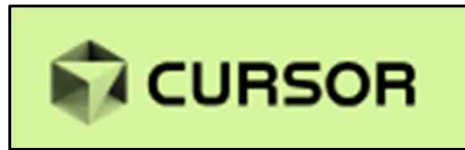


Cursor 기능 사용법 : Chat



1.1 Cursor 주요 기능



1.1 Cursor 주요 기능 설명

Cursor AI 코드 에디터의 기능

Tab
Agent
Background Agent
Inline Edit
Chat
Rules
Memories
Codebase Indexing
MCP
Context
Models

<https://docs.cursor.com/ko/get-started/concepts>

1.1 Cursor 기능 설명

Tab: 지능형 자동 완성

Tab 기능은 단순한 코드 자동 완성을 넘어, 개발자의 다음 코드를 예측하여 여러 줄에 걸친 코드 수정을 한 번에 제안하는 강력한 기능이다. 사용자의 최근 변경 사항과 코드의 전체적인 맥락을 파악하여 다음에 작성할 코드를 예측하고, **Tab** 키를 누르는 것만으로 제안된 코드를 채택할 수 있다. 이를 통해 타이핑 실수를 줄이고, 반복적인 코드 작성을 자동화하여 개발 속도를 획기적으로 향상시킨다.

- **다중 라인 수정:** 한 번에 여러 줄의 코드 변경을 제안한다.
- **스마트한 재작성:** 부주의하게 작성된 코드도 문맥에 맞게 수정해 준다.
- **빠른 이동 및 수정:** **Tab** 키를 연이어 눌러 여러 파일에 걸친 수정 사항을 빠르게 적용할 수 있다.

1.1 Cursor 기능 설명

Agent: 자율적인 AI 개발자

Agent는 단순한 코드 제안을 넘어, 개발자의 지시를 받아 스스로 복잡한 작업을 수행하는 AI 기능이다. 예를 들어 "새로운 기능을 추가해줘" 또는 "이 버그를 수정해줘"와 같은 자연어 명령을 내리면, Agent가 전체 코드 베이스를 이해하고, 여러 파일을 동시에 수정하며, 필요한 경우 터미널 명령어까지 실행하여 작업을 완료한다. 개발자는 Agent의 작업 과정을 실시간으로 확인하고 제어할 수 있다.

- **엔드투엔드(End-to-End) 작업 수행:** 기능 구현, 버그 수정, 리팩토링 등 복잡한 개발 작업을 처음부터 끝까지 처리한다.
- **컨텍스트 자동 파악:** 전체 코드베이스의 구조와 맥락을 이해하여 작업에 필요한 파일을 스스로 찾아낸다.
- **다중 파일 동시 수정:** 하나의 작업을 위해 여러 파일을 동시에 생성하거나 수정할 수 있다.

1.1 Cursor 기능 설명

Background Agent: 백그라운드 작업 수행

Background Agent는 Agent 기능을 클라우드 기반 환경에서 백그라운드로 실행하는 기능이다. 개발자의 로컬 컴퓨터에 영향을 주지 않고, 저장소를 복제하여 독립적인 환경에서 작은 작업들을 병렬로 처리한다. 작업이 완료되면 개발자는 변경 사항을 검토하고 풀 리퀘스트(Pull Request)를 통해 손쉽게 프로젝트에 반영할 수 있다. 이를 통해 개발자는 다른 작업에 집중하면서 동시에 여러 개선 사항을 처리할 수 있다.

- **비동기 병렬 작업:** 여러 개의 작은 작업을 동시에 백그라운드에서 처리하여 효율성을 높인다.
- **로컬 환경 독립성:** 개발자의 로컬 개발 환경에 전혀 영향을 주지 않고 작업을 수행한다.
- **자동 풀 리퀘스트 생성:** 작업 완료 후 검토할 수 있는 풀 리퀘스트를 자동으로 생성하여 편리하다.

1.1 Cursor 기능 설명

Inline Edit: 인라인 코드 수정 및 생성

Inline Edit는 코드 편집기 내에서 직접 자연어 명령으로 코드를 수정하거나 생성하는 기능이다. 특정 코드 블록을 선택하고 `Ctrl+K` (macOS에서는 `Cmd+K`)를 누르면 입력창이 나타나며, 여기에 "이 함수를 비동기 함수로 바꿔줘"와 같이 원하는 변경 사항을 글로 작성하면 AI가 즉시 코드를 수정해 준다. 코드를 선택하지 않고 명령을 입력하면 새로운 코드를 생성한다.

- **선택 영역 수정:** 선택한 코드를 지시에 따라 정확하게 수정한다.
- **새 코드 생성:** 커서 위치에 지시에 맞는 새로운 코드를 생성한다.
- **빠른 질문과 답변:** 코드에 대해 질문하고 제안된 해결책을 즉시 코드로 변환할 수 있다.

1.1 Cursor 기능 설명

Chat: 코드베이스와 대화하기

Chat 기능은 코드베이스 전체를 이해하는 AI와 대화할 수 있는 인터페이스이다. `Ctrl+L` (macOS에서는 `Cmd+L`)을 눌러 채팅창을 열고, "이 프로젝트에서 사용자 인증은 어떻게 처리해?"와 같이 질문하면, AI가 코드 전체를 분석하여 맥락에 맞는 답변을 제공한다. 특정 파일을 @ 기호로 참조하거나, 웹 검색 결과를 참고하여 질문할 수도 있어 복잡한 문제 해결에 유용하다.

- **코드베이스 기반 답변:** 프로젝트 전체 코드에 대한 깊이 있는 질문에 답변한다.
- **다양한 컨텍스트 활용:** @ 기호를 사용하여 특정 파일, 폴더, 웹 페이지, 라이브러리 문서 등을 참조하여 질문의 정확도를 높일 수 있다.
- **다중 파일 수정:** 채팅을 통해 여러 파일에 걸친 코드 변경을 요청하고 적용할 수 있다.

1.1 Cursor 기능 설명

Rules: AI 행동 규칙 정의

Rules는 AI가 코드를 생성하거나 수정할 때 따라야 할 규칙을 `.cursorrules` 파일을 통해 직접 정의하는 기능이다. 예를 들어, 특정 코딩 스타일 가이드(예: "항상 세미콜론 사용")나 프로젝트의 아키텍처 패턴을 규칙으로 설정해두면, AI가 해당 규칙을 준수하여 일관성 있는 코드를 생성한다. 이를 통해 팀 전체의 코드 품질을 유지하고, 반복적인 리뷰 과정을 줄일 수 있다.

- **사용자 맞춤형 AI:** 프로젝트의 특성과 요구사항에 맞게 AI의 행동을 맞춤 설정할 수 있다.
- **코드 일관성 유지:** 팀의 코딩 표준과 스타일 가이드를 AI가 자동으로 따르도록 하여 코드의 일관성을 보장한다.

1.1 Cursor 기능 설명

Memories: 프로젝트 정보 기억

Memories는 AI가 채팅 대화나 코드 수정 과정에서 학습한 프로젝트 관련 중요 정보를 자동으로 기억하는 기능이다. 예를 들어, 특정 라이브러리의 사용법이나 프로젝트의 주요 아키텍처 결정 사항을 AI가 한번 학습하면, 나중에 관련된 작업을 할 때 그 정보를 기억하고 활용한다. 이 정보는 `.cursor/rules` 디렉토리에 `learned-memories.mdc` 파일로 저장되며, 사용자가 직접 관리할 수도 있다.

- **지속적인 컨텍스트 유지:** 세션이 바뀌어도 AI가 프로젝트의 중요 정보를 기억하여 일관된 지원을 제공한다.
- **자동 학습 및 저장:** AI가 대화 내용을 바탕으로 기억할 만한 정보를 스스로 판단하고 저장한다.

1.1 Cursor 기능 설명

Codebase Indexing: 코드베이스 전체 이해

Codebase Indexing은 프로젝트의 모든 코드를 AI가 미리 분석하고 색인화하여 더 빠르고 정확하게 코드의 맥락을 이해하도록 돕는 기능이다. 프로젝트를 열면 Cursor가 자동으로 코드베이스 인덱싱을 시작하며, 이를 통해 AI는 특정 코드 조각이 프로젝트의 다른 부분과 어떻게 상호작용하는지 깊이 이해하게 된다. 이는 Agent나 Chat 기능의 정확도를 크게 향상시킨다.

- **정확한 컨텍스트 이해:** 코드베이스 전체의 구조와 관계를 파악하여 AI 기능의 정확도를 높인다.
- **빠른 검색 및 분석:** 의미 기반 검색을 통해 관련 코드를 신속하게 찾아낸다.
- **보안:** 소스 코드는 서버에 저장되지 않으며, 파일명은 난독화되고 코드 조각은 암호화되어 안전하게 처리된다.

1.1 Cursor 기능 설명

MCP (Model Context Protocol): 외부 도구 연동

MCP는 Cursor의 AI 모델이 외부의 다른 도구나 서비스와 상호작용할 수 있도록 하는 개방형 프로토콜이다. 개발자는 MCP 서버를 직접 구축하여 Cursor의 기능을 확장할 수 있다. 예를 들어, 사내 문서 시스템이나 특정 데이터베이스와 연동하는 MCP 서버를 만들면, AI가 해당 시스템의 정보를 가져와 코드 생성에 활용할 수 있다.

- **기능 확장성:** 사용자가 직접 필요한 도구를 만들어 Cursor의 AI 기능과 연동할 수 있다.
- **자동화된 워크플로우:** 반복적인 작업을 자동화하는 커스텀 도구를 개발하여 생산성을 높일 수 있다.

1.1 Cursor 기능 설명

Context: AI에게 제공되는 정보

Context는 AI가 코드 생성이나 수정을 할 때 참고하는 모든 정보를 의미한다. 여기에는 사용자가 입력한 지시사항(의도 컨텍스트)과 현재 파일의 코드, 관련된 다른 파일, 에러 메시지 등 현재 상태에 대한 정보(상태 컨텍스트)가 모두 포함된다. Cursor는 자동으로 코드베이스에서 관련성 높은 정보를 컨텍스트로 가져오지만, 사용자가 @ 기호를 사용해 직접 특정 파일이나 폴더를 컨텍스트로 지정하여 AI의 정확도를 더욱 높일 수 있다.

- **의도(Intent) 컨텍스트:** 사용자가 AI에게 원하는 바를 명확히 전달하는 정보 (예: "버튼 색깔을 파란색으로 바꿔줘").
- **상태(State) 컨텍스트:** 현재 코드의 상태를 설명하는 정보 (예: 현재 열려있는 파일의 코드, 오류 로그).

1.1 Cursor 기능 설명

Models: 최신 AI 모델 지원

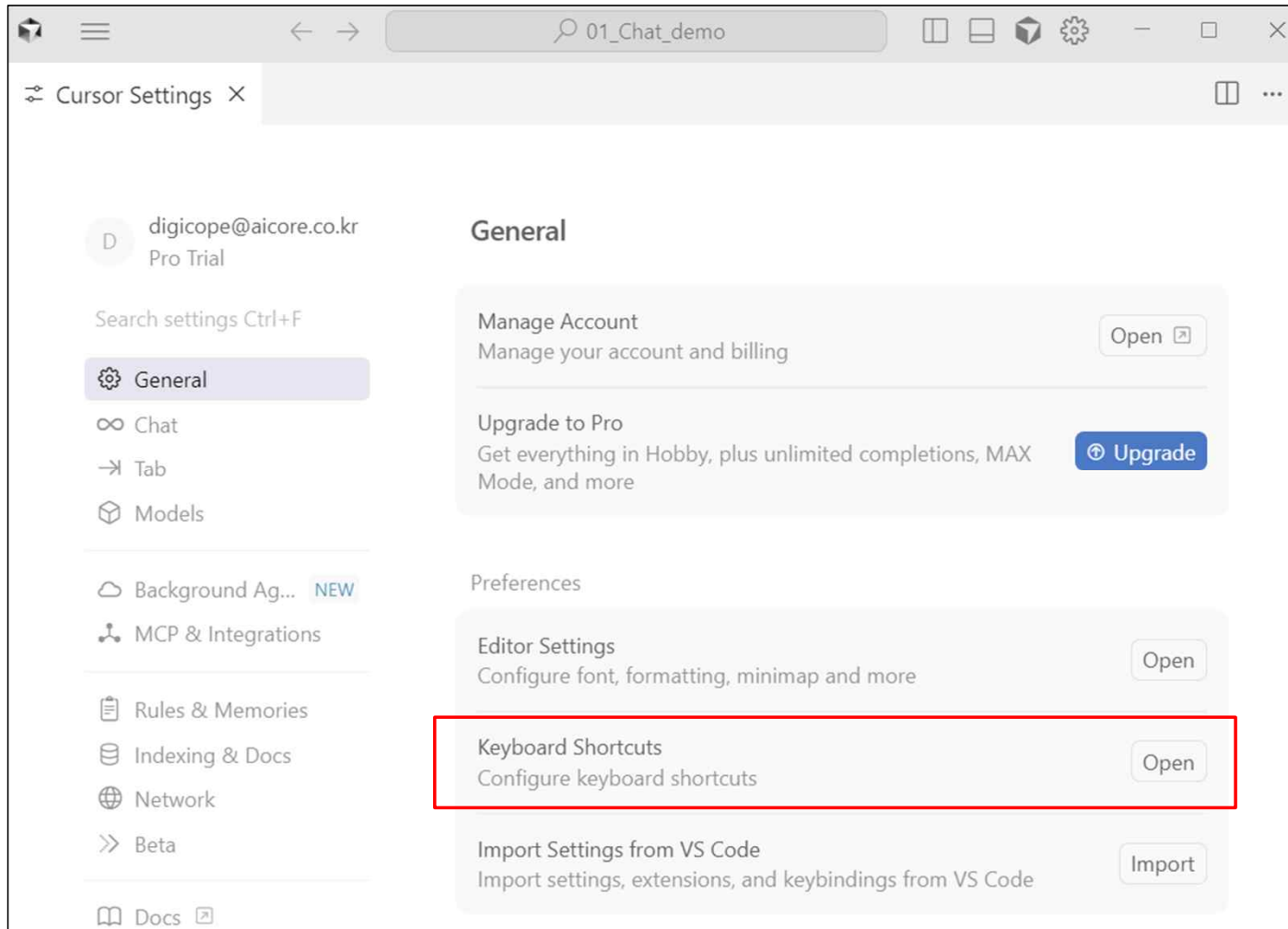
Cursor는 OpenAI의 GPT 시리즈, Anthropic의 Claude, Google의 Gemini 등 업계 최고 수준의 **다양한 AI 모델**을 지원한다. 사용자는 자신의 필요와 선호에 따라 사용할 모델을 선택할 수 있으며, Cursor는 각 모델의 최신 버전을 빠르게 도입하여 항상 최고의 성능을 제공한다. Pro 요금제 이상 사용자는 더 많은 모델을 제한 없이 사용할 수 있다.

- **다양한 모델 선택권:** OpenAI, Anthropic, Google 등의 최신 AI 모델을 자유롭게 선택하여 사용할 수 있다.
- **최고 성능 유지:** 지속적인 모델 업데이트를 통해 항상 최첨단 AI 기술을 활용할 수 있다.

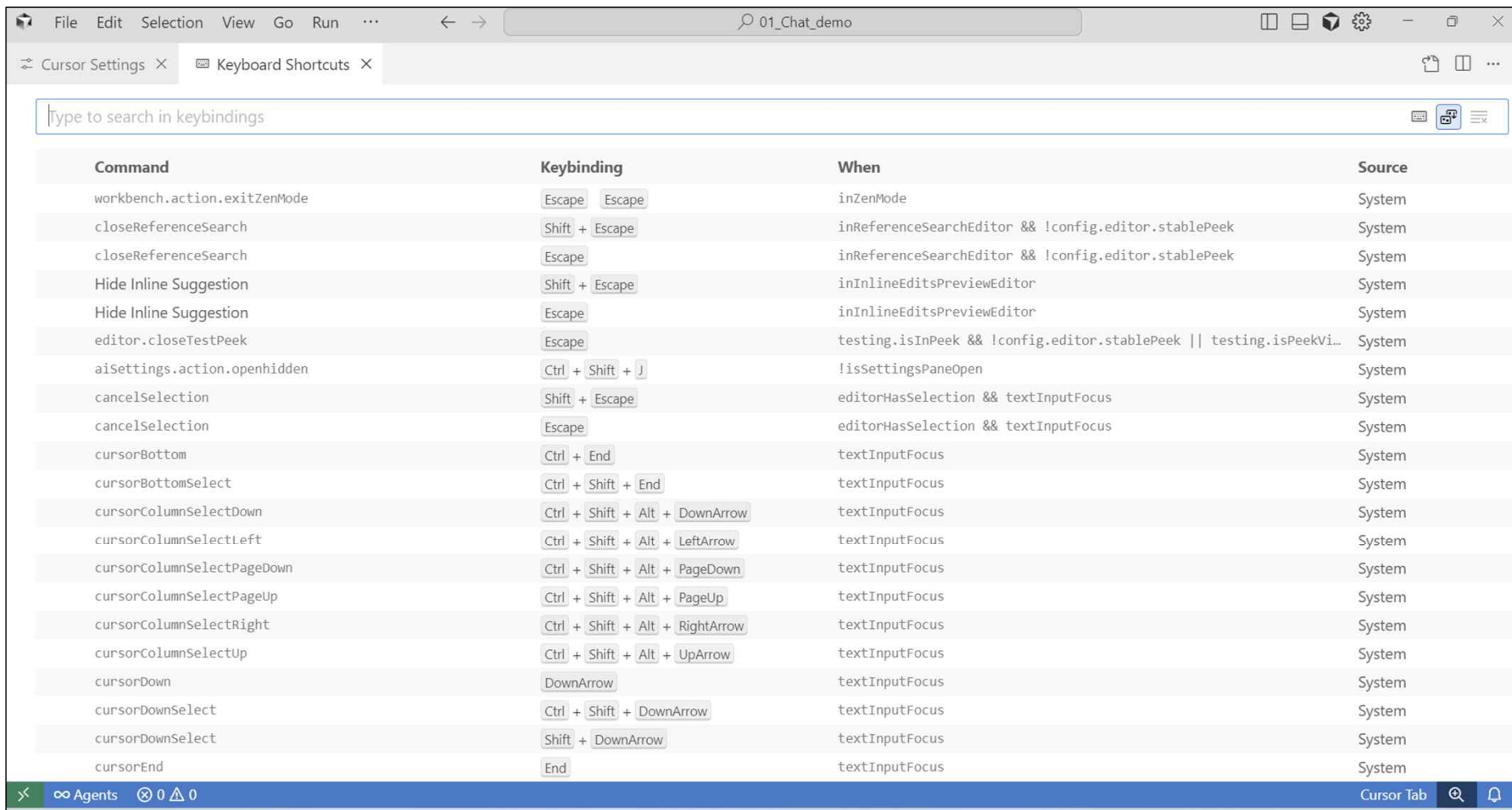
1.2 Cursor 단축키 요약



1.2 Cursor 단축키 요약



1.2 Cursor 단축키 요약



Command	Keybinding	When	Source
workbench.action.exitZenMode	Escape	inZenMode	System
closeReferenceSearch	Shift + Escape	inReferenceSearchEditor && !config.editor.stablePeek	System
closeReferenceSearch	Escape	inReferenceSearchEditor && !config.editor.stablePeek	System
Hide Inline Suggestion	Shift + Escape	inInlineEditsPreviewEditor	System
Hide Inline Suggestion	Escape	inInlineEditsPreviewEditor	System
editor.closeTestPeek	Escape	testing.isInPeek && !config.editor.stablePeek testing.isPeekVi...	System
aiSettings.action.openhidden	Ctrl + Shift + J	!isSettingsPaneOpen	System
cancelSelection	Shift + Escape	editorHasSelection && textInputFocus	System
cancelSelection	Escape	editorHasSelection && textInputFocus	System
cursorBottom	Ctrl + End	textInputFocus	System
cursorBottomSelect	Ctrl + Shift + End	textInputFocus	System
cursorColumnSelectDown	Ctrl + Shift + Alt + DownArrow	textInputFocus	System
cursorColumnSelectLeft	Ctrl + Shift + Alt + LeftArrow	textInputFocus	System
cursorColumnSelectPageDown	Ctrl + Shift + Alt + PageDown	textInputFocus	System
cursorColumnSelectPageUp	Ctrl + Shift + Alt + PageUp	textInputFocus	System
cursorColumnSelectRight	Ctrl + Shift + Alt + RightArrow	textInputFocus	System
cursorColumnSelectUp	Ctrl + Shift + Alt + UpArrow	textInputFocus	System
cursorDown	DownArrow	textInputFocus	System
cursorDownSelect	Ctrl + Shift + DownArrow	textInputFocus	System
cursorDownSelect	Shift + DownArrow	textInputFocus	System
cursorEnd	End	textInputFocus	System

Ctrl+M을 누르고 Ctrl+S를 누르면 Keyboard Shortcuts 화면이 보임

1.2 Cursor 단축키 요약

핵심 AI 기능 단축키 (필수)

이 단축키들은 Cursor의 가장 본질적인 기능이며, 생산성 향상을 위해 반드시 알아두어야 한다.

기능	Mac	Windows / Linux	설명
인라인 AI (코드 수정/생성)	Cmd + K	Ctrl + K	선택한 코드 블록을 수정하거나, 새로운 코드를 생성하도록 지시하는 가장 중요한 기능이다.
AI 채팅 (질문/전체 파일)	Cmd + L	Ctrl + L	에디터 우측에 채팅 패널을 열어 AI에게 질문하거나 프로젝트 전체에 대한 작업을 요청할 수 있다.
AI에 코드 조각 추가	Cmd + Shift + L	Ctrl + Shift + L	선택한 코드를 AI 채팅의 컨텍스트에 포함시켜 더 정확한 답변을 얻도록 한다.

1.2 Cursor 단축키 요약

주요 기능별 단축키

1. AI 채팅 (Cmd/Ctrl + L)

AI 채팅 패널 내에서 사용하는 주요 단축키는 다음과 같다.

기능	Mac	Windows / Linux	설명
생성된 코드 적용하기	Cmd + Enter	Ctrl + Enter	AI가 제안한 코드 변경사항을 현재 파일에 전체 적용한다.
생성된 코드 거절하기	Cmd + Backspace	Ctrl + Backspace	AI가 제안한 모든 코드 변경사항을 거절하고 취소한다.
AI 모델 전환하기	Cmd + /	Ctrl + /	채팅 중에 GPT-4, Claude 등 다른 AI 모델로 쉽게 전환할 수 있다.
새로운 채팅 시작	Cmd + N / R	Ctrl + N / R	이전 대화 내용을 초기화하고 새로운 채팅 세션을 시작한다.

1.2 Cursor 단축키 요약

2. 인라인 AI (`Cmd/Ctrl + K`)

코드 편집창에서 직접 AI를 호출했을 때 사용하는 단축키이다.

기능	Mac	Windows / Linux	설명
변경사항 적용하기	<code>Cmd</code> + <code>Enter</code>	<code>Ctrl</code> + <code>Enter</code>	인라인 AI가 제안한 코드를 수락하여 적용한다.
변경사항 취소하기	<code>Esc</code> 또는 <code>Cmd</code> + <code>⌫</code>	<code>Esc</code> 또는 <code>Ctrl</code> + <code>⌫</code>	제안된 코드를 적용하지 않고 원래 상태로 되돌린다.

1.2 Cursor 단축키 요약

3. AI 자동 완성

코드를 작성할 때 AI가 회색으로 다음 코드를 제안하며, 이때 아래 단축키를 사용할 수 있다.

기능	Mac	Windows / Linux	설명
제안 수락	Tab	Tab	AI가 제안하는 자동 완성 코드를 그대로 받아들인다.
제안 거절	Esc	Esc	자동 완성 제안을 무시하고 직접 코드를 계속 작성한다.
단어 단위로 수락	Cmd + →	Ctrl + →	제안된 자동 완성 코드의 전체가 아닌 일부(한 단어)만 수락한다.

1.2 Cursor 단축키 요약

기본 VS Code 공통 단축키

Cursor는 VS Code의 포크(fork)이므로, 아래의 표준 단축키들 역시 매우 유용하다.

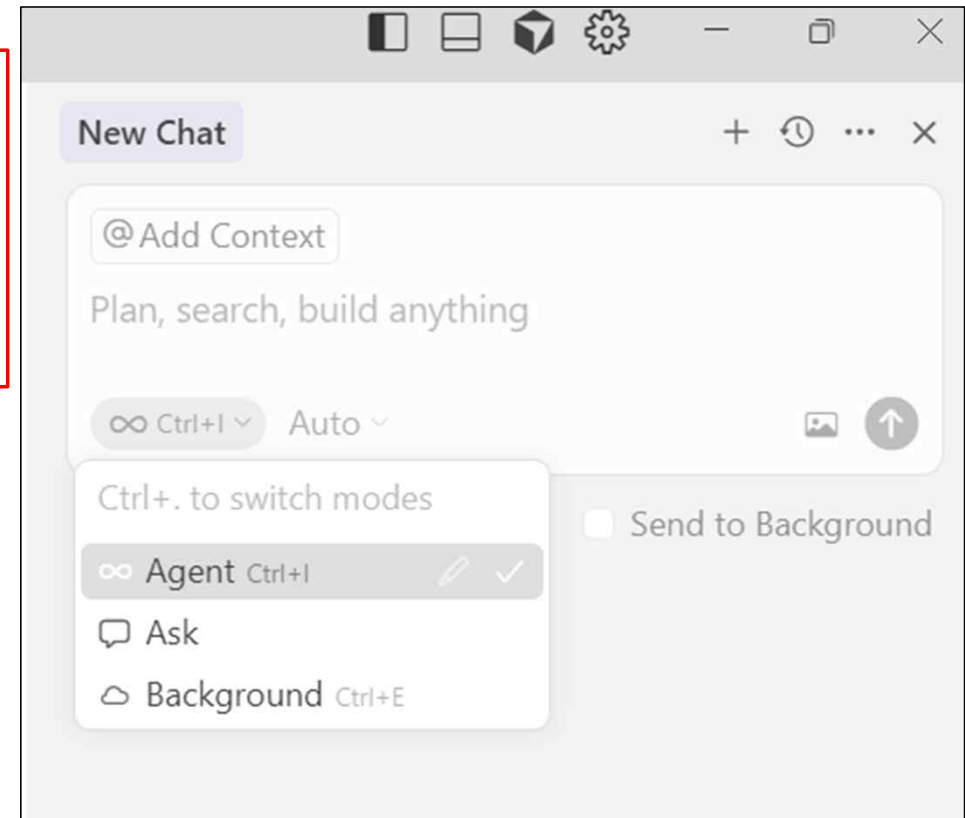
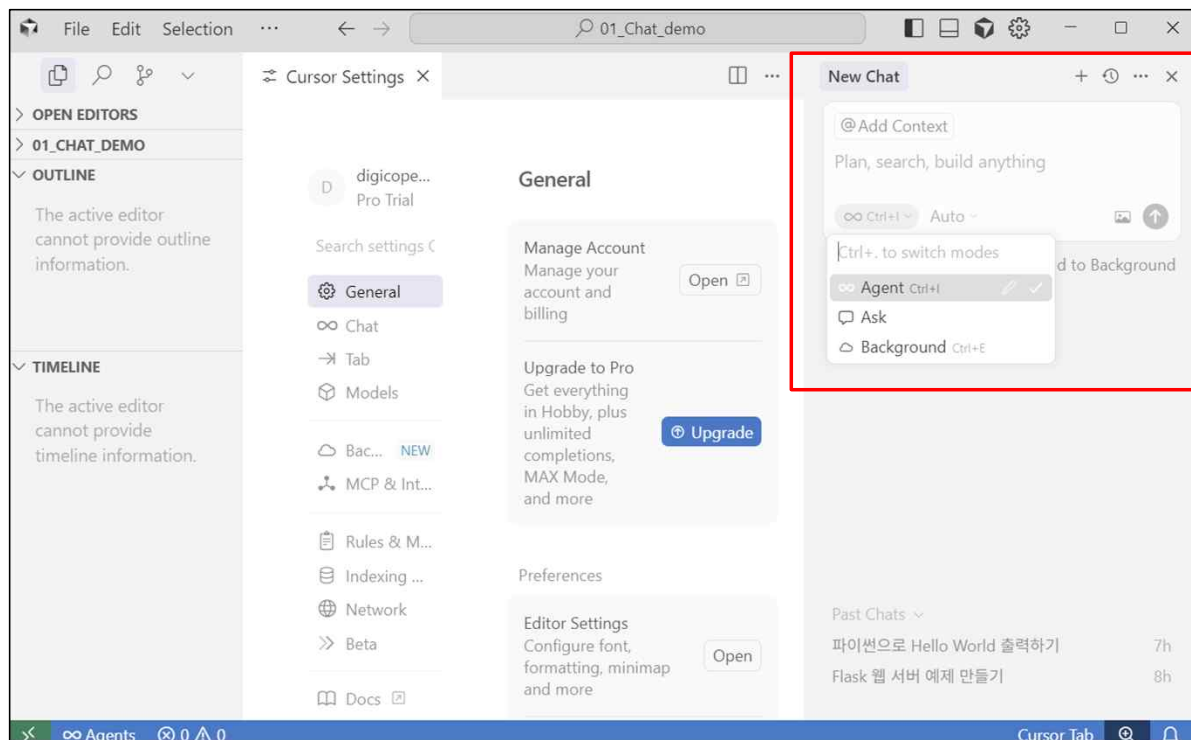
기능	Mac	Windows / Linux
명령 팔레트 열기	<code>Cmd + Shift + P</code>	<code>Ctrl + Shift + P</code>
파일 빠르게 열기	<code>Cmd + P</code>	<code>Ctrl + P</code>
사이드바 보이기/숨기기	<code>Cmd + B</code>	<code>Ctrl + B</code>
터미널 열기/닫기	<code>Ctrl + `</code>	<code>Ctrl + `</code>
전체 검색	<code>Cmd + Shift + F</code>	<code>Ctrl + Shift + F</code>
한 줄 주석 처리/해제	<code>Cmd + /</code>	<code>Ctrl + /</code>

1.3 Cursor 주요 기능 : Chat



1.3 Cursor 주요 기능 : Chat

Cursor화면 우측 상단의 아이콘을 클릭하여 AI 사이드바(AI Pane)가 보여진다



1.1 Cursor 기능 설명

Chat: 코드베이스와 대화하기

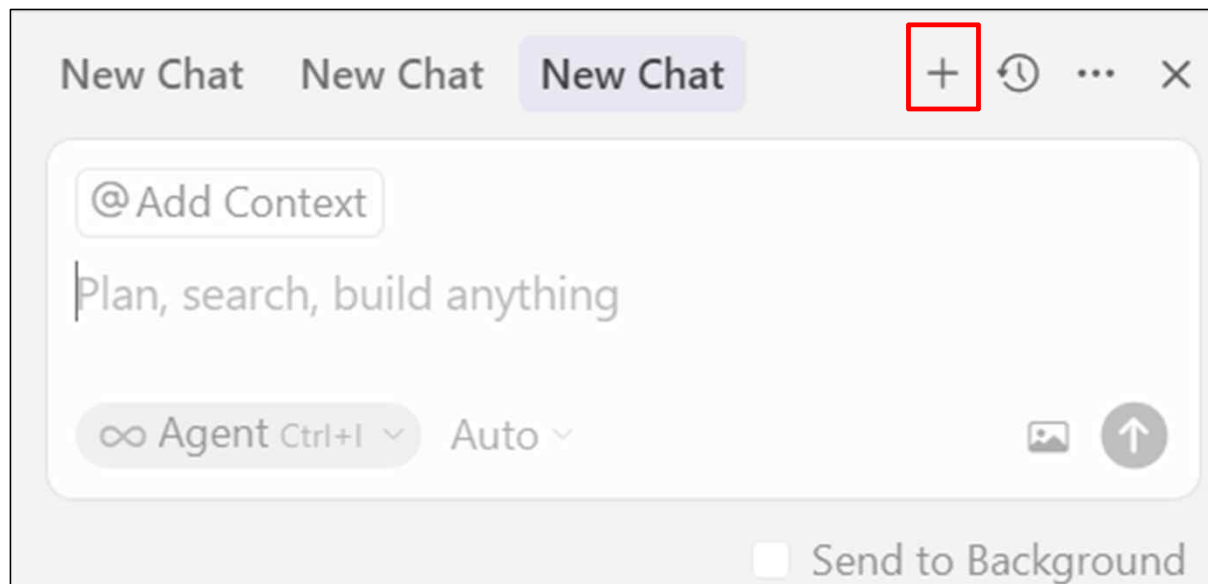
Chat 기능은 코드베이스 전체를 이해하는 AI와 대화할 수 있는 인터페이스이다. `Ctrl+L` (macOS에서는 `Cmd+L`)을 눌러 채팅창을 열고, "이 프로젝트에서 사용자 인증은 어떻게 처리해?"와 같이 질문하면, AI가 코드 전체를 분석하여 맥락에 맞는 답변을 제공한다. 특정 파일을 @ 기호로 참조하거나, 웹 검색 결과를 참고하여 질문할 수도 있어 복잡한 문제 해결에 유용하다.

- **코드베이스 기반 답변:** 프로젝트 전체 코드에 대한 깊이 있는 질문에 답변한다.
- **다양한 컨텍스트 활용:** @ 기호를 사용하여 특정 파일, 폴더, 웹 페이지, 라이브러리 문서 등을 참조하여 질문의 정확도를 높일 수 있다.
- **다중 파일 수정:** 채팅을 통해 여러 파일에 걸친 코드 변경을 요청하고 적용할 수 있다.

1.3 Cursor 주요 기능 : Chat

New Chat

[+] 버튼을 클릭할 때마다 비어 있는 새로운 채팅 화면이 만들어진다
대화 내용은 자동으로 저장된다



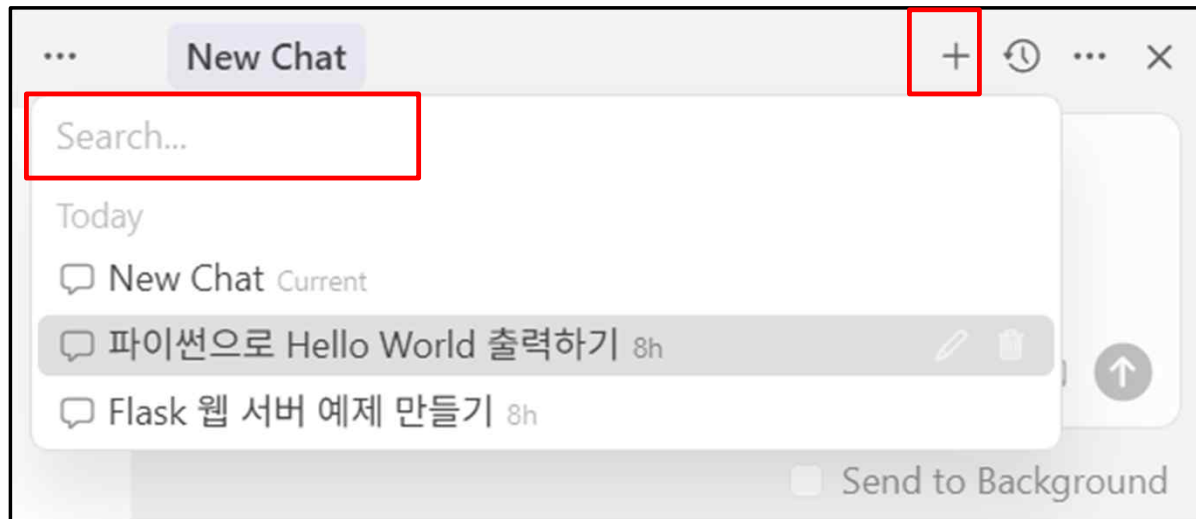
1.3 Cursor 주요 기능 : Chat

Show Chat History

시계 모양 아이콘 버튼을 클릭하면 이전 채팅 목록이 나타난다

목록에서 선택해서 이전에 작성했던 채팅 내용을 다시 가져올 수 있다

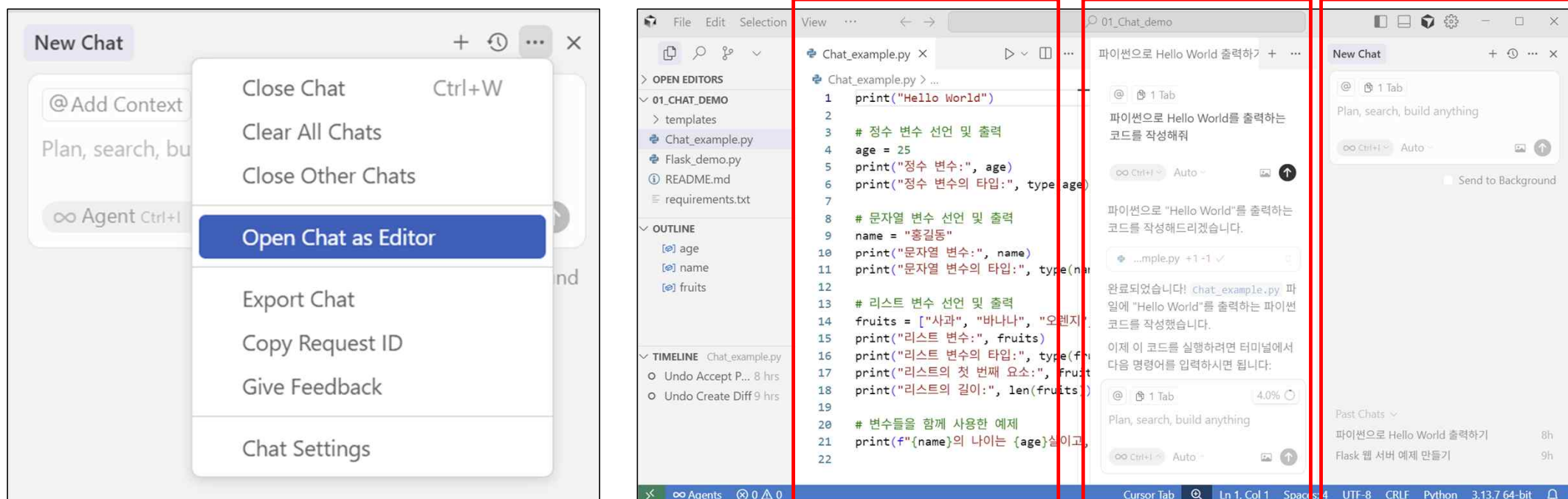
목록 옆의 연필 아이콘으로 이름을 수정할 수 있고 휴지통 아이콘을 클릭하여 삭제할 수 있다. 목록이 많을 때는 Search 입력란에서 목록을 빠르게 검색할 수 있다



1.3 Cursor 주요 기능 : Chat

Open Chat as Editor

[...] 버튼을 클릭하고 [Open Chat as Editor]를 클릭하면 별도의 에디터 탭에 Chat이 표시된다. 마우스로 끌어다 에디터 옆에 동시에 배치하여 사용이 가능하다



Chat의 Mode

Agent/Ask/Background/Custom 모드를 제공한다

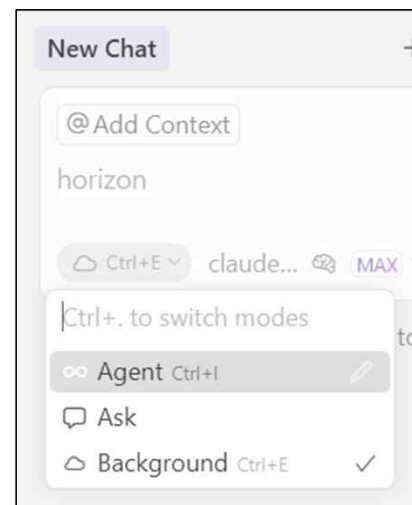
Agent는 특정 작업에 최적화된 여러 모드를 제공해. 각 모드는 워크플로우에 맞게 서로 다른 기능과 도구가 활성화돼 있어.

Mode	For	Capabilities	Tools
<u>Agent</u>	복잡한 기능 개발, 리팩터링	자율 탐색, 다중 파일 편집	모든 도구 사용 가능
<u>Ask</u>	학습, 계획, 질문	읽기 전용 탐색, 자동 변경 없음	검색 도구만
<u>Custom</u>	특화된 워크플로우	사용자 정의 기능	구성 가능

1.3 Cursor 주요 기능 : Chat

1. Agent 모드 (Ctrl+I)

- **역할:** AI가 코딩 조수처럼 적극적으로 개입해서, 코드 생성·수정·디버깅 등을 자동으로 도와줌.
- **특징:**
 - 사용자가 직접 실행하지 않아도, AI가 "에이전트"처럼 여러 단계를 거쳐 답안을 만들어냄.
 - 연속적인 맥락 유지, 문제 해결 과정 자동화.
 - 예: "파이썬으로 웹 스크래핑 코드 만들어줘" → 완성된 코드, 실행 방법, 추가 개선점까지 제안.



1.3 Cursor 주요 기능 : Chat

2. Ask 모드

- **역할:** 단순히 질문 → 답변 형태.
 - **특징:**
 - ChatGPT처럼 **지시형/질문형 질의응답**에 적합.
 - 코드 자동 실행이나 적극적 개입 없이, "설명, 예제, 참고 코드"를 위주로 답함.
 - 예: "Python에서 list와 tuple 차이 알려줘" → 정의와 예제 코드 답변.
-

3. Background 모드 (Ctrl+E)

- **역할:** 백그라운드에서 코드 실행·결과 확인만 조용히 수행.
- **특징:**
 - 사용자가 작성한 코드나 요청을 **백그라운드 실행기처럼 처리**.
 - 긴 프로세스, 데이터 분석, 로그 확인 등에 적합.
 - Ask/Agent보다 출력이 간단하고 조용함.
 - 예: "이 코드 1000번 반복 실행하고 결과 로그만 남겨" → 백그라운드에서 수행 후 결과만 반환.

1.3 Cursor 주요 기능 : Chat

- ◆ **Custom 모드 (Beta)** : **CursorSettings → Chat → Custom Modes**에서 켜야 사용가능
 - **역할**: 사용자가 원하는 방식으로 AI의 답변 스타일과 동작 방식을 **직접 정의**할 수 있는 모드.
 - **특징**:
 - 프롬프트(시스템 메시지 역할)를 미리 저장해두고, 매번 새로운 대화에서 불러와 사용 가능.
 - 특정 목적에 맞는 **전문화된 비서처**럼 AI를 세팅 가능.
 - 예:
 - **코드 리뷰어 모드**: "내가 작성한 코드를 효율성과 가독성 기준으로 리뷰해줘."
 - **강의 보조 모드**: "질문에 대해 교재 스타일(~이다, ~한다)로 정리해서 답변해줘."
 - **번역 모드**: "모든 답변을 영어↔한국어 병기해서 보여줘."

1.3 Cursor 주요 기능 : Chat

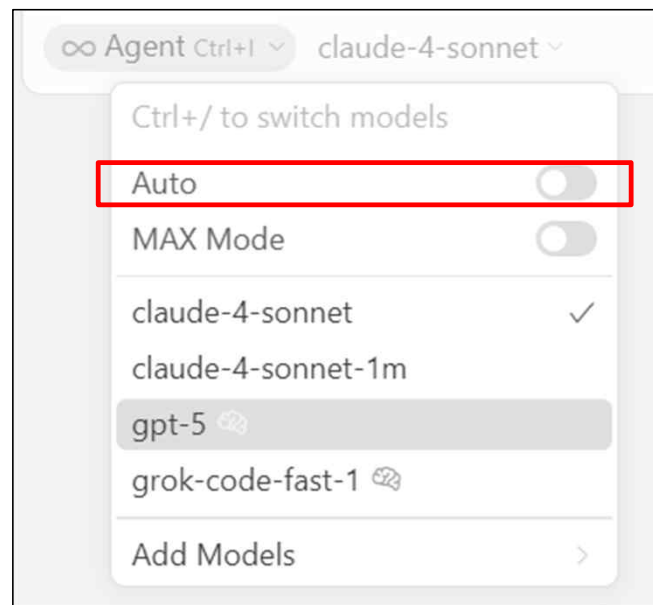
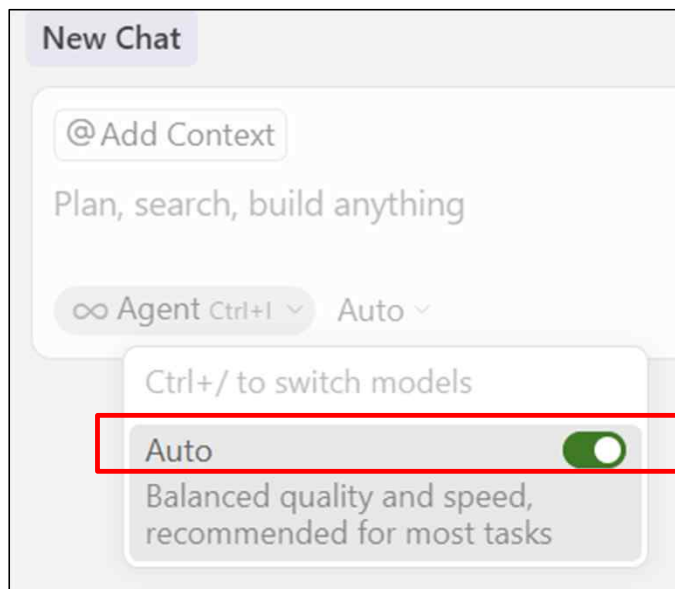
👉 정리하면:

- **Agent** → 능동형, AI가 알아서 도와주는 조수 모드.
- **Ask** → 수동형, 질문/답변 위주의 기본 모드.
- **Background** → 실행형, 조용히 코드나 작업을 뒷단에서 처리하는 모드.
- **Custom**: 내가 원하는 대로 세팅한 맞춤형 AI

1.3 Cursor 주요 기능 : Chat

Chat의 AI Models




















Agent 옆의 “Auto”를 클릭하고 체크를 해제하면 아래와 같이 원하는 AI Model을 선택할 수 있다. Auto로 설정(기본)되어 있을 때에는 성능과 서버 상태 등을 기준으로 가장 최적화 모델을 자동으로 선택해서 사용된다



1.3 Cursor 주요 기능 : Chat

Cursor에서 사용 가능한 주요 모델































[https://docs.cursor.com/en/
models](https://docs.cursor.com/en/models)






































Name ^	Default Context	Max Mode	Capabilities
 Claude 4 Sonnet	200k	-	∞ 
 Claude 4 Sonnet 1M	-	1M	∞ 
 Claude 4.1 Opus	-	200k	∞ 
 Gemini 2.5 Flash	-	1M	∞ 
 Gemini 2.5 Pro	200k	1M	∞ 
 GPT 4.1	200k	1M	∞
 GPT-5 	272k	-	∞ 
 GPT-5 Fast 	272k	-	∞ 
 Grok Code	256k	-	∞ 

Show more models

1.3 Cursor 주요 기능 : Chat

Cursor에서 사용 가능한 모델 전체 목록

Name ^	Default Context	Max Mode	Capabilities
 Claude 3 Opus	60k	-	
 Claude 3.5 Haiku	60k	-	
 Claude 3.5 Sonnet	200k	-	 
 Claude 3.7 Sonnet	200k	-	 
 Claude 4 Opus	-	200k	 
 Claude 4 Sonnet	200k	-	 
 Claude 4 Sonnet 1M	-	1M	 
 Claude 4.1 Opus	-	200k	 
 Cursor Small	60k	-	
 Deepseek R1	60k	-	
 Deepseek R1 (05/28)	60k	-	
 Deepseek V3	60k	-	
 Deepseek V3.1	60k	-	

 Gemini 2.0 Pro (exp)	60k	-	
 Gemini 2.5 Flash	-	1M	 
 Gemini 2.5 Pro	200k	1M	 
 GPT 4.1	200k	1M	
 GPT 4.5 Preview	60k	-	
 GPT-4o	128k	-	 
 GPT-4o mini 	60k	-	
 GPT-5 	272k	-	 
 GPT-5 Fast 	272k	-	 
 GPT-5 Mini	272k	-	 
 GPT-5 Nano	272k	-	 
 Grok 2	60k	-	
 Grok 3 Beta	132k	-	 
 Grok 3 Mini	132k	-	

1.3 Cursor 주요 기능 : Chat

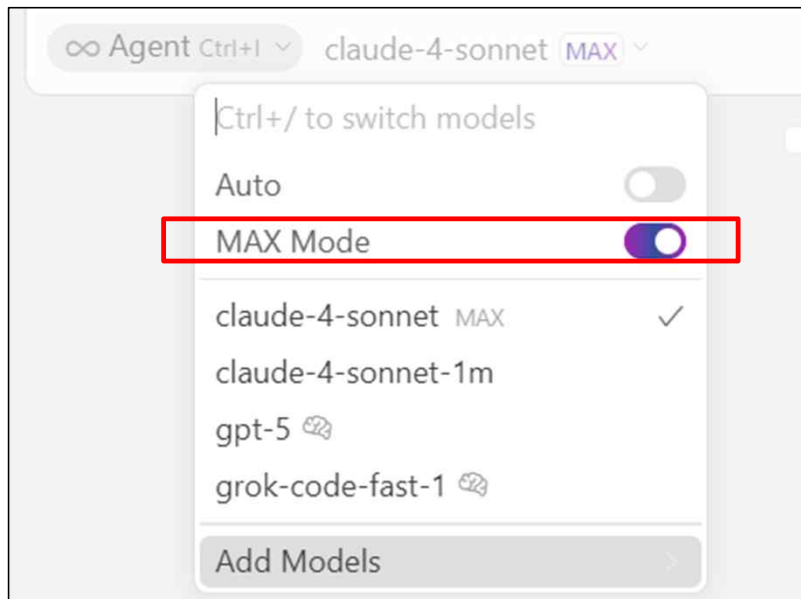
Cursor에서 사용 가능한 모델 전체 목록

 Grok 4	256k	-	∞ 
 Grok Code	256k	-	∞ 
 o1	200k	-	
 o1 Mini	128k	-	
 o3 	200k	-	∞ 
 o3-mini 	200k	-	∞ 
 o4-mini 	200k	-	∞ 

1.3 Cursor 주요 기능 : Chat

Max Mode

일반적으로 커서는 20만 토큰(약 1만 5천 줄의 코드)의 컨텍스트 윈도우를 사용한다. Max 모드는 컨텍스트 윈도우를 모델에서 사용 가능한 최대값으로 확장한다. 이 모드는 다소 느리고 비용이 많이 든다. 컨텍스트 윈도우가 20만 토큰보다 큰 Gemini 2.5 Flash, Gemini 2.5 Pro, GPT 4.1 및 Grok 4에 가장 적합하다.



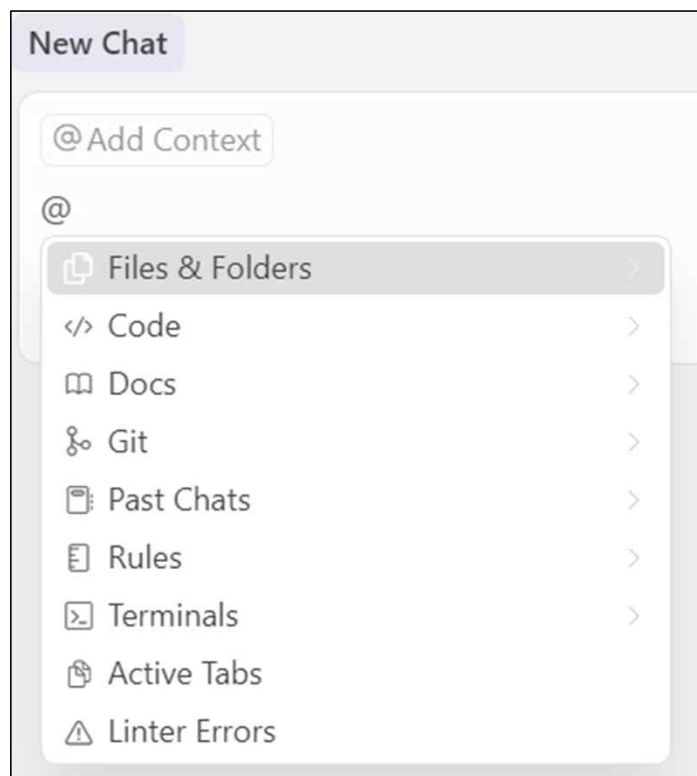
컨텍스트 윈도우(Context windows)

: AI Model이 한번에 처리할 수 있는 텍스트와 코드의 범위. 질문과 생성된 응답을 모두 포함한다

1.3 Cursor 주요 기능 : Chat

@ 기호

프롬프트 입력란에 @ 기호를 사용하여 코드,파일,문서를 참조한다



파일 참조 질의

1.3 Cursor 주요 기능 : Chat

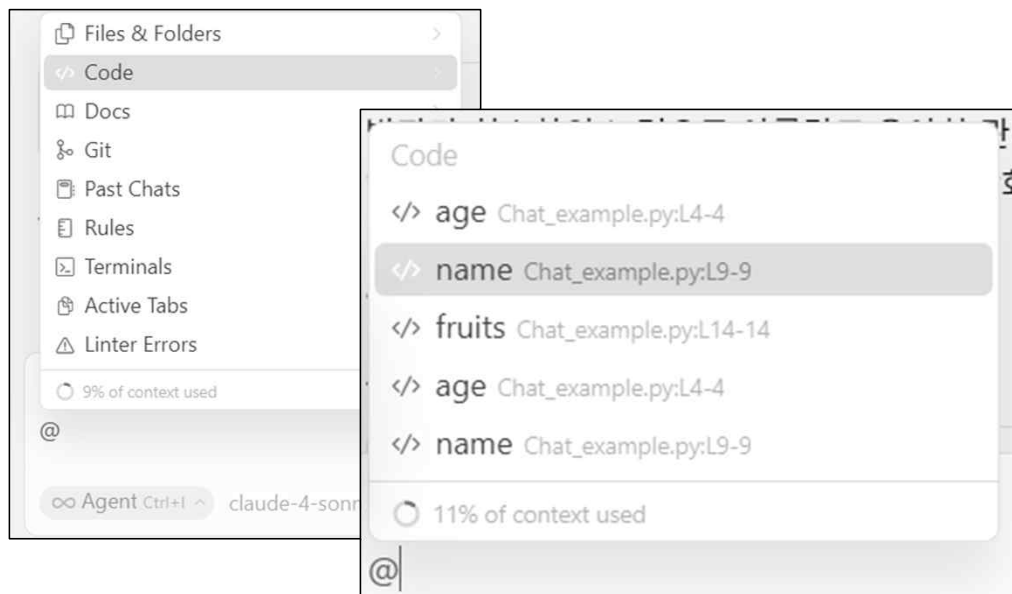
사용가능한 @ 기호

- @Files - 프로젝트의 특정 파일을 참조해
- @Folders - 더 넓은 컨텍스트를 위해 전체 폴더를 참조해
- @Code - 코드베이스의 특정 코드 스니펫이나 심볼을 참조해
- @Docs - 문서와 가이드를 열람해
- @Git - Git 히스토리와 변경 사항에 접근해
- @Past Chats - 요약된 Composer 세션으로 작업해
- @Cursor Rules - Cursor 규칙으로 작업해
- @Web - 외부 웹 리소스와 문서를 참조해
- @Link (paste) - 특정 코드나 문서로 연결되는 링크를 만들어
- @Recent Changes - 최근 변경 사항을 참조해
- @Lint Errors - 린트 오류를 참조해 (**Chat** 전용)
- @Definitions - 심볼 정의를 찾아봐 (**Inline Edit** 전용)
- # Files - 참조 없이 파일을 컨텍스트에 추가해
- / Commands - 열린 파일과 활성 파일을 컨텍스트에 추가해

1.3 Cursor 주요 기능 : Chat

@ 기호

Code 참조 질의 : 소스코드에서 변수/함수 등의 심볼을 참조해서 질의할 수 있다



@name 변수에 대하여 설명해줘

Chat_example.py 파일의 name 변수에 대해 설명해드리겠습니다.

name 변수 상세 분석

변수 선언

python

```
name = "홍길동"
```

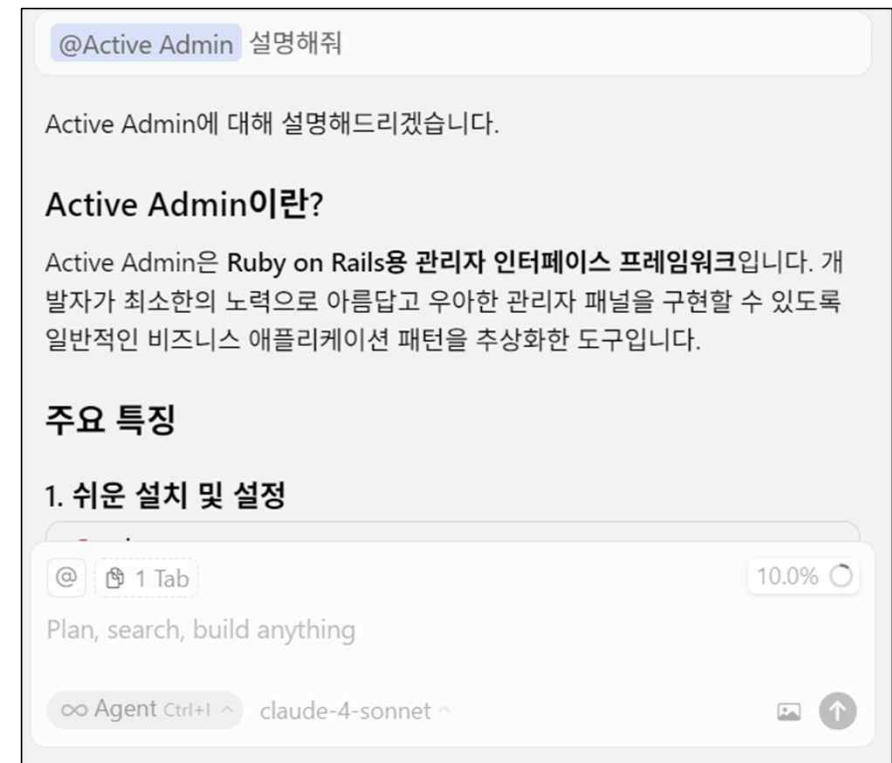
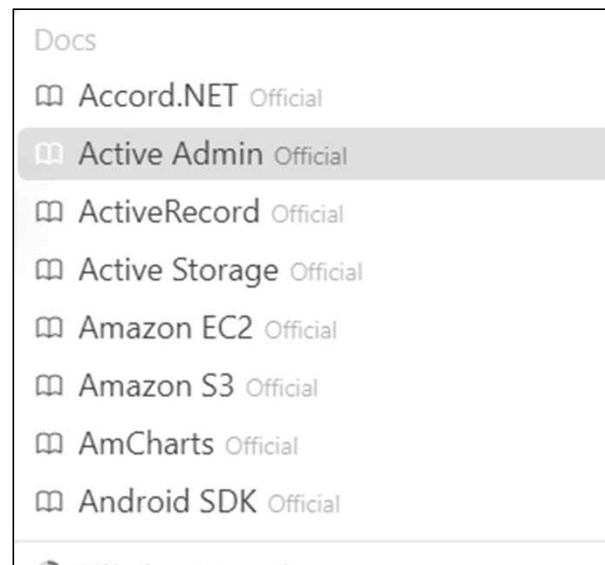
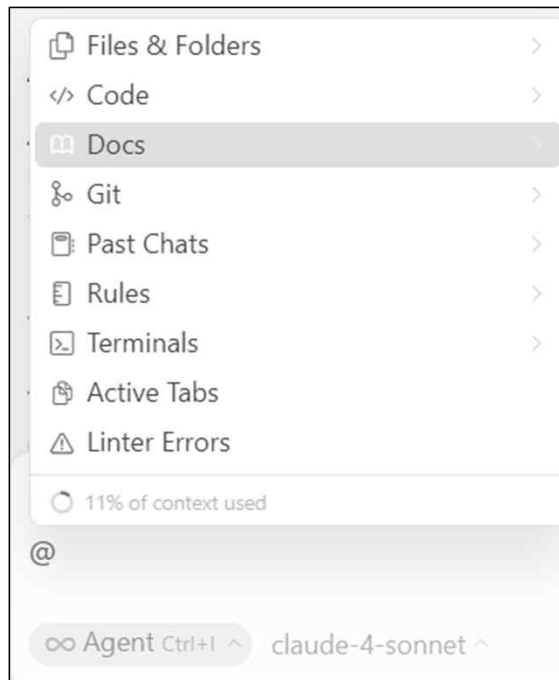
특징

1. 데이터 타입: 문자열(string, str)
2. 값: "홍길동" (한글 문자열)

1.3 Cursor 주요 기능 : Chat

@ 기호

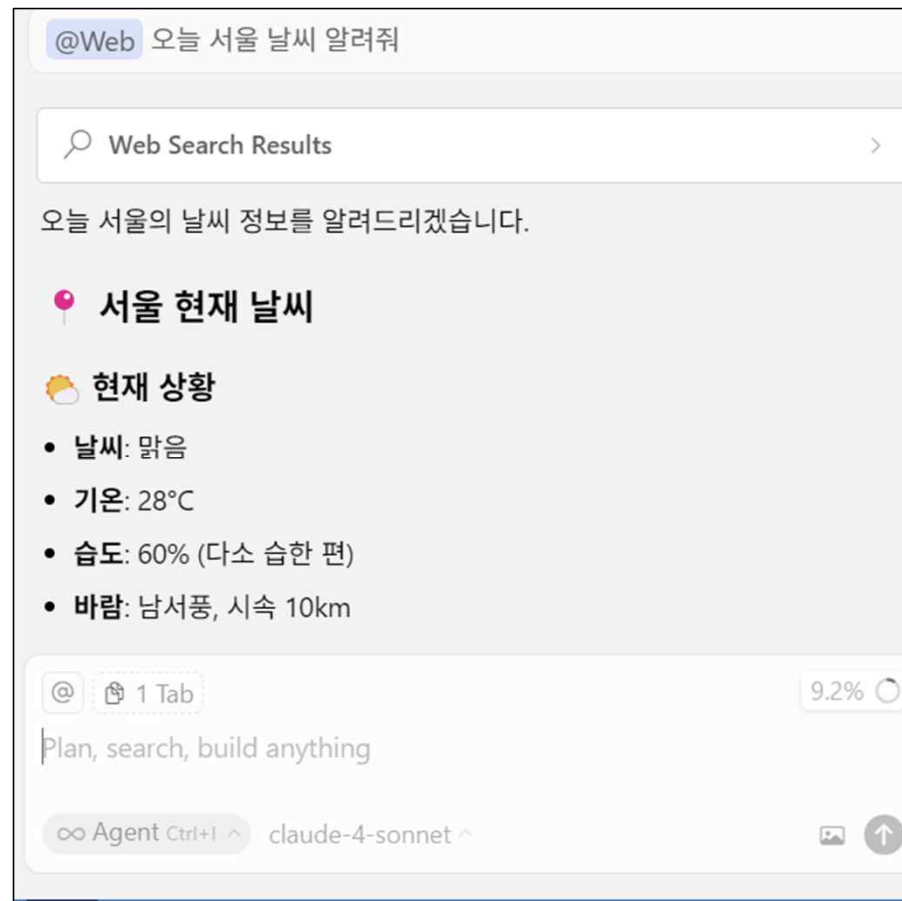
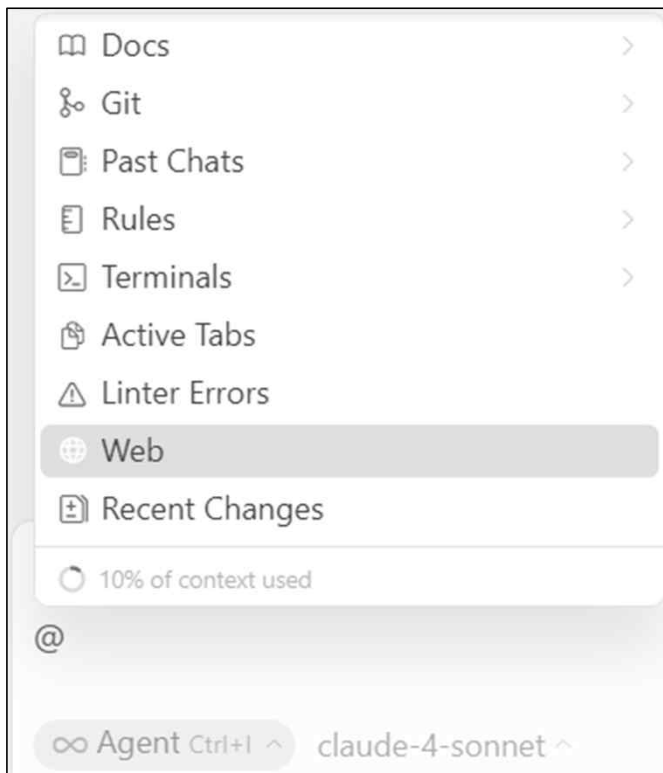
Docs 참조 질의 : Cursor에서 제공되는 공식 문서를 참조해서 질의할 수 있다



1.3 Cursor 주요 기능 : Chat

@ 기호

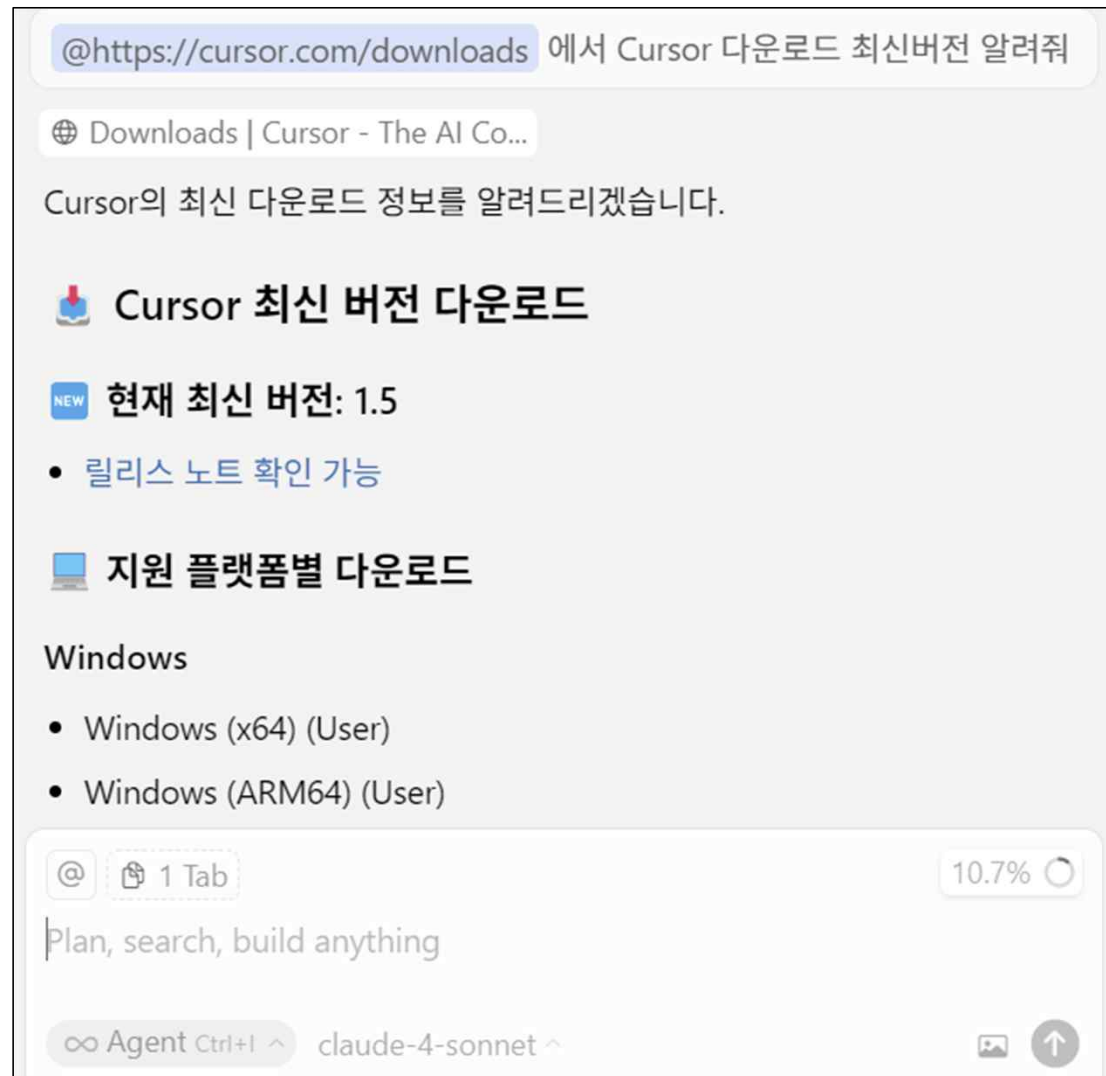
Web 검색 질의 : 웹 검색을 수행하고 그 결과를 반영한다



1.3 Cursor 주요 기능 : Chat

@ 기호

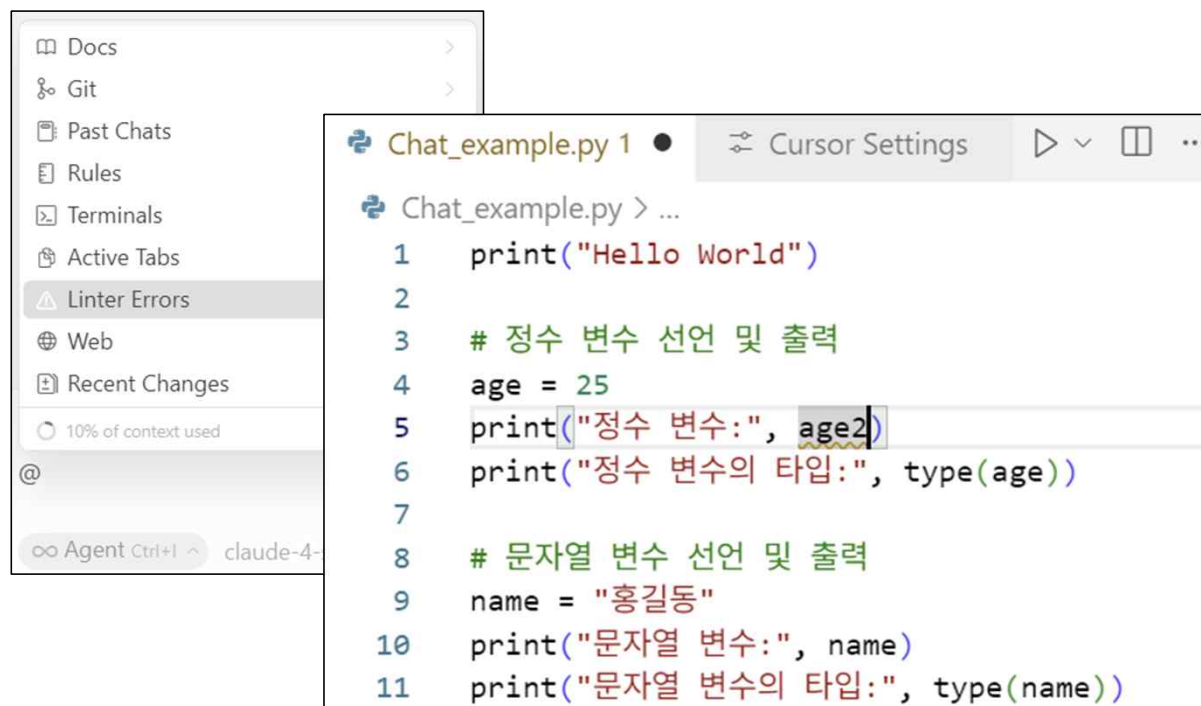
Link 질의 : @에 URL을 직접
입력하면 해당 웹페이지 검색
을 수행하고 그 결과를 반영한
다



1.3 Cursor 주요 기능 : Chat

@ 기호

Lint Errors : 소스 코드에서 Lint 에러(구문 오류)를 참조하여 자동으로 수정하는데 사용



```
1 print("Hello World")
2
3 # 정수 변수 선언 및 출력
4 age = 25
5 print("정수 변수:", age2)
6 print("정수 변수의 타입:", type(age))
7
8 # 문자열 변수 선언 및 출력
9 name = "홍길동"
10 print("문자열 변수:", name)
11 print("문자열 변수의 타입:", type(name))
```

변수 오류 소스 코드



@Linter Errors 수정해줘

현재 워크스페이스의 모든 파일에서 linter 오류를 확인해보겠습니다.

Found 1 error in all files

Linter 오류를 발견했습니다! `chat_example.py` 파일의 5번째 줄에서 정의되지 않은 변수 `age2` 가 사용되고 있습니다.

현재 파일 내용을 확인하고 수정하겠습니다.

Read `Chat_example.py`

문제를 발견했습니다! 5번째 줄에서 `age2` 변수를 사용하고 있지만, 실제로는 4번째 줄에서 선언한 `age` 변수를 사용해야 합니다. 이를 수정하겠습니다.

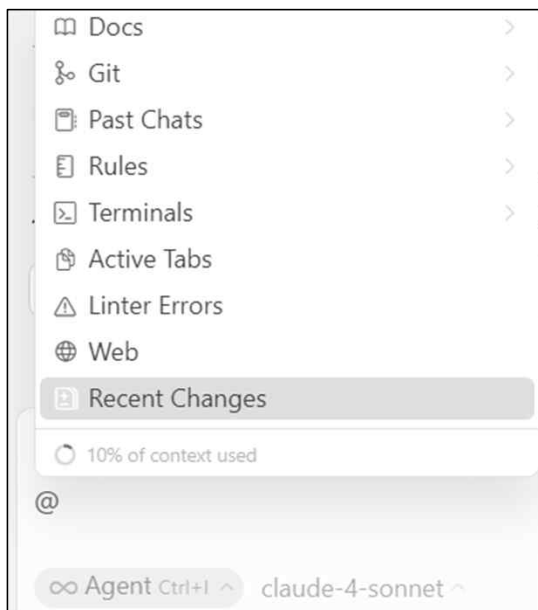
Chat_example.py +1 -1

> 1 File Undo All Keep All

1.3 Cursor 주요 기능 : Chat

@ 기호

Recent Changes 참조 질의 : 최근 변경 사항을 자동으로 추적한다

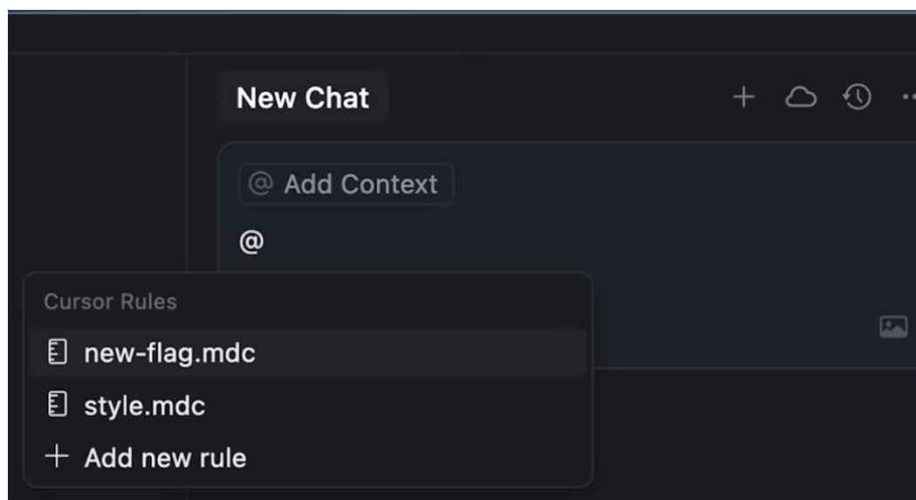


1.3 Cursor 주요 기능 : Chat

@ 기호

Rules 참조 질의 : 설정한 프로젝트 규칙과 가이드라인에 접근할 수 있게 해 주고, 이를 컨텍스트에 명시적으로 적용할 수 있게 해준다.

참조할 Rule 이 미리 설정 등록되어 있어야 가능하다 (Rule 설정은 뒤에서 다룸)

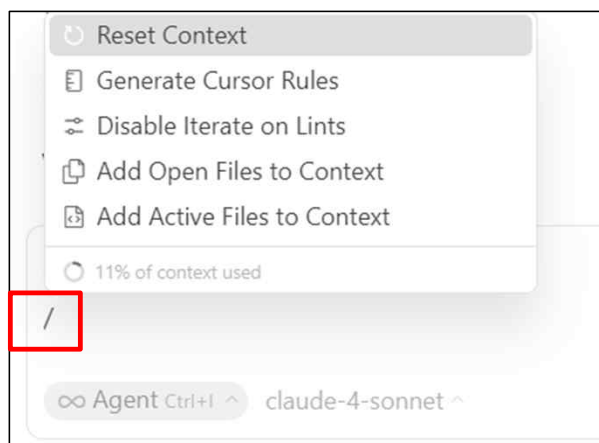


1.3 Cursor 주요 기능 : Chat

/ Command

파일 추가 및 컨텍스트 제어를 위한 빠른 명령

/ 명령은 열려 있는 에디터 탭에 빠르게 접근해서 여러 파일을 컨텍스트로 추가할 수 있게 해준다.



Commands

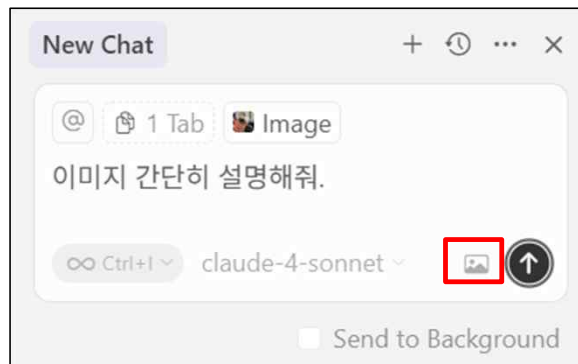
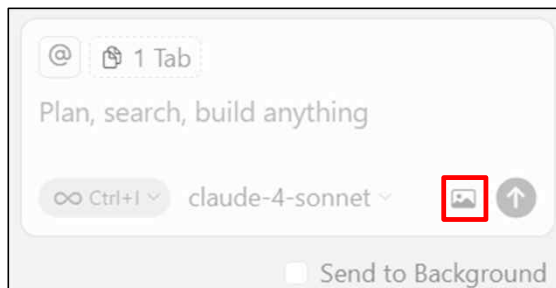
- `/Reset Context` : 컨텍스트를 기본 상태로 초기화
- `/Generate Cursor Rules` : Cursor가 따를 규칙 생성
- `/Disable Iterate on Lints` : 린터 오류와 경고를 수정하려고 시도하지 않음
- `/Add Open Files to Context` : 현재 열려 있는 모든 편집기 탭을 컨텍스트에 추가
- `/Add Active Files to Context` : 현재 화면에 보이는 모든 편집기 탭을 컨텍스트에 추가(분할 레이아웃에서 유용)

<https://docs.cursor.com/ko/context/@-symbols/slash-commands>

1.3 Cursor 주요 기능 : Chat

이미지 버튼

프롬프트에 이미지를 첨부하여 질의 요청 가능



예시
이미지
파일

이미지 간단히 설명해줘.

이 이미지는 NASA 우주비행사의 공식 초상 사진입니다.

주요 특징:

- **인물:** 짧은 금발 머리의 여성 우주비행사
- **복장:** 주황색 NASA 우주복 (ACES, Advanced Crew Escape Suit)
- **장비:** 우주 헬멧을 손에 들고 있음
- **배경:** 미국 국기와 우주왕복선 모형이 배치된 공식적인 NASA 스튜디오 배경
- **패치:** 우주복에 NASA 미션 패치들이 부착되어 있음

이는 우주왕복선 시대(1980년대-2010년대)의 전형적인 NASA 우주비행사 공식 포트로이트 스타일로, 미션 전 홍보나 기록 목적으로 촬영된 것으로 보입니다. 주황색 우주복은 발사와 재진입 시 착용하는 안전복으로, 우주비행사들의 상징적인 의상입니다.



1.3 Cursor 주요 기능 : Chat

Copy 버튼

AI 모델이 생성해준 답변이나 소스 코드를 클립보드에 복사하여 사용할 수 있다

이는 우주왕복선 시대(1980년대-2010년대)의 전형적인 NASA 우주비행사 공식 포트레이트 스타일로, 미션 전 홍보나 기록 목적으로 촬영된 것으로 보입니다. 주황색 우주복은 발사와 재진입 시 착용하는 안전복으로, 우주비행사들의 상징적인 의상입니다.

Copy message



제공해주신 우주비행사 이미지를 출력하는 파이썬 소스를 만들겠습니다. 먼저 이미지 파일을 저장하고, 이를 표시하는 파이썬 스크립트를 작성하겠습니다.

...ge_viewer.py +81 -1 ✓ ... 

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
import requests
import os

def display_astronaut_image():
    """
    우주비행사 이미지를 표시하는 함수
    """
    try:
        # 이미지 파일이 있는지 확인
```

Thank You!!