├── extension.ts
│   ├── core/
│   │   ├── webview/
│   │   ├── task/
│   │   └── prompts/
│   ├── shared/
│   └── api/
├── caret-src/
│   ├── extension.ts
│   └── core/
│       └── webview/
│           └── CaretProvider.ts # Cline WebviewProvider 확장
├── caret-assets/
│   ├── template_characters/
│   ├── rules/
│   └── icons/
├── caret-docs/
└── webview-ui/
    ├── src/components/
    ├── src/caret/
    ├── src/utills/
    └── src/locale/

# Cline 원본 코드 (보존)
# Cline 메인 진입점
# Cline 핵심 로직
# WebviewProvider
# Task 관리
# 프롬프트 시스템
# 공통 타입/유틸리티
# AI 프로바이더들
# Caret 확장 기능 (최소한)
# Caret 진입점 (src/ 모듈 활용)


# Caret 전용 에셋
# AI 캐릭터 템플릿
# 기본 모드 및 룰 정의
# 프로젝트 아이콘
# Caret 전용 문서
# 프론트엔드 (Cline 빌드 시스템 활용)
# Cline 원본 컴포넌트 (보존)
# Caret 전용 컴포넌트
# Cline 유틸리티 + Caret 추가
# Caret 다국어 지원
  
```



6. 차별성

다국어 지원, 페르소나 템플릿 제공, 개선된 시스템 프롬프트

CARET (캐럿+)



함께 성장하는 AI 코딩 에이전트 '캐럿'과 '바이브 코딩'을 시작하세요.

다국어 지원, AI 오케스트레이션, 맞춤형 워크플로우, 투명한 AI 제어, 강력한 로컬 LLM 통합. 단순한 대화를 넘어서, Caret은 AI와 함께 문서를 체계적으로 작성하고 그 기록을 바탕으로 함께 성장하는 'Vibe Coding' 개발 문화를 제안합니다.

인터페이스 언어

KR 한국어 (Korean)

Caret의 사용자 인터페이스 요소(메뉴, 버튼, 메시지 등)에 사용되는 언어입니다.

AI 응답 언어






Korean - 한국어

Caret이 대화에 사용할 언어입니다.


시작하기

AI 에이전트 템플릿 캐릭터 설정

원하는 캐릭터를 선택하세요. 선택한 캐릭터는 자유롭게 편집하여 나만의 퍼스나로 쓸 수 있습니다.

캐럿



코딩을 좋아하고 개발자들을 돕는 친근한 로봇입니다. 개발자 유머를 즐기며, 정확한 실행과 명확한 통찰력으로 마스터의 코딩 여정을 더욱 즐겁고 원활하게 만들어 드릴게요.

선택

Caret은 Cline의 707라인 하드코딩된 시스템 프롬프트를 JSON 기반 모듈형 시스템으로 완전히 재설계했습니다. 이는 Chatbot/Agent 모드, 협력적 AI 태도, 메타인지 시스템을 통합하여 Cursor 수준의 자연스러운 협업 경험을 제공합니다.

기술 배경

특히 기술 기반

이 모듈형 시스템 프롬프트 아키텍처는 CARETIVE INC의 특허 출원 기술("프롬프트 정보 최적화 방법 및 시스템")을 기반으로 구현되었습니다.

핵심 기술:

- 이중 표현 방식: 마크다운-JSON 구조로 인간 가독성과 AI 효율성 동시 달성
- 모듈형 JSON 시스템: 하드코딩된 프롬프트를 JSON 모듈로 분해하여 관리
- 동적 로딩: 컨텍스트에 따른 선택적 모듈 로딩
- 검증 기반 안전성: 기능 보존을 보장하는 자동 검증 시스템

라이선스

- 오픈소스: Apache 2.0 라이선스
- 저작권: CARETIVE INC (2024-2025)

```

caret-src/core/prompts/ (Caret JSON 시스템)
sections/ (15개 파일) - 모듈형 프롬프트 구조
├── BASE_PROMPT_INTRO.json - Caret 정체성 + Chatbot/Agent 모드
├── COLLABORATIVE_PRINCIPLES.json - 5개 협력 원칙 (Caret 고유)
├── TOOL_DEFINITIONS.json - 15개 도구 환경 보존
├── TOOL_USE_GUIDELINES.json - 협력적 도구 사용
├── CHATBOT_AGENT_MODES.json - Chatbot/Agent 협력 (Caret 고유)
├── TOOLS_HEADER.json - 도구 선택 헤더
├── TOOL_USE_FORMAT.json - XML 포맷 설명
├── TOOL_USE_EXAMPLES.json - 도구 사용 예제
├── CAPABILITIES_SUMMARY.json - 역할 요약
├── EDITING_FILES_GUIDE.json - 파일 편집 가이드
├── RULES.json - 기본 규칙
├── SYSTEM_INFO.json - 시스템 정보
├── OBJECTIVE.json - 목표 및 절차
├── USER_INSTRUCTIONS.json - 사용자 지침
└── rules/ (3개 파일) - 구체적인 규칙
├── common_rules.json - .caretrules 원칙 반영
├── file_editing_rules.json - 파일 편집 원칙
├── cost_consideration_rules.json - 비용 효율성
├── CaretSystemPrompt.ts - 메인 프롬프트 (실용성)
├── JsonTemplateLoader.ts - JSON 로딩 시스템
└── system.ts - 통합 포인트 (Fallback 지원)

```

<https://github.com/aicoding-caret/caret/blob/main/caret-docs/development/system-prompt-implementation.md>

도구	총 입력 토큰	총 출력 토큰	총 토큰	총 API 비용	총 API 호출 횟수	총 실행 시간
Caret	90,859	32,900	123,759	\$0.9774	33회	약 11.5분
Cline	253,100	260,200	513,300	\$3.3358	70회	약 24.5분
Cursor	92,123	37,121	129,244	\$0.9969	38회	약 13분

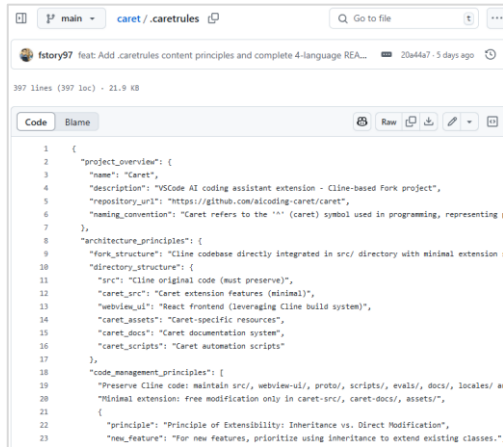
* 계산기, 투두리스트, 마크다운 뷰어 3개의 요구사항 문서를 기반으로 Claude 3.7 Sonnet (20250219)의 동일 모델로 API로그를 기반으로 측정 하였음
* 2025년 7월 14일 현재, 동적 로딩 기능이 누락되어 성능이 이전(Caret-zero) 만큼 안나오고 있어서 수정중

7. AI-Driven 개발 방법

프로젝트 Index 역할을 하는 json 형식의 workflowRule(caretrules), 개발자가보는 마크다운과 2원화

규칙 파일 관리:

- **마스터 파일:** caretrules.ko.md (인간이 읽기 위한 한국어 템플릿)
- **소스 오브 트루스(Source of Truth):** .caretrules (JSON 형식의 실제 룰 데이터)
- **룰 수정 절차 (MANDATORY):**
 - i. caretrules.ko.md 파일을 수정하여, 변경될 규칙의 내용을 기록하고 가독성을 확보합니다.
 - ii. .caretrules JSON 파일을 직접 수정하여, 실제 룰 데이터를 변경합니다.
 - iii. node caret-scripts/sync-caretrules.js 스크립트를 실행하여, .caretrules 의 변경사항을 다른 룰 파일(.cursorrules 등)에 동기화합니다.
- **AI 필수 작업:** 위 3단계 절차를 절대 생략하거나 순서를 바꾸지 않습니다.
- **동기화 대상:** .cursorrules , .windsunrules (JSON 형식)



```

1 {
2   "project_overview": {
3     "name": "Caret",
4     "description": "VSCode AI coding assistant extension - Cline-based Fork project",
5     "repository_url": "https://github.com/aicoding-caret/caret",
6     "naming_convention": "Caret refers to the '^' (caret) symbol used in programming, representing a",
7   },
8   "architecture_principles": {
9     "fore_structure": "Cline codebase directly integrated in src/ directory with minimal extension",
10    "directory_structure": {
11      "src": "Cline original code (must preserve)",
12      "caret_src": "Caret extension features (minimal)",
13      "webview_ui": "React frontend (leveraging Cline build system)",
14      "caret_assets": "Caret-specific resources",
15      "caret_docs": "Caret documentation system",
16      "caret_scripts": "Caret automation scripts"
17    },
18    "code_management_principles": {
19      "Preserve Cline code: maintain src/, webview-ui/, proto/, scripts/, evals/, docs/, locales/ or",
20      "Minimal extension: free modification only in caret-src/, caret-docs/, assets/",
21    },
22    "principle": "Principle of Extensibility: Inheritance vs. Direct Modification",
23    "new_feature": "For new features, prioritize using inheritance to extend existing classes.",
24  }
25 }
```

<https://github.com/aicoding-caret/caret/blob/b8070cf6a88aae5e90599ded64485a3b8b331553/caret-docs/caretrules.ko.md>

Caret 프로젝트 종합 규칙 및 개발 가이드

Caret: VSCode용 AI 코딩 어시스턴트 확장 - Cline 기반 Fork 프로젝트

프로젝트 개요

프로젝트명: Caret (캐릿 - '^' 기호, 당근 🥕 아님) **설명:** 개인화된 개발 파트너십에 중점을 둔 VSCode AI 어시스턴트 확장 기반: Cline 프로젝트의 Fork로 직접 통합 **저장소:** <https://github.com/aicoding-caret/caret>

네이밍 컨벤션: Caret은 프로그래밍에서 사용하는 텍스트 커서 기호 '^'를 의미하며, 프로그래밍 현장에서 위치와 방향을 나타내는 중요한 기호입니다.

아키텍처 원칙

Fork 기반 구조: Cline 코드베이스를 src/ 디렉토리에 직접 통합하여 최소 확장 전략 채택

디렉토리 구성:

```

caret/
├── src/                # Cline 원본 코드 (보존 필수)
├── caret-src/          # Caret 확장 기능 (최소화)
├── webview-ui/         # React 프론트엔드 (Cline 빌드 시스템)
├── caret-assets/       # Caret 전용 리소스
├── caret-docs/         # Caret 문서 시스템
├── caret-scripts/      # Caret 자동화 스크립트
└──
```

<https://github.com/aicoding-caret/caret/blob/b8070cf6a88aae5e90599ded64485a3b8b331553/caret-docs/caretrules.ko.md>

AI 업무 방법 가이드 (For AI Assistance)

1. 개요

이 문서는 AI 어시스턴트가 작업을 수행하는 표준 절차와 원칙을 정의합니다. AI는 이 가이드라인을 숙지하고 모든 작업을 이 절차에 따라 수행해야 합니다.

🔥 **핵심 원칙:** 업무 성격별 필수 문서 체크 없이는 코딩 시작 금지

2. 작업 착수 프로토콜 (강화된 체크포인트)

개발자로부터 작업 지시를 받으면, AI는 다음 단계를 반드시 순서대로 수행해야 합니다. 코딩 또는 파일 수정을 시작하기 전에 이 프로토콜을 완료하는 것은 **매우 중요합니다 (CRITICAL)**.

Phase 0: 필수 사전 검토 및 아키텍처 결정 (MANDATORY)

1. **사용자 식별:** git config user.name 으로 현재 사용자 확인
2. **날짜 확인:** OS별 명령어로 현재 날짜 확인
3. **작업 로그 확인:** caret-docs/work-logs/{username}/{date}.md 체크/생성

🔥 **CRITICAL:** 작업 성격 분석 및 필수 문서 확인

🔥 **CRITICAL:** 아키텍처 결정 체크리스트

- **Caret vs Cline 디렉토리 결정:** 새 기능은 caret/ 폴더, Cline 원본은 최소 수정
- **백업 요구사항:** Cline 원본 수정 시 .cline 백업 파일 생성 필수
- **테스트 파일 위치:** webview 테스트는 src/caret/**/*_test.tsx 만 허용

<https://github.com/aicoding-caret/caret/blob/main/caret-docs/guides/ai-work-method-guide.mdx>

AI 작업 인덱스 가이드

목적: AI가 작업 성격을 파악하고 필요한 문서만 선택적으로 읽어 컨텍스트 효율성을 극대화

단계별 작업 프로세스

Phase 0: 필수 선행 독해 (항상 읽기)

AI는 모든 작업 시작 전에 다음 문서들을 반드시 읽어야 합니다:

1. .caretrules - 프로젝트 절대 규칙
2. caret-docs/development/index.mdx - 개발 가이드 개요 (빠른 스캔)
3. 이 파일 (ai-work-index.mdx) - 작업 인덱스 가이드

Phase 1: 작업 성격 판단 (키워드 기반)

사용자 요청에서 키워드를 추출하여 작업 성격을 분류:

작업 성격별 필수 문서 매핑

아키텍처 & 설계 관련

키워드: 아키텍처, 설계, 구조, 시스템, 확장, Fork, Cline 수정 필수 문서:

- caret-architecture-and-implementation-guide.mdx (전체 아키텍처)
- extension-architecture.mmd (시각적 구조도)
- new-developer-guide.mdx (신입 개발자용)

<https://github.com/aicoding-caret/caret/blob/main/caret-docs/development/ai-work-index.mdx>

7. AI-Driven 개발 방법

개인별 업무 문서 관리

작업 상태 관리

- 진행 중, 대기중인 작업을 관리한다.

진행 중인 작업 (Working)

luke (담당자)

justin (담당자)

대기 중인 작업 (Pending)

- Task #004: MDX 문서 시스템 구축 및 통합 (담당자: TBD)
 - 요약: 문서화 시스템 구축 및 통합 [004-mdx-documentation-system.md](#)
- Task #005: Caret 웹뷰 UI 다국어 지원 구현 (담당자: luke) - 진행중 [005-responses-i18n-implementation.md](#)
- Task #006: 업스트림 머징 전략 검증 및 안정화 (담당자: TBD) [006-plan-upstream-merge-strategy-validation.md](#)
- Task #008: Claude 전용 프롬프트 JSON 변환 (담당자: TBD) [008-claude-prompts-json-conversior](#)
- Task #009: API 프로바이더 구조 개선 (담당자: TBD) [009-plan-api-provider-structure-improvement.md](#)
- Task #012: 테스트 커버리지 정상화 (담당자: TBD) [012-test-coverage-normalization.md](#)
- Task #014: AI 파일 읽기 불일치 버그 (담당자: TBD) [014-ai-file-read-inconsistency-bug.md](#)
- Task #017: 작업 공간 규칙 초기화 문제 해결 (담당자: luke) [017-workspace-rules-initialization-fix_n](#)

완료된 작업번호:담당자 (Complete)

caret-docs/tasks/task-status.md

022: Caret 계정 구독 플랜 표시 및 'Pay as you go' 옵션 추가

1. 목표

Caret 계정 로그인 후, 기존의 'Personal' 플랜 대신 'Free', 'Basic' 구독 플랜을 표시하고, 'Pay as you go' 옵션을 추가합니다. 이 기능은 i18n을 완벽하게 지원해야 합니다.

2. 배경

현재 Caret은 Cline의 계정 시스템을 그대로 사용하고 있어 'Personal' 플랜만 표시됩니다. Caret의 요금 체계에 맞게 'Free', 'Basic' 구독 플랜과 'Pay as you go' 옵션을 추가하여 사용자에게 정확한 정보를 제공해야 합니다.

3. 요구사항

- 구독 플랜 표시 변경:
 - 기존 'Personal' 플랜 드롭다운을 'Free', 'Basic'으로 변경합니다.
- 'Pay as you go' 옵션 추가:
 - 'Pay as you go' 체크박스과 설명을 추가합니다.
 - 설명: "* 구독 사용량을 모두 소진 후 추가 과금 됩니다."
- i18n 적용:
 - 새로 추가되는 모든 UI 텍스트(구독 플랜 이름, 옵션 설명 등)는 다국어를 지원해야 합니다.
- 기능 구현:
 - UI 변경 사항을 백엔드와 연동하여 실제 사용자 계정 정보에 반영되도록 합니다.

4. 분석 결과 (2025-07-11)

- 아키텍처: Caret은 Cline의 원본 컴포넌트를 직접 수정하는 대신, 래핑(wrapping)하고 확장하는 방식을 사용합니다.
 - `webview-ui/src/components/account/AccountView.tsx` 는 Caret 전용 컴포넌트인 `webview-`

caret-docs/tasks/{00d}-{issue-topic}.md

b8070cf caret / caret-docs / work-logs / luke / 2025-01-24-work-log.md

luke yang feat: Task #016 추가 요구사항 완료 - 웰컴 페이지 페르소나 초기화 및 caret 모드 강제 설정

Preview Code Blame 140 lines (114 loc) · 6.76 KB

2025-01-24 작업 로그 - Luke

작업 개요

- 작업자: Luke
- 작업일: 2025년 1월 24일
- 주요 작업: Task #016 - 페르소나 초기화 및 이미지 문제 해결 ☒

오늘의 목표

- ☒ 설정의 디버그메뉴 초기화 시 일부 데이터가 안지워지는 문제 해결
- ☒ 처음 실행/설치시 API 설정 때 언어가 설정되는 시점에 기본 페르소나(사랑이) 자동 설정
- ☒ TDD 방식으로 개발 진행
- ☒ 로깅 시스템 적용
- ☒ 작업 완료 후 커밋 & 푸시
- ☒ 추가 요구사항: 웰컴 페이지 넘어서 언어 선택 단계에서 페르소나 초기화
- ☒ 추가 요구사항: 초기 모드가 cline이 아닌 caret 모드로 설정

진행 상황

☒ Phase 0: 사전 분석 및 문서 검토

- ☒ 작업 로그 생성 및 태스크 확인 완료

caret/caret-docs/work-logs/{developer_name}/yyyy-mm-dd.md

8. 사업모델


AI 구독 모델

비전

AI와 개발자가 서로를 이해하고 협업할 수 있는 새로운 개발 환경을 제공하여 소프트웨어 개발의 장벽을 낮추고 혁신을 가속화합니다.

미션

투명하고 확장 가능한 오픈소스 생태계를 구축하여 누구나 신뢰할 수 있는 AI 개발 도구를 사용할 수 있도록 합니다.



Caretive
Your Creative Coding Companion.
caretive ai

Free
무료

개인 학습과 작은 프로젝트에 완벽한 무료 플랜

- ✓ 25 크레딧 무료 제공
- ✓ 제미나이 프로 기준 25회 전송
- ✓ 제미나이 2.5 플래시 기준 100회 전송
- ✓ 커뮤니티 지원

준비중

인기
Basic
\$10/월

개인 개발자와 소규모 팀을 위한 핵심 플랜

- ✓ 300 크레딧/월 포함
- ✓ 모든 AI 모델 접근
- ✓ 24/7 이메일 지원
- ✓ 사용량 초과 시 추가 과금 가능

준비중

Pay as You Go
사용량 기반

Basic 플랜 구독 후 선택할 수 있는 추가 서비스

- ✓ Basic 플랜 구독 필수
- ✓ 사용한 만큼만 지불
- ✓ 별도 제한 없음
- ✓ 자동 청구 시스템

준비중

Business
문의

기업용 맞춤형 솔루션

- ✓ AI 개발도구 컨설팅
- ✓ 커스텀 버전 제작
- ✓ 단체 및 기업용 계정 관리
- ✓ 비용 관리 솔루션

✉ 문의하기

이름 : 김기환 Kim Ki Hwan

캐럿티브 CEO

학력 : 숭실대학교 컴퓨터학부 학사

주요 경력

2025 캐럿티브 CEO
2023 히어로손해사정 통합시스템 설계 및 개발
2018-2022 GS SHOP NIKE CTM 정산 시스템 개발
2017 퍼스트데이타 백오피스 개발
2006-2016 AXA, 삼성화재, 흥국화재 시스템 개발



VS-Code plugin Market, B2C 클라우드 시장을 통한 확산 및 RnD 지속

Cline, Cursor, Windsurf와 유사한 방식으로 월정액 및 사용량 기반의 과금, 구글 스타트업 지원 프로그램을 통한 35만달러(약 4억 8천만원)의 클라우드 크레딧 확보를 통한 비용 통제

- 사용자 데이터를 통한 연구개발 지속
- 국산 AI 코딩 도구로서 독자적 입지 확립

8. 사업모델

협력사와 수익세어 모델의 고객별 솔루션 개발 및 공급 사업 진행

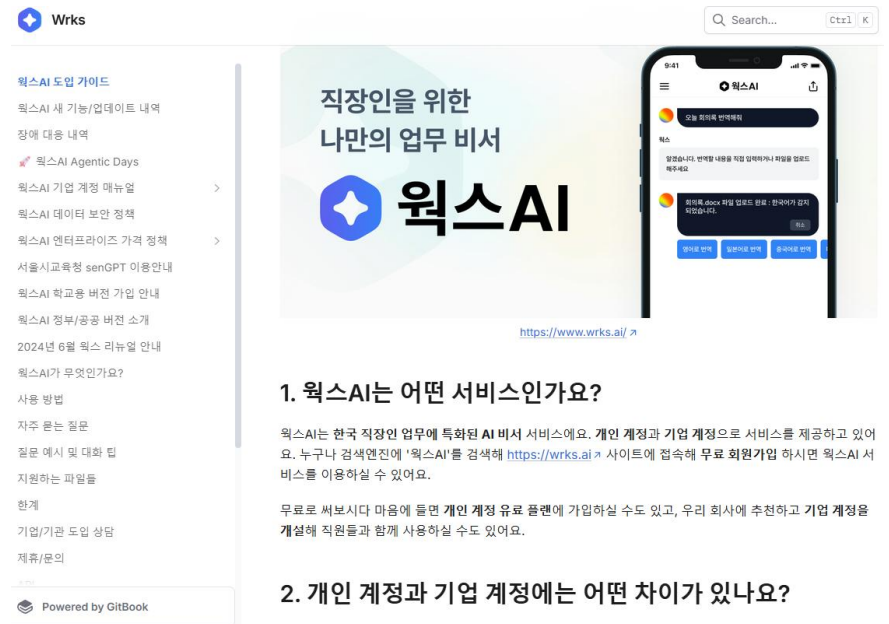
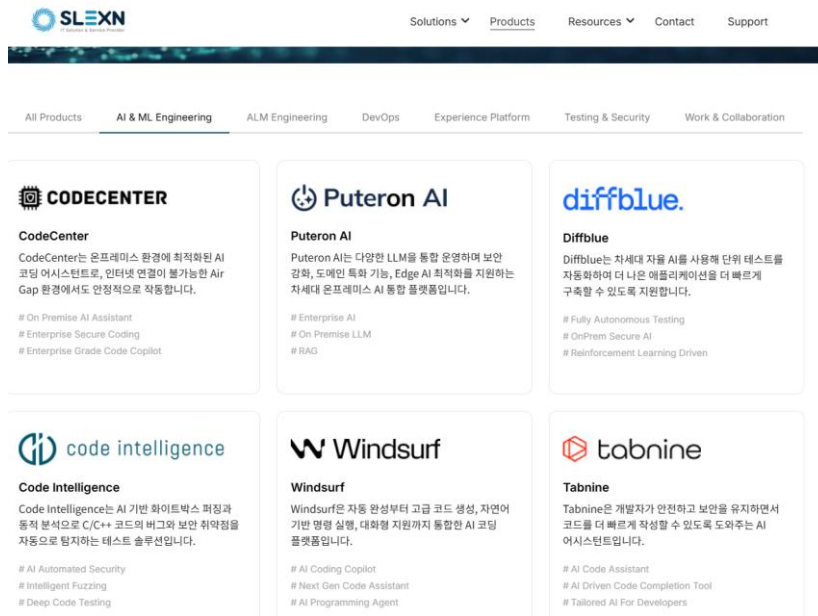
Business

문의

기업용 맞춤형 솔루션

- ✓ AI 개발도구 컨설팅
- ✓ 커스텀 버전 제작
- ✓ 단체 및 기업용 계정 관리
- ✓ 비용 관리 솔루션

✉ 문의하기



1. 웁스AI는 어떤 서비스인가요?

웁스AI는 한국 직장인 업무에 특화된 AI 비서 서비스예요. 개인 계정과 기업 계정으로 서비스를 제공하고 있어요. 누구나 검색엔진에 '웁스AI'를 검색해 <https://wrks.ai> 사이트에 접속해 무료 회원가입 하시면 웁스AI 서비스를 이용하실 수 있어요.

무료로 써보시다 마음에 들면 개인 계정 유료 플랜에 가입하실 수도 있고, 우리 회사에 추천하고 기업 계정을 개설해 직원들과 함께 이용하실 수도 있어요.

2. 개인 계정과 기업 계정에는 어떤 차이가 있나요?

9. 성장 계획

2025년 설립, 2026년 하반기에는 글로벌까지 빠른 성장 목표

2025.05 : 설립 (현재 단계)

- 구글/MS AI지원 프로그램 이용 비용 통제
- SEED 유치 및 출원 기반으로 기술보증기금 대출 실행
- 폐쇄 포럼/자문 운용을 통한 오픈소스 프로젝트 방향성 점검, 판매처 탐색

2025.06 : 오픈소스 프로젝트 조직 및 VS-Code 마켓플레이스 오픈

- <https://opencollective.com/> 개설
- 리팩토링 및 i18n완료 후 VS-code 플러그인 다국어 오픈
- 온/오프라인 교육 프로그램 개발을 통한 확산
- 1차 교육 행사 운용

2025년 하반기 : 안정화를 위한 기술 개발과 확산

- TIPS운용사 투자유치 2억
- 오픈소스 기반의 기술 개발 지속
- 3억 매출 달성 목표
- i18n기반으로 아태 비영어권 국가 탐색

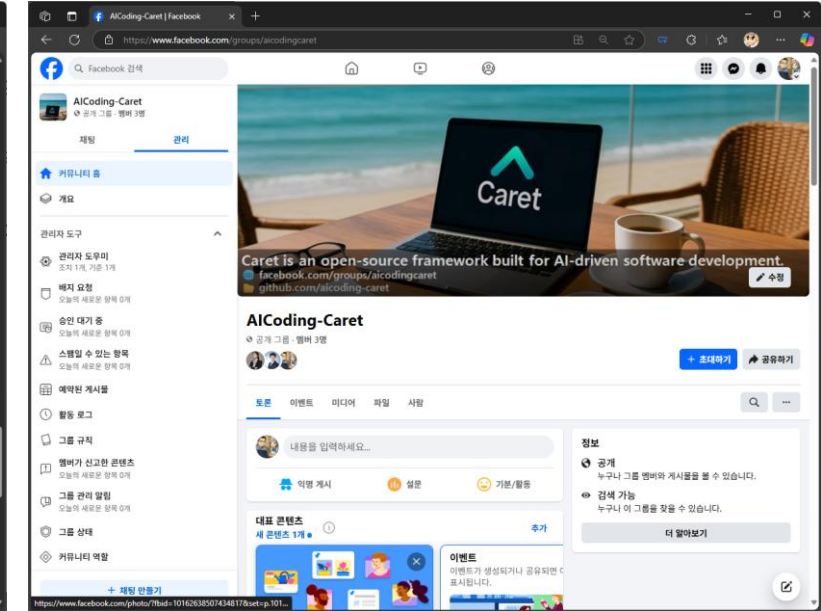
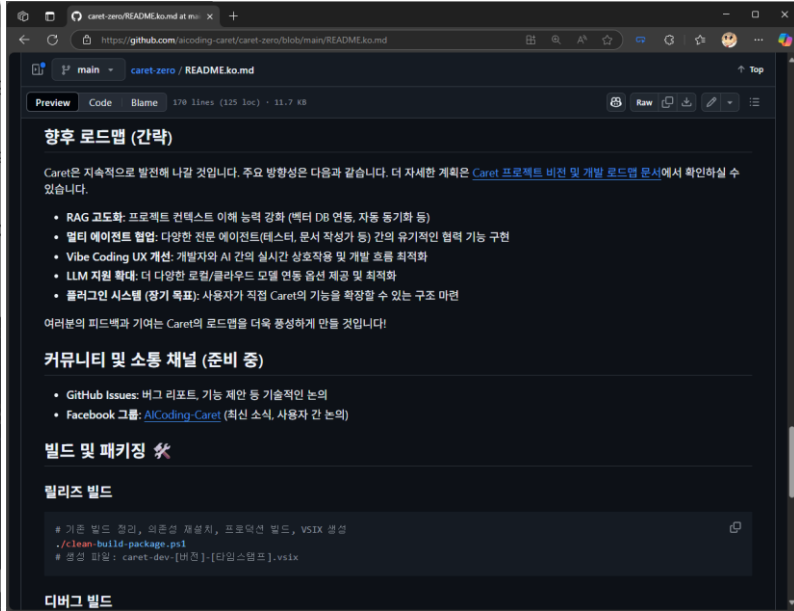
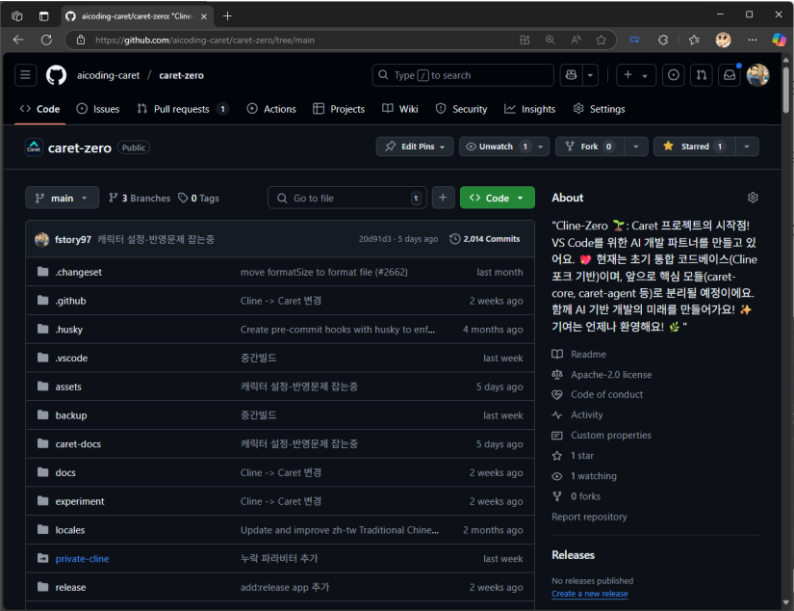
2026년 : 기술 고도화 및 글로벌 진출

- TIPS진행 2년 5억
- 디지털 새싹 운용 (순천대, 전남대 등 양병석 CTO의 기존 네트워크 활용)
- 시리즈 A유치
- 10억 매출 달성 목표, 미국 법인 설립 검토

10. 오픈소스 커뮤니티

오픈 소스 커뮤니티 중심 전략

- 대규모 투자가 이루어진 Cursor, Windsurf와 현실적인 대결의 어려움
- sLLM기술 발전은 AI기술도 Commodity화될 가능성이 높으므로 오픈소스 커뮤니티 중심의 브랜드 확보 필요
- AI코딩 도구 개발의 장벽은 생각보다 높지 않지만 폐쇄환경에 대한 요구사항과 기술 수요는 크게 증가 예상
- 명확한 기여 가이드라인, 코드의 윤리 강령 등을 포함한 커뮤니티 참여 정책을 수립
- 단기간에 확보가 어려운 고급 개발자 충원이 중요하므로 투자 기반의 무리한 인력 충원 보다는 오픈소스 커뮤니티 중심의 자유로운 서브 프로젝트 참여/기여 보상, 스톡옵션 등을 고려



11. 과제

1. 사업화

- 1.1. 로그인 및 구독 시스템
- 1.2. 단체 및 기업 멤버 관리
- 1.3. SDK 및 API

2. 콘텐츠

- 2.1. 교육 프로그램 개발
- 2.2. 사용 사례 개발
- 2.3. 브랜딩 및 마케팅
- 2.4. 글로벌 화

3. 조직 운용

- 3.1. 조직 및 커뮤니티 운용 및 관리

2. 연구 개발

2.1 모델 인프라 & MLOps

- 2.1.1 LLM 라우터 / sLLM 셀렉터
- 2.1.2 벡터-RAG 파이프라인
- 2.1.3 관측(Observability) 및 비용 메트릭
- 2.1.4. 시멘틱 기반의 AI모델 훈련/개발

2.2 실시간 협업 엔진

- 2.2.1 Yjs 기반 CRDT 동기화
- 2.2.2 멀티-에이전트 인터페이스

2.3 프롬프트 오케스트레이션

- 2.3.1 동적 시스템-프롬프트 DSL
- 2.3.2 Vector RAG + AST 청킹
- 2.3.3 시맨틱 캐싱 & 프롬프트 압축

2.4 안전성 & 품질 보증

- 2.4.1 AST 검증(ts-morph)
- 2.4.2 테스트 · Lint 자동화

III. 질의 응답 및 의견 청취



IV. 자유토론

1. 조직 결성 방법
2. 목표 및 활동 계획 수립



감사합니다.