



Institute of High Energy Physics, CAS

National HEP Data Center

AI编程助手：Claude Code的基本原理和灵活应用



Zhengde ZHANG, on behalf of HepAI Group
Computing Center, Institute of High Energy Physics, CAS

2025.08.29



第三届先进光源中子源科学数据
及软件研讨会
2023年3月

ChatGPT的原理和用于科学的思考

张正德 zdzhang@ihep.ac.cn

中科院高能物理研究所

2023年3月16日



Content

1

自动编程辅助的发展历程

2

Claude Code演示和使用

Chinese Academy of Sciences

3

Cluade Code的原理解析

4

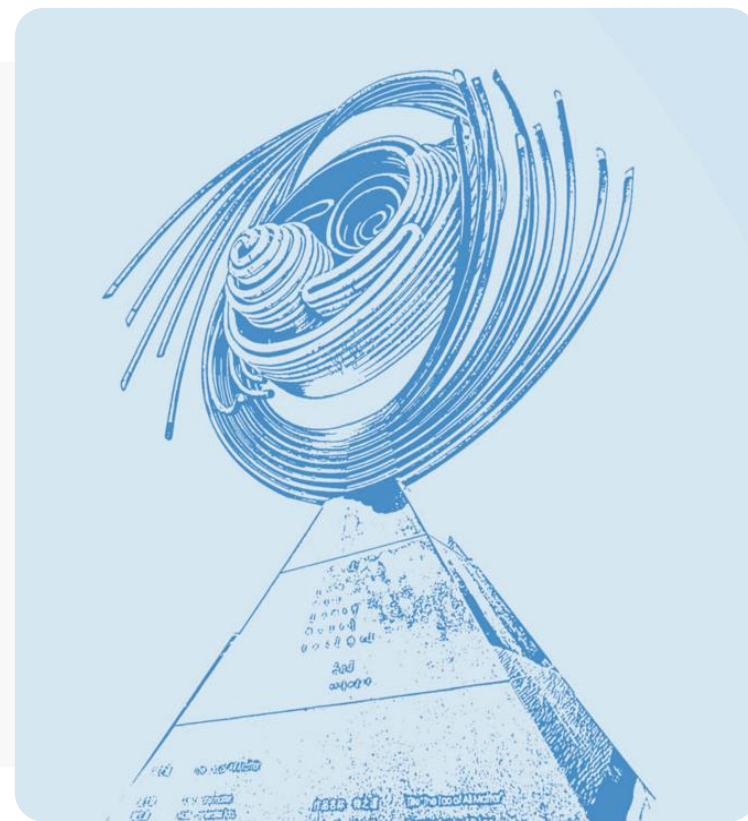
Claude Code的使用技巧

5

总结与展望

Part 1

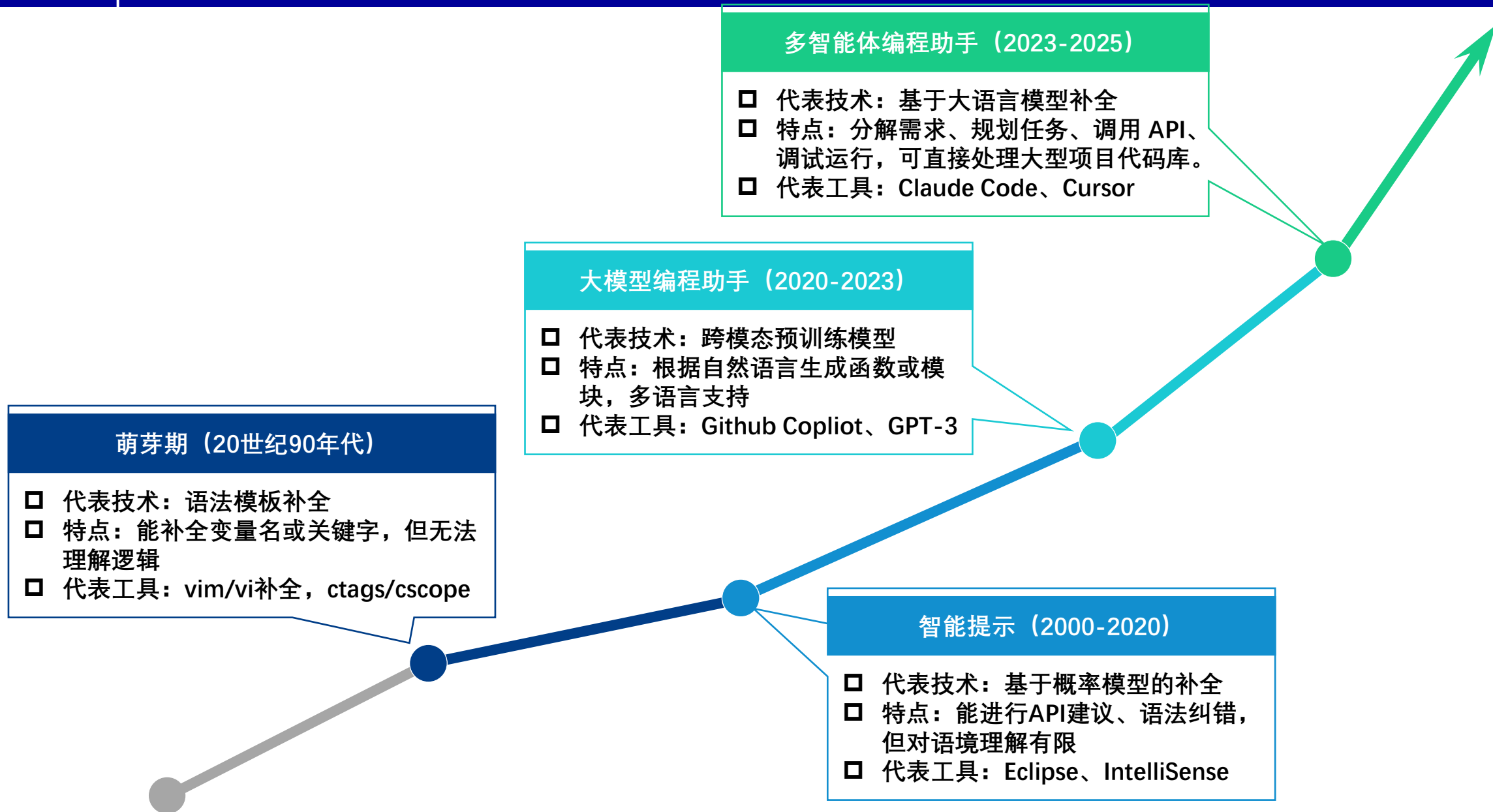
自动编程辅助 的发展历史



自动辅助编程工具的发展历程

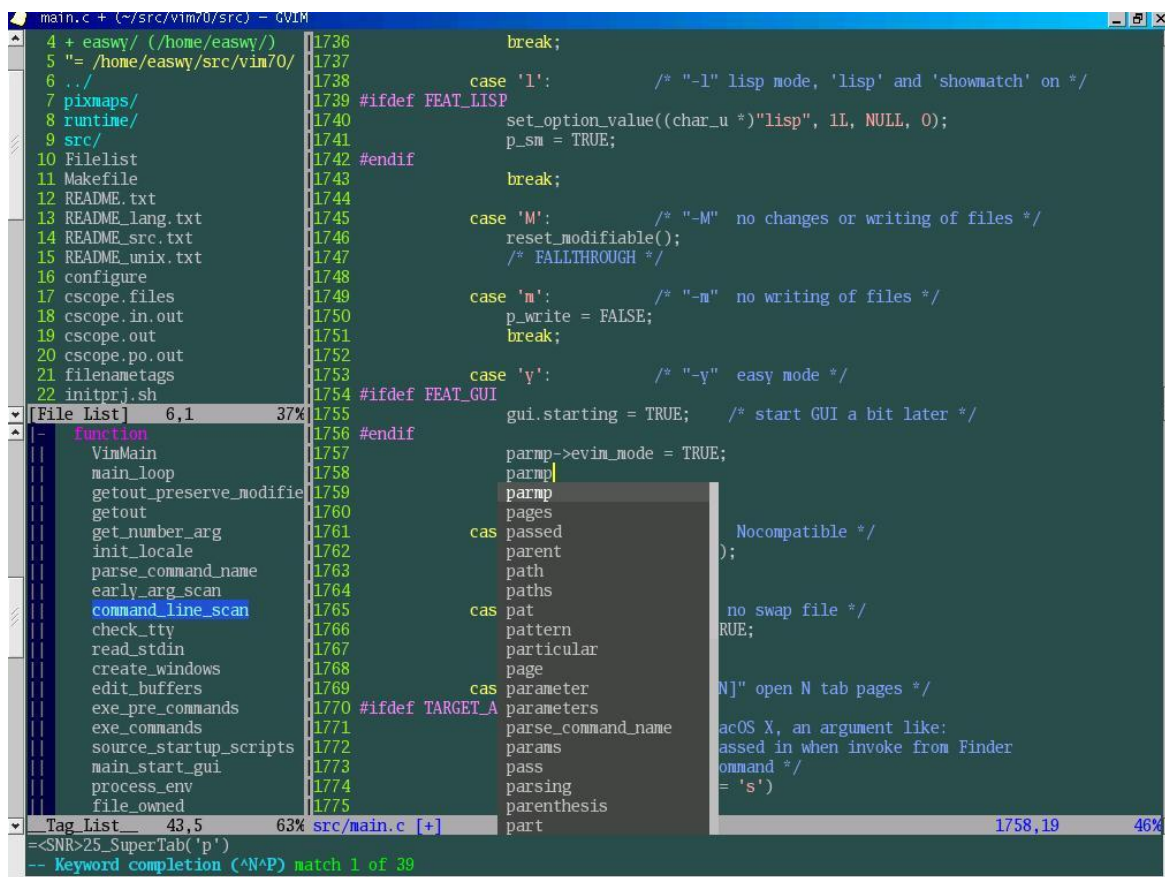


国家高能物理科学数据中心
National HEP Data Center



vim/vi自动补全

<https://www.vim.org/>



```
main.c + (~/.src/vim70/src) - GVIM
4 + easwy/ (/home/easwy/)
5 "= /home/easwy/src/vim70/
6 ../
7 pixmaps/
8 runtime/
9 src/
10 Filelist
11 Makefile
12 README.txt
13 README_lang.txt
14 README_src.txt
15 README_unix.txt
16 configure
17 cscope.files
18 cscope.in.out
19 cscope.out
20 cscope.po.out
21 filenameetags
22 initprj.sh
[File List] 6,1 37%
function
  VinMain
  main_loop
  getout_preserve_modifie
  getout
  get_number_arg
  init_locale
  parse_command_name
  early_arg_scan
  command_line_scan
  check_tty
  read_stdin
  create_windows
  edit_buffers
  exe_pre_commands
  exe_commands
  source_startup_scripts
  main_start_gui
  process_env
  file_owned
1736 break;
1737
1738 case 'l': /* "-l" lisp mode, 'lisp' and 'showmatch' on */
1739 #ifdef FEAT_LISP
1740 set_option_value((char_u *)"lisp", 1L, NULL, 0);
1741 p_sm = TRUE;
1742 #endif
1743 break;
1744
1745 case 'M': /* "-M" no changes or writing of files */
1746 reset_modifiable();
1747 /* FALLTHROUGH */
1748
1749 case 'm': /* "-m" no writing of files */
1750 p_write = FALSE;
1751 break;
1752
1753 case 'y': /* "-y" easy mode */
1754 #ifdef FEAT_GUI
1755 gui.starting = TRUE; /* start GUI a bit later */
1756 #endif
1757 parmp->vim_mode = TRUE;
1758 parmp
1759 parmp
1760 pages
1761 passed Nocompatible */
1762 parent );
1763 path
1764 paths
1765 pat no swap file */
1766 pattern TRUE;
1767 particular
1768 page
1769 parameter N]" open N tab pages */
1770 #ifdef TARGET_A parameters
1771 parse_command_name MacOS X, an argument like:
1772 params passed in when invoke from Finder
1773 pass command */
1774 parsing = 's')
1775 parenthesis
1776 part
```

特点:

- ◆ 依赖硬编码规则、固定的语法模板和简单的模式匹配
- ◆ 能在 IDE 或命令行中根据上下文补全变量名或关键字
- ◆ 无法理解代码逻辑

IntelliSense

Microsoft 为提升代码编写效率设计的一套智能代码编辑辅助工具。

- <https://code.visualstudio.com/docs/editing/intellisense> , 1995年

```
JS server.js > ...  
1  const express = require('express');  
2  const app = express();  
3  
4
```

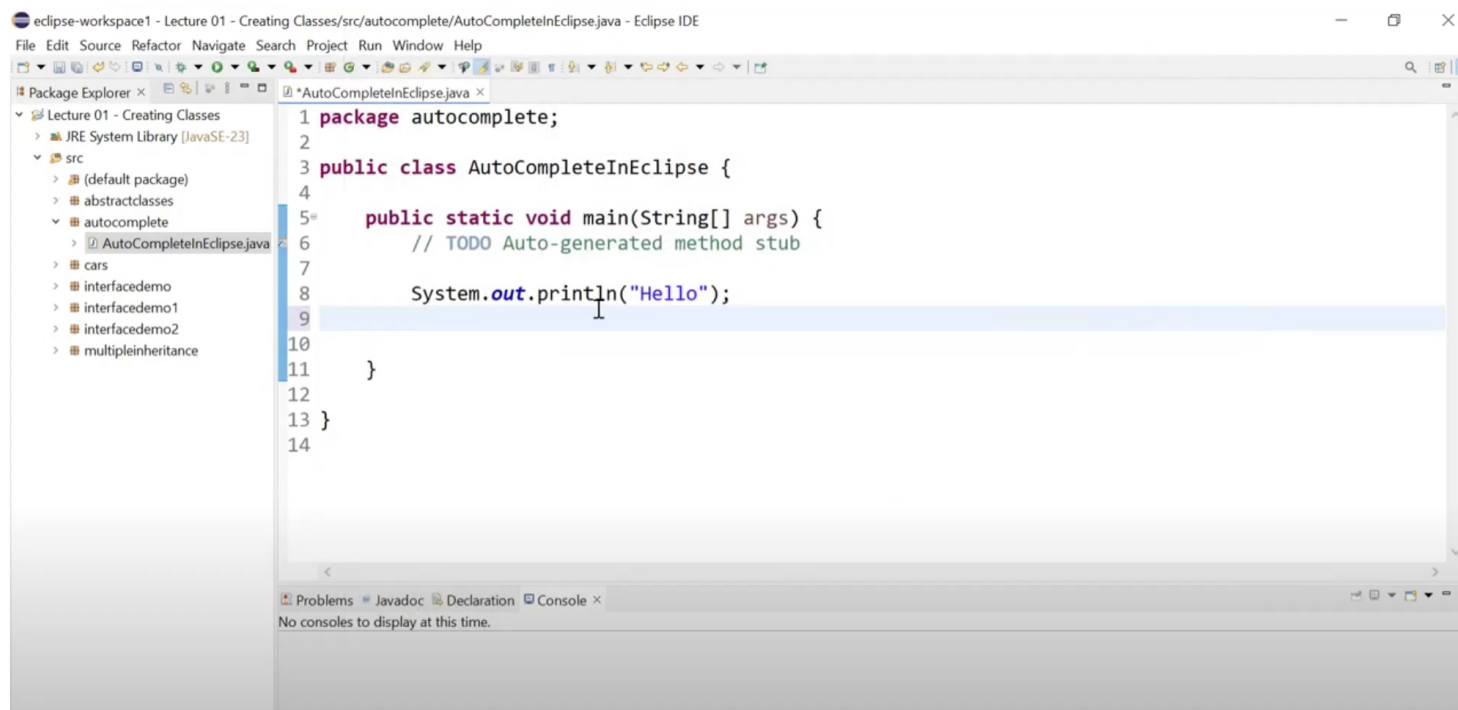
I

特点:

- ◆ 代码补全 (code completion)
- ◆ 成员列表 (List Members)
- ◆ 参数提示 (Parameter Info)
- ◆ 快速信息 (Quick Info)
- ◆ 完整单词补全 (Complete Word)

Eclipse

<https://eclipseide.org/>



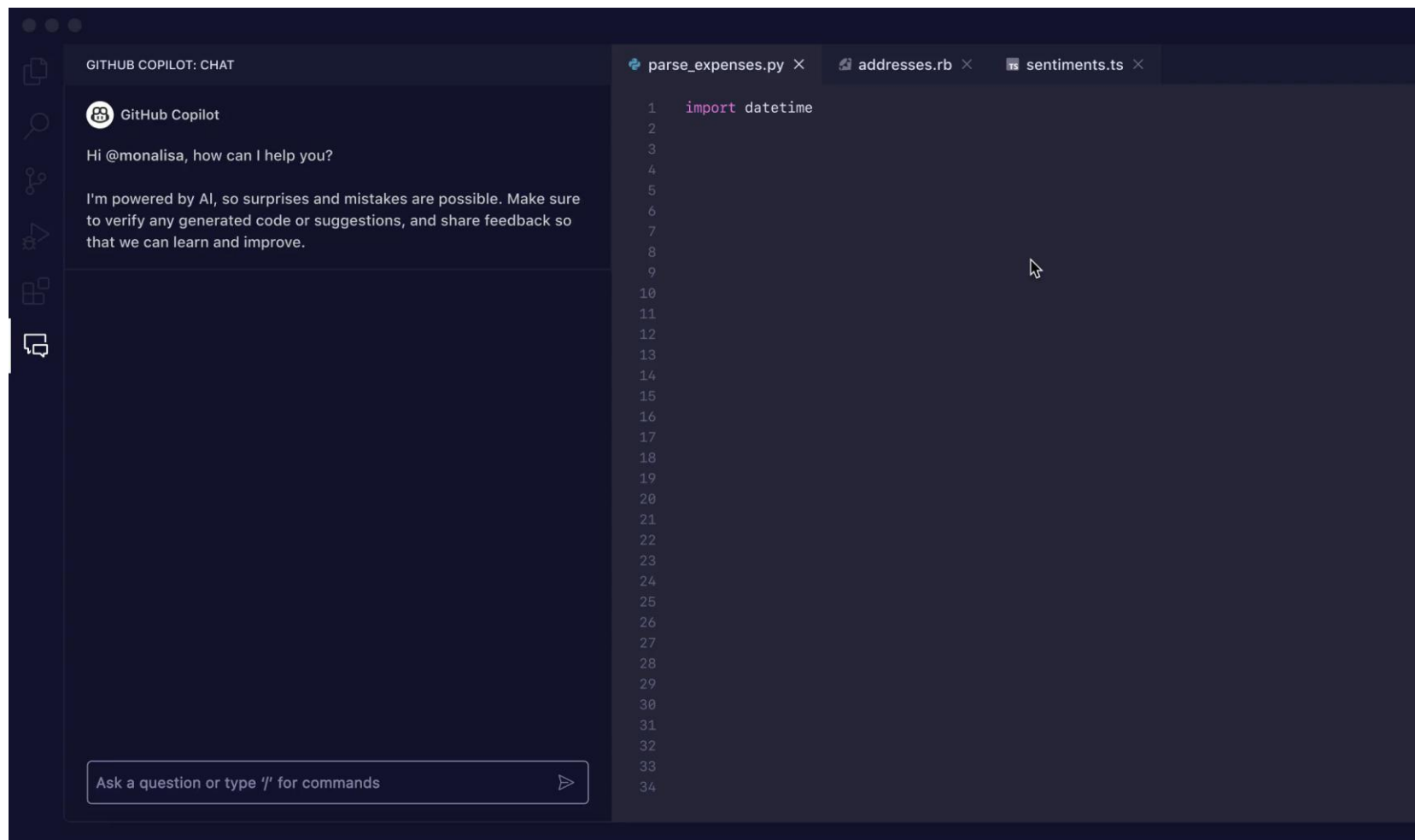
2001年, IBM
特点:

- ◆ 智能补全、API 建议、语法纠错等
- ◆ 通过插件扩展功能
- ◆ 源自Java开发, 支持多种语言

Github Copilot

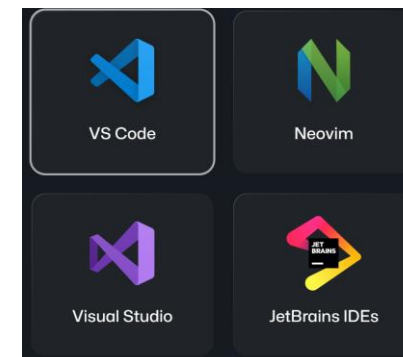
<https://github.com/features/copilot>

Github提供的世界上最广泛使用的人工智能开发工具，2021年



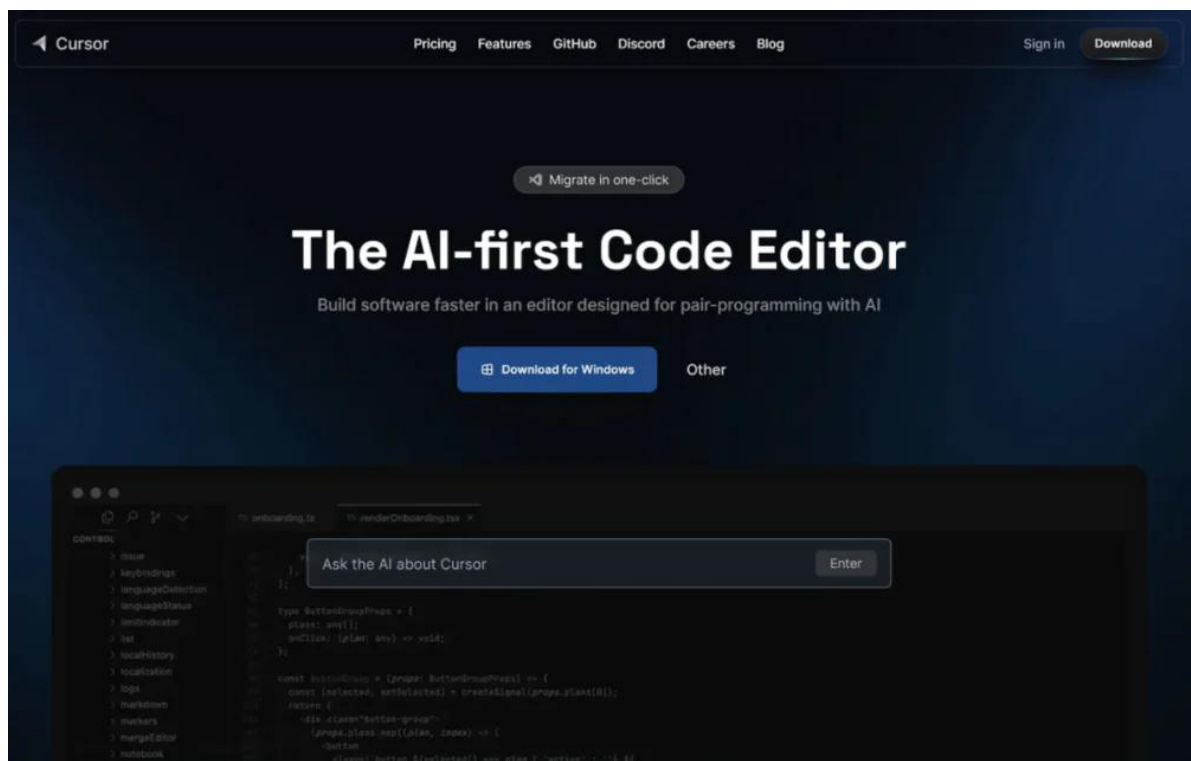
特点:

- 支持多种开发环境
- 支持多种语言
- 极大提升开发效率



Cursor

颠覆程序员工作流的初代AI IDE，2025年



>|

相比单纯依赖 Copilot 或 ChatGPT，Cursor 更像一个深度集成了 AI 的 IDE，把“写代码”这件事，真正带到了**对话式 + 自动化**的新阶段

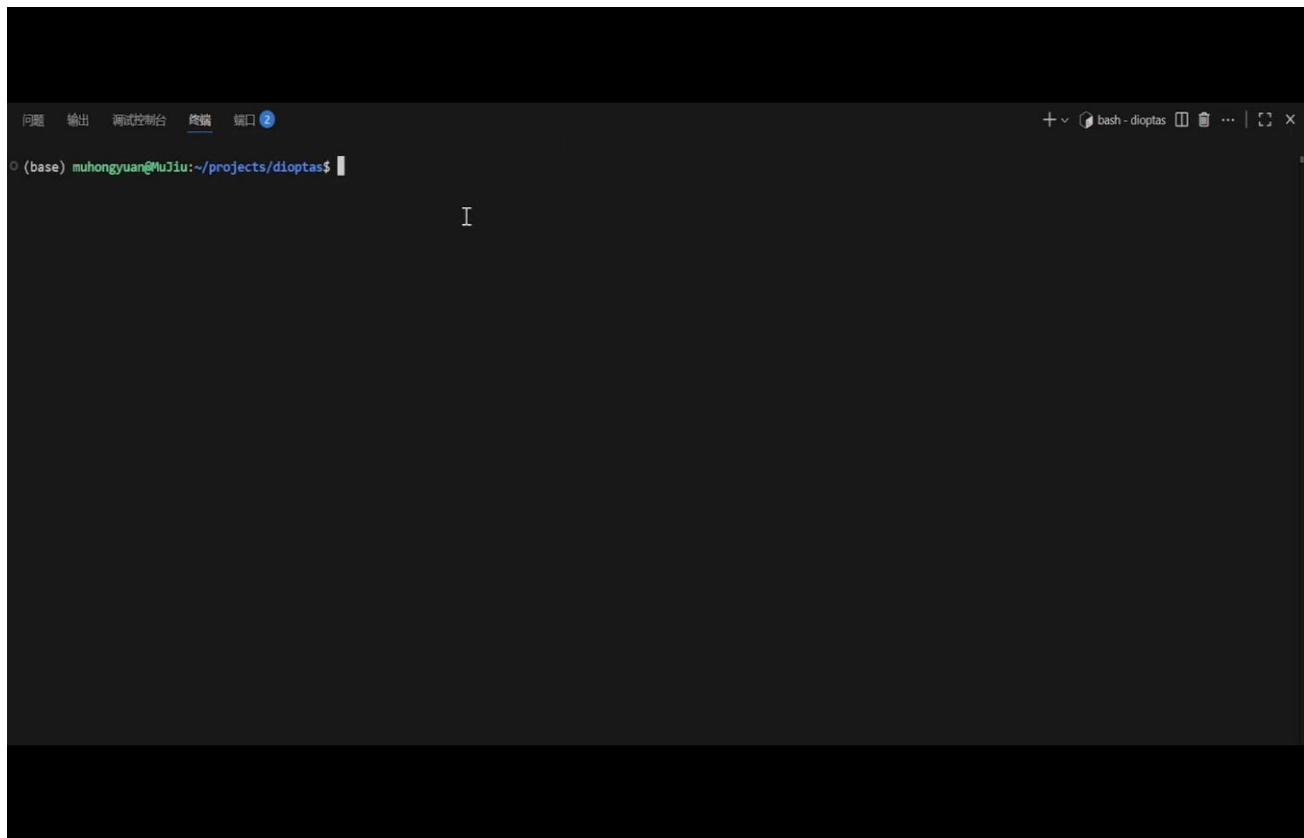
核心亮点：

- 内置 GPT-4、Claude、Gemini 等多种模型
- 编辑器内对话，不用来回切 ChatGPT
- 理解整个项目上下文，而不仅仅是单个文件

Gemini CLI

<https://github.com/google-gemini/gemini-cli>

Google提供的，接入其旗舰模型gemini-2.5-pro的终端工具，2025年



为什么选择 Gemini CLI?

-  免费套餐：60 个请求/分钟和 1,000 个请求/天（个人 Google 帐户）
-  强大的 Gemini 2.5 Pro：访问 1M 令牌上下文窗口
-  内置工具：Google 搜索接地、文件作、shell 命令、网络获取
-  可扩展：MCP（模型上下文协议）支持自定义集成
-  终端优先：专为使用命令行的开发人员而设计
-  开源：Apache 2.0 许可

Claude Code是什么?



国家高能物理科学数据中心
National HEP Data Center

Claude Code

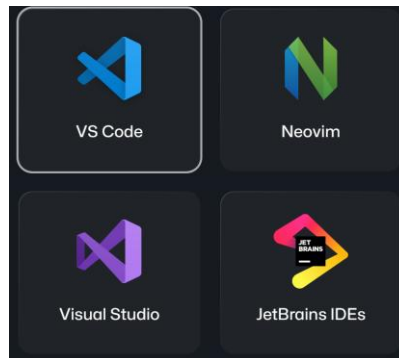
Anthropic公司2025年5月正式发布的AI代码助手，专注于为开发者提供智能化的代码生成和优化助力，提升软件开发效率与质量。

The screenshot shows the Claude website with the following content:

- Header: Claude logo, navigation links (Features 特性, Pricing 定价, FAQ), and a "Get started 开始使用" button.
- Section: "Meet Claude 遇见克劳德"
- Text: "Claude is a next generation AI assistant built by Anthropic and trained to be safe, accurate, and secure to help you do your best work." and "Claude 是由 Anthropic 开发的新一代 AI 助手，经过训练以确保安全、准确和可靠，以帮助你完成最佳工作。"
- Section: "Create with Claude 使用 Claude 创作"
- Text: "Draft and iterate on websites, graphics, documents, and code alongside your chat with Artifacts." and "与 Artifacts 一起起草和迭代网站、图形、文档和代码。"
- Section: "Bring your knowledge 带来你的知识"
- Text: "Share and collaborate with your team" and "与你的团队分享和协作"
- Image: A screenshot of the Claude interface showing a chat conversation about creating a scatter plot, with a corresponding scatter plot visualization.

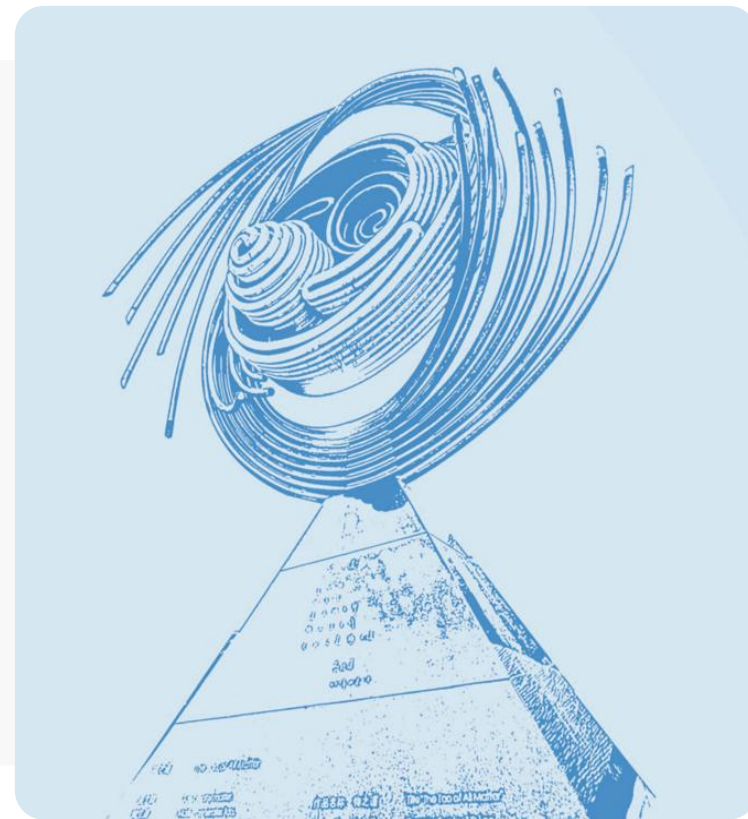
特点:

- 多语言代码生成
- 支持多种开发环境
- 代码优化与重构
- 支持超长代码上下文



Part 2

Claude Code 的演示与使用



Claude Code演示



国家高能物理科学数据中心
National HEP Data Center

例1：视频展示：
帮我开发一个
TODOLIST工具，
使用VUE3 +
Tailwindcss，页面
设计精美一些。

```
(base) yangjian@RockYang ~ -/test mkdir dodolist
(base) yangjian@RockYang ~ -/test cd dodolist
(base) yangjian@RockYang ~ -/test/dodolist claude
```

我就是极客学长 bilibili

Do you trust the files in this folder?

/Users/yangjian/test/dodolist

Claude Code may read files in this folder. Reading untrusted files may lead Claude Code to behave in unexpected ways.

With your permission Claude Code may execute files in this folder. Executing untrusted code is unsafe.

<https://docs.anthropic.com/en/claude-code-security>

1. Yes, proceed
2. No, exit

Enter to confirm · Esc to exit

• Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/yangjian/test/dodolist

Overrides (via env):

- API Key: sk-ant-...R2i5ganOVHvGMOa8og3fkR6kCzT9
- API Base URL: <https://api.geekai.me>

Tips for getting started:

1. Ask Claude to create a new app or clone a repository
2. Use Claude to help with file analysis, editing, bash commands and git
3. Be as specific as you would with another engineer for the best results
4. ✓ Run /terminal-setup to set up terminal integration

• Tip: Use /permissions to pre-approve and pre-deny bash, edit, and MCP tools

> 帮我开发一个 TODOLIST 工具。 █

plan mode on (shift+tab to cycle)

✗ Auto-update failed · Try claude doctor or npm i -g @anthropic-ai/claude-code

例2：

<https://aiapi.ihep.ac.cn>

HaiDDF模型服务的前端界面，5天从零到上线。

Claude Code能做什么?



国家高能物理科学数据中心
National HEP Data Center

快速原型开发与功能验证

开发周期2周→3天

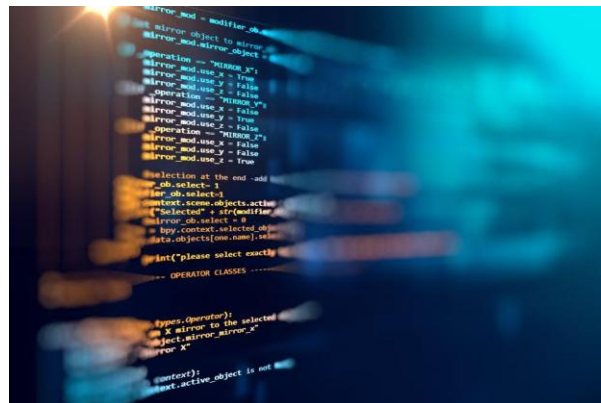
某初创公司计划开发一款生鲜电商APP，传统原型开发需前端、后端、产品团队协作，预估周期2周。

产品经理用自然语言描述功能流程

基于方案，Claude Code并行生成前后端仅3天完成可运行原型。

演示中发现问题，开发者告知 Claude Code，10分钟内完成代码修改与更新。

原型开发周期从2周缩短至3天，效率提升80%，成功用于融资演示并获得首轮投资。



Claude Code能做什么？



国家高能物理科学数据中心
National HEP Data Center

复杂实现与逻辑梳理

场景描述

01

某金融机构遗留Java项目，代码老旧，需进行全面重构优化，传统评估需2个月人力投入。

重构实施

02

Claude Code快速扫描整个代码库，生成详细问题报告与重构优先级建议，1周内完成全面分析，重构实施。

性能调优

03

针对性能瓶颈，Claude Code通过优化数据库查询语句、调整资源加载策略等方式，显著提升系统响应速度。

成果展示

04

重构项目在1个月内完成，节省25%开发周期。系统性能大幅提升。

Claude Code能做什么?



国家高能物理科学数据中心
National HEP Data Center

8

编程学习和技术问题解决



场景描述

大学生小王学习Python时遇到“链表反转”问题，课本示例代码抽象难懂，学习进度受阻。



Claude Code介绍

小王提问“如何用迭代法反转单链表？”，Claude Code先输出可视化步骤图，再逐行注释代码。



问题诊断

小王写出代码后报错，上传代码后，Claude Code定位到错误，提供修复方案并解释。



练习与反馈

Claude Code生成3道变式练习，小王提交答案后，即时指出存在的错误并给出优化建议。

结果

小王1小时内理解链表反转逻辑，成功独立完成作业，较预期学习时间缩短60%。

01

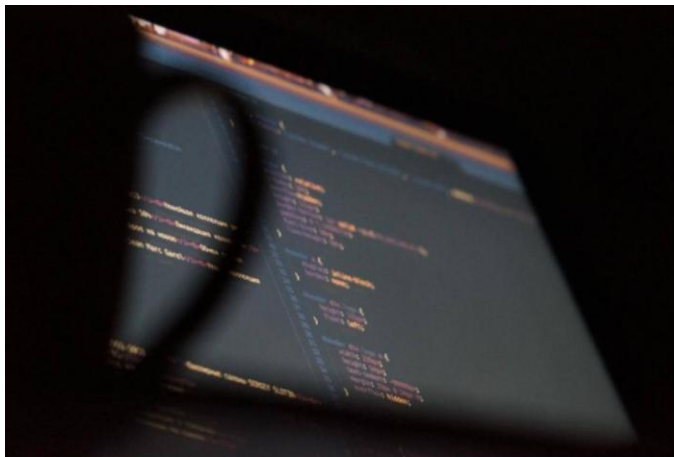
02

03

04

Claude Code能做什么？

团队开发中的代码审查辅助



场景描述

某电商平台技术团队进行月度代码审查，涉及多个核心模块的代码提交，需检查代码存在不足，传统人工审查需耗时2天，易遗漏隐性问题。

Claude Code介入

团队将代码提交记录同步给Claude Code,10分钟内完成全量扫描，生成结构化审查报告。

修复建议与示例：

针对“数据库循环查询”问题，Claude Code提供具体优化代码，团队只需进行调试。

成果展示

代码审查时间从2天缩短至0.5天，效率提升75%，发现隐性问题数量增加40%。团队协作效率显著提升。

Claude Code的安装使用



国家高能物理科学数据中心
National HEP Data Center

1. 安装Claude-code

在终端中运行 `npm install -g @anthropic-ai/claude-code`

```
(base) muhongyuan@MuJiu:~/projects/dioptas$ npm install -g @anthropic-ai/claude-code  
  
added 3 packages in 808ms  
  
2 packages are looking for funding  
run `npm fund` for details
```

2. 配置HepAI DDF 信息

Api-key见: <https://aiapi.ihep.ac.cn/>

我的API-KEY

使用步骤: (1) 复制; (2) 根据接口文档调用。注意base_url地址: <https://aiapi.ihep.ac.cn/apiv2> 复制base_urlapi_key

序号	应用名称	API密钥	过期时间	备注	操作
1				-	复制 编辑 删除
2				-	复制 编辑 删除

+ 添加密钥

若已有apikey则点击复制

没有则创建并复制

2.配置HepAI DDF 信息

编辑~/.bashrc 或 ~/.zshrc配置文件，将apikey和baseurl添加进去

可在终端输入code ~/.bashrc打开文件进行编辑

在文件中粘贴

```
export ANTHROPIC_BASE_URL="https://aiapi.ihep.ac.cn/apiv2/anthropic"  
export ANTHROPIC_API_KEY="sk-*****" # 刚才复制的HepAI的 apikey
```

Ctrl+s保存

在终端输入**source ~/.bashrc** # or source ~/.zshrc进行

在终端输入**echo \$ANTHROPIC_BASE_URL**进行测试

```
● (base) muhongyuan@MuJiu:~/projects/dioptas$ echo $ANTHROPIC_BASE_URL  
https://aiapi.ihep.ac.cn/apiv2/anthropic
```

终端可以正确输出配置的信息，配置成功

3.解除限制

在终端输入

```
bash -c "$(curl -fsSL https://aiapi.ihep.ac.cn/apiv2/files/file-387a4a06a89d4989adeb2a40c626f7ba/preview)"
```

终端弹出提示，再次粘贴apikey,此处不会显示你的输入，确保正确粘贴后，ENTER键确认

```
(dioplas) muhongyuan@MuJiu:~/projects/dioplas$ bash -c "$(curl -fsSL https://aiapi.ihep.ac.cn/apiv2/files/file-387a4a06a89d4989adeb2a40c626f7ba/preview)"
Node.js is already installed: v22.18.0
Claude Code is already installed: 1.0.88 (Claude Code)
Configuring Claude Code to skip onboarding...
🔑 Please enter your HepAI API key:
You can get your API key from: https://aiapi.ihep.ac.cn/
Note: The input is hidden for security. Please paste your API key directly.
```

终端出现此提示代表Claude code可以使用了

```
📄 Adding environment variables to /home/muhongyuan/.bashrc...
⚠️ Environment variables already exist in /home/muhongyuan/.bashrc. Skipping...

🎉 Installation completed successfully!

🔄 Please restart your terminal or run:
source /home/muhongyuan/.bashrc

🚀 Then you can start using Claude Code with:
claude
```

Claude Code的安装使用



国家高能物理科学数据中心
National HEP Data Center

4.启动Claude-code

可以在终端输入`claude`进行启动

首次使用可能出现下面提示，在终端按1然后ENTER键进行确认

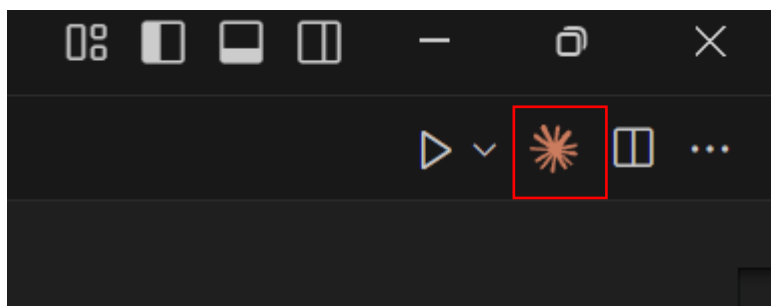
```
(drsai) [xionddb@hepai-4090 software]$ claude

Detected a custom API key in your environment
ANTHROPIC_API_KEY: sk-ant-...lThRQLBmFiriHEcpegFN
Do you want to use this API key?
> 1. Yes
   2. No (recommended)✓

Enter to confirm . Esc to cancel
```

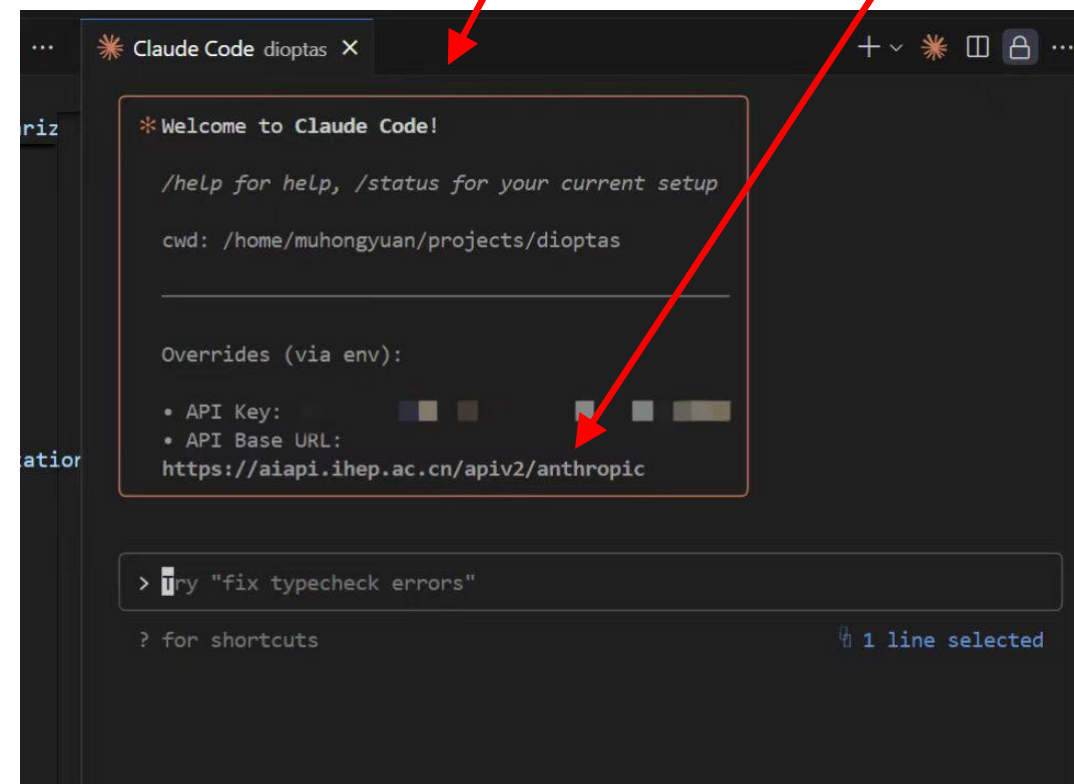
选 "Yes"
使用自定义Key

或点击右上角claude的logo 一键启动



Claude Code启动界面

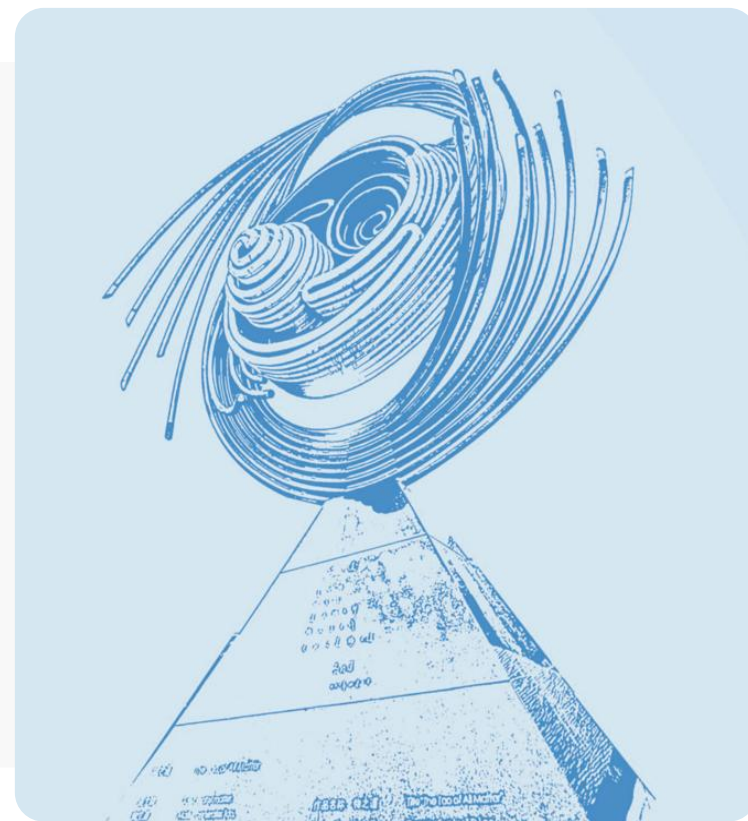
自定义base_url和
自定义api-key



输入`hello`测试连通性

Part 3

Claude Code 的原理解析

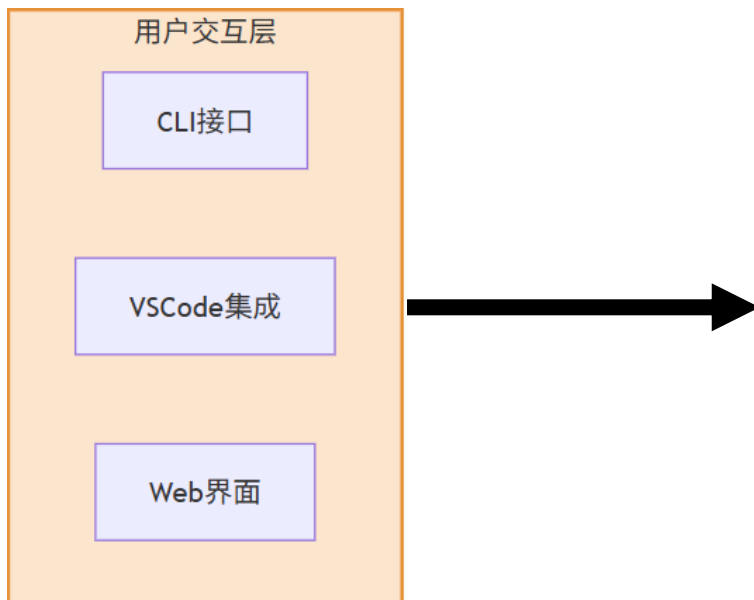
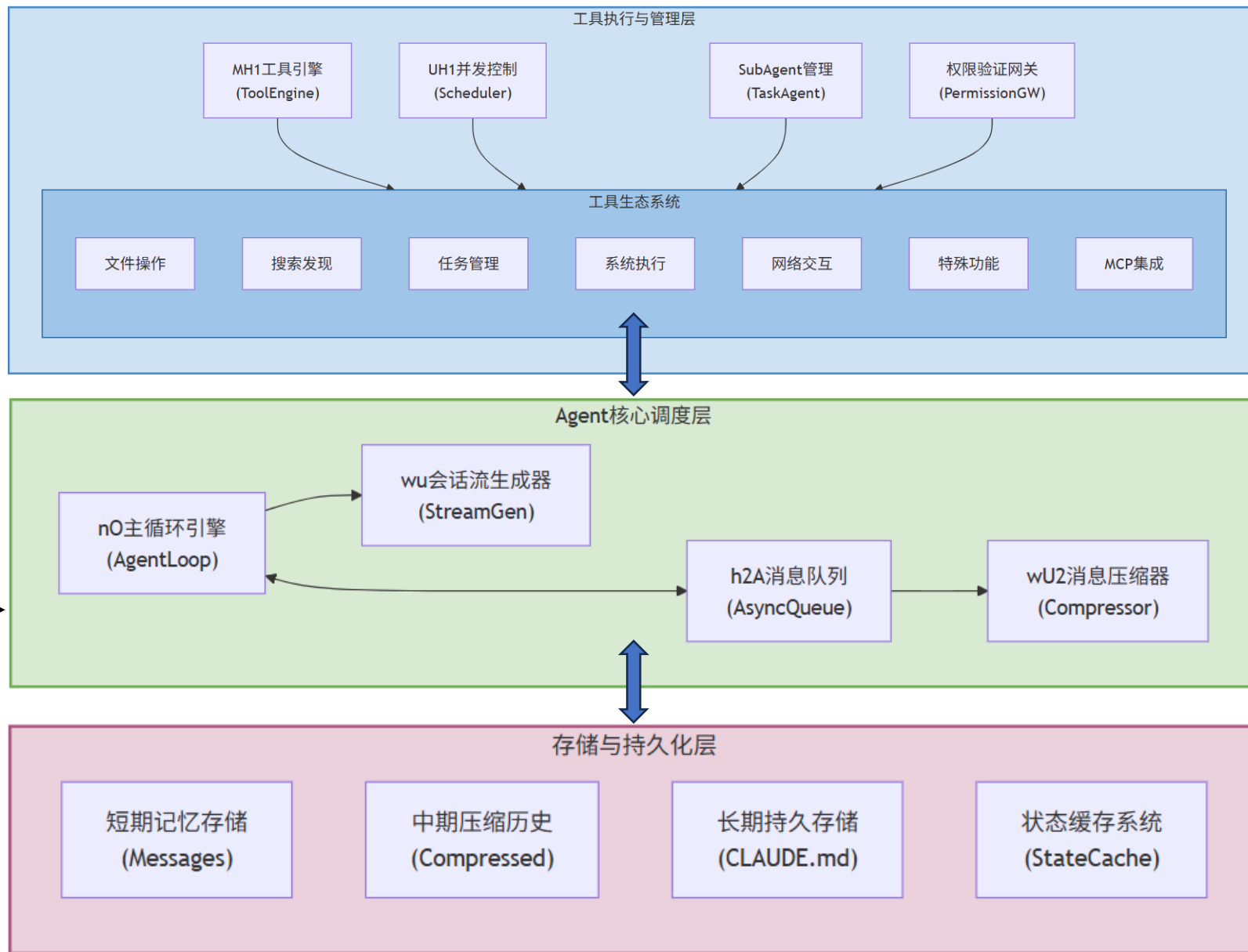


Claude Code的实现方案——AI智能体

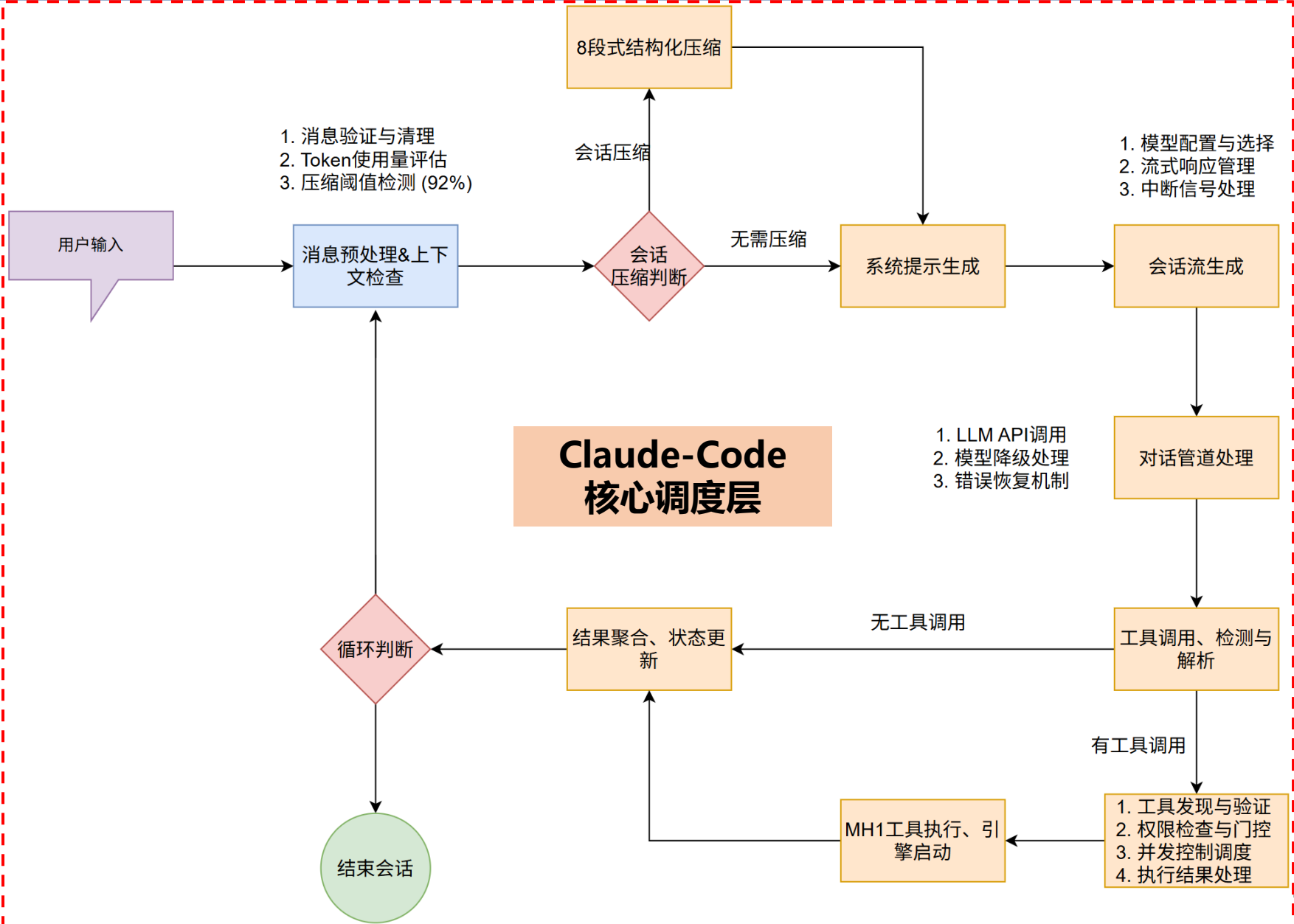
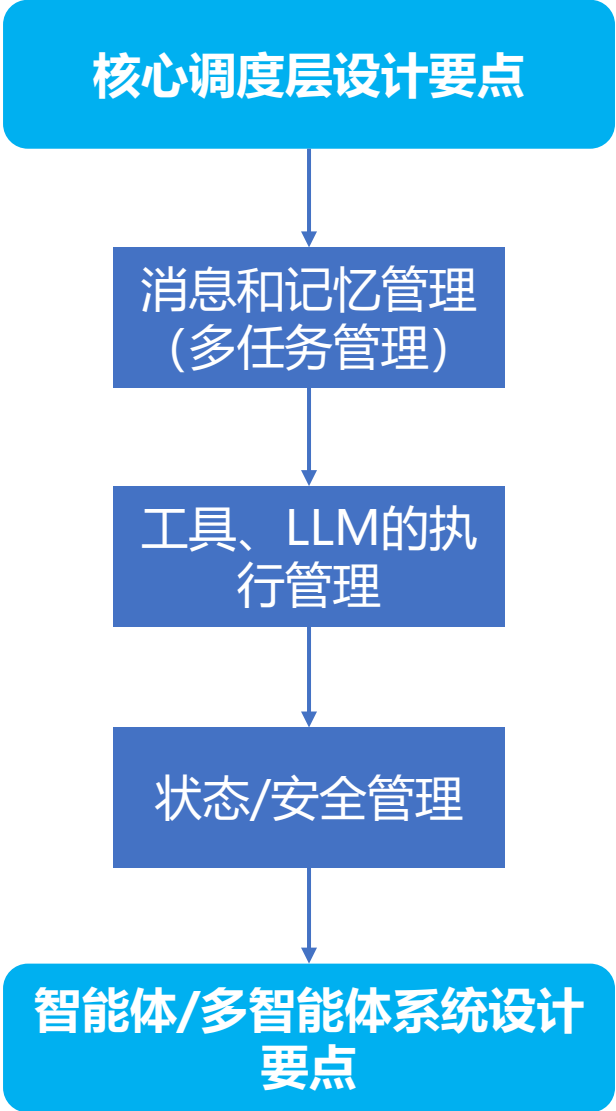


Claude Code智能体

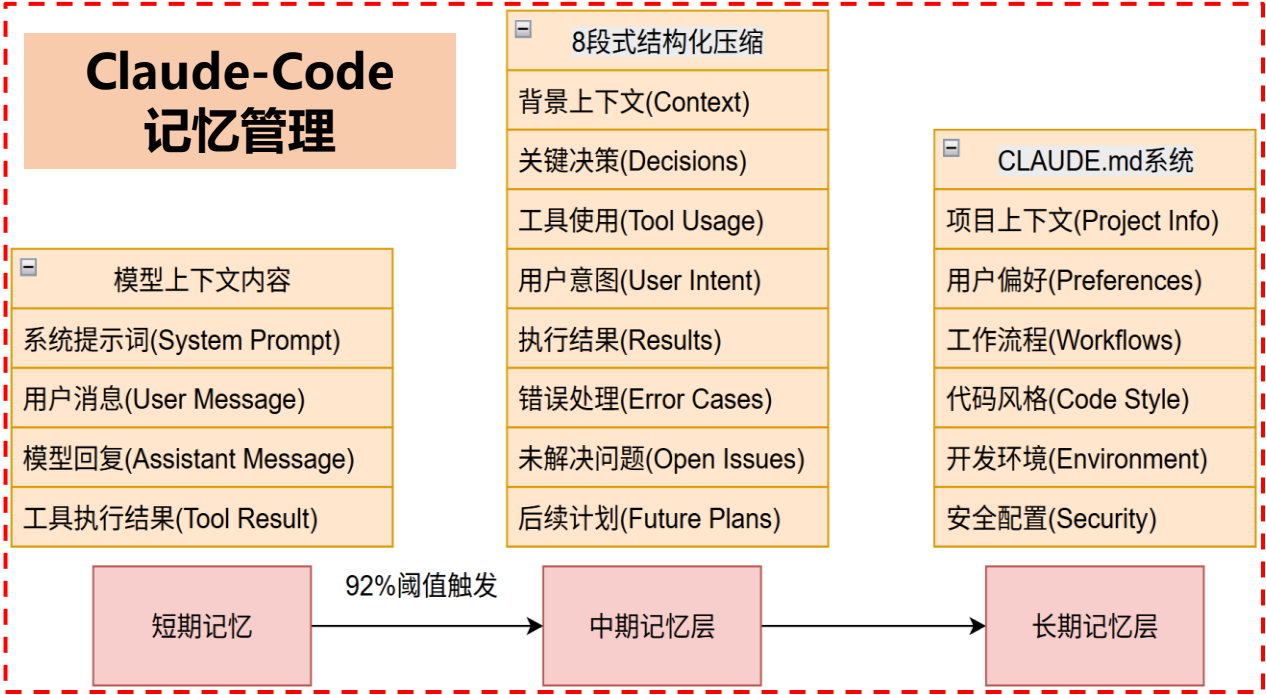
- 用户交互层
- Agent核心调度层
- Sub Agent /工具执行与管理
- 长短记忆和储存管理



Claude Code-核心调度层

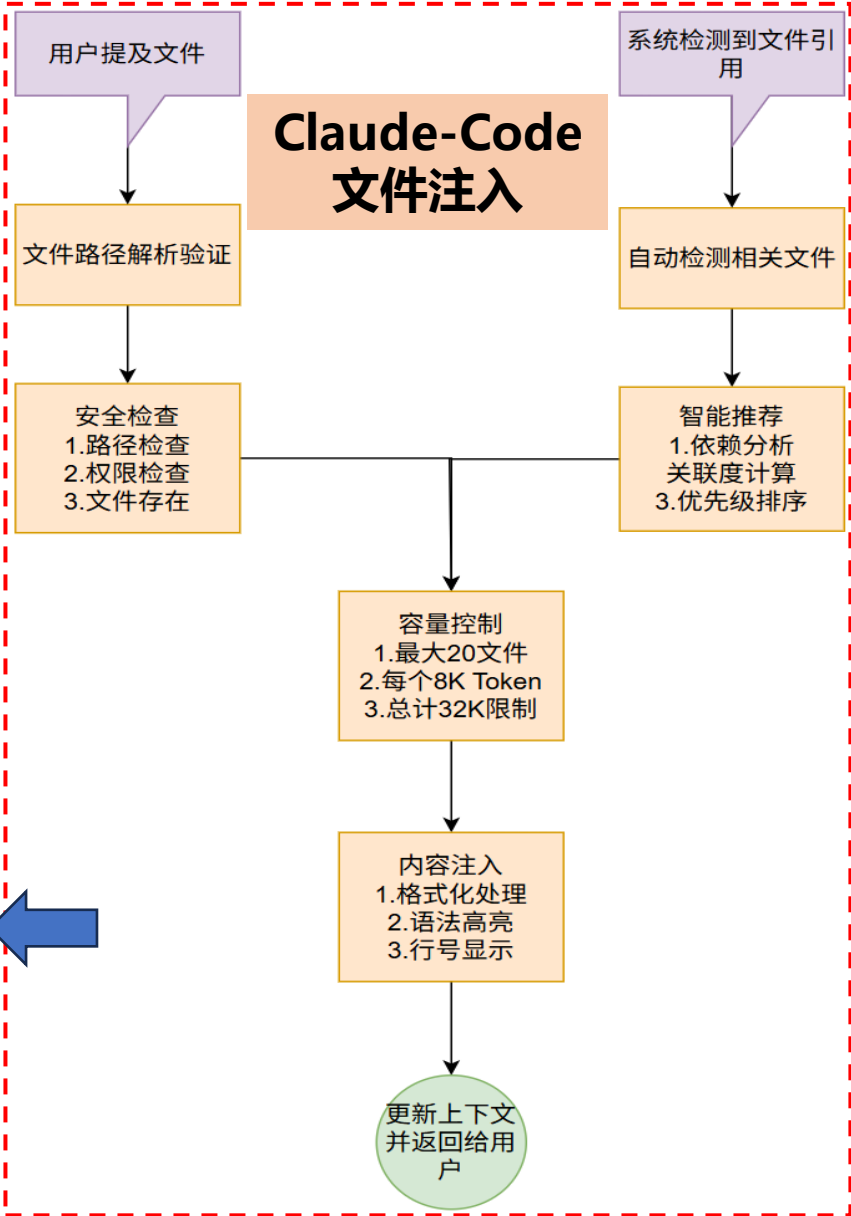


Claude Code-记忆与上下文管理机制



- 要点:
- 1.根据项目和任务压缩、保留、精炼上下文
 - 2.跨会话恢复，用户定制，项目持续记忆

- 要点:
- 1.文件的系统检测与管理
 - 2.文件的上下文注入



Claude Code-SubAgent架构与Task工具

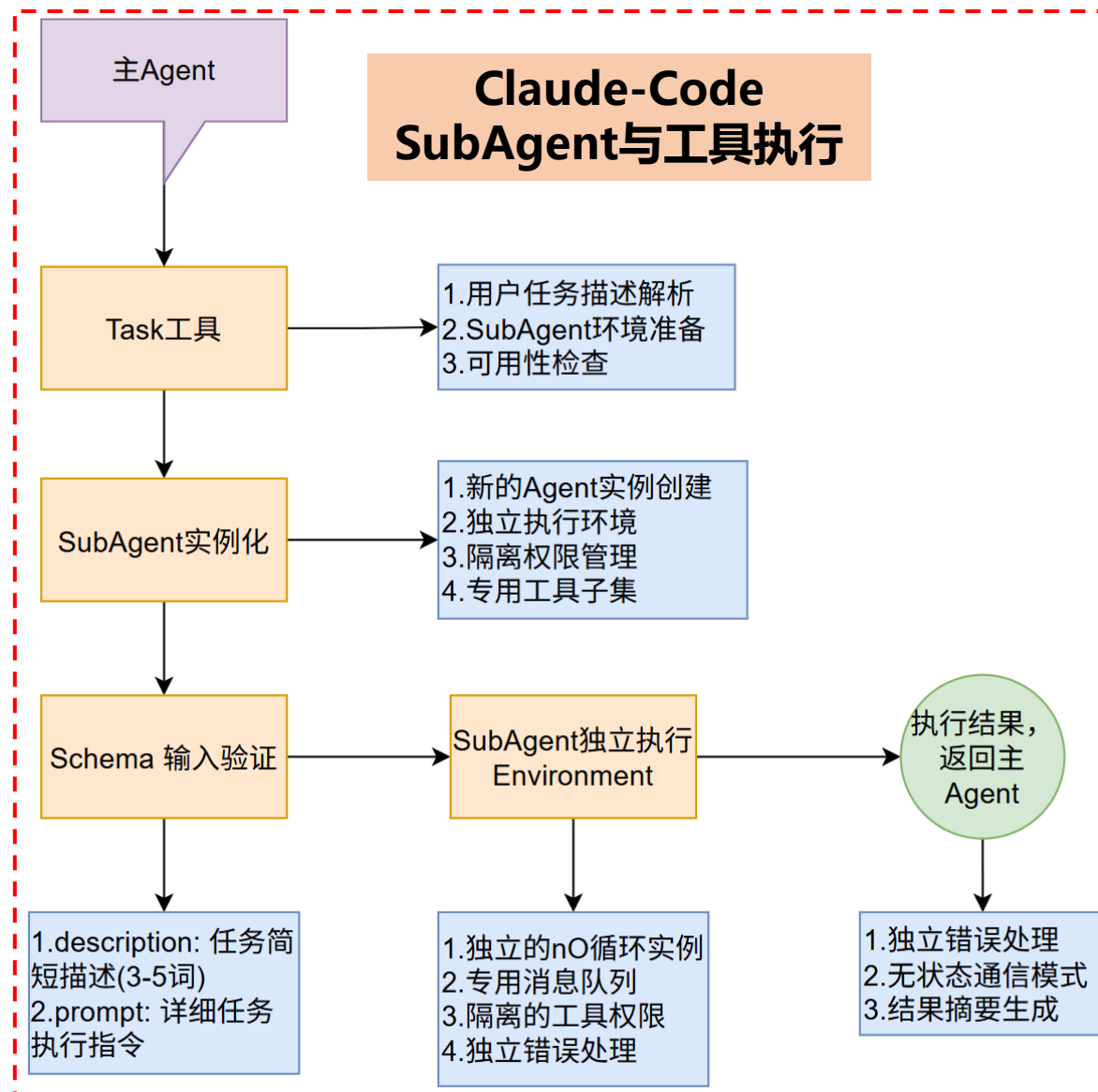


Claude Code-SubAgent

- 运行时实例化与可用性检查
- 具有专门的执行环境隔离和权限管理
- 内容多步骤执行，最终结果进行摘要精炼



多智能体系统中的专业（子）智能体的执行与管理



Claude Code-工具系统实现与协同机制

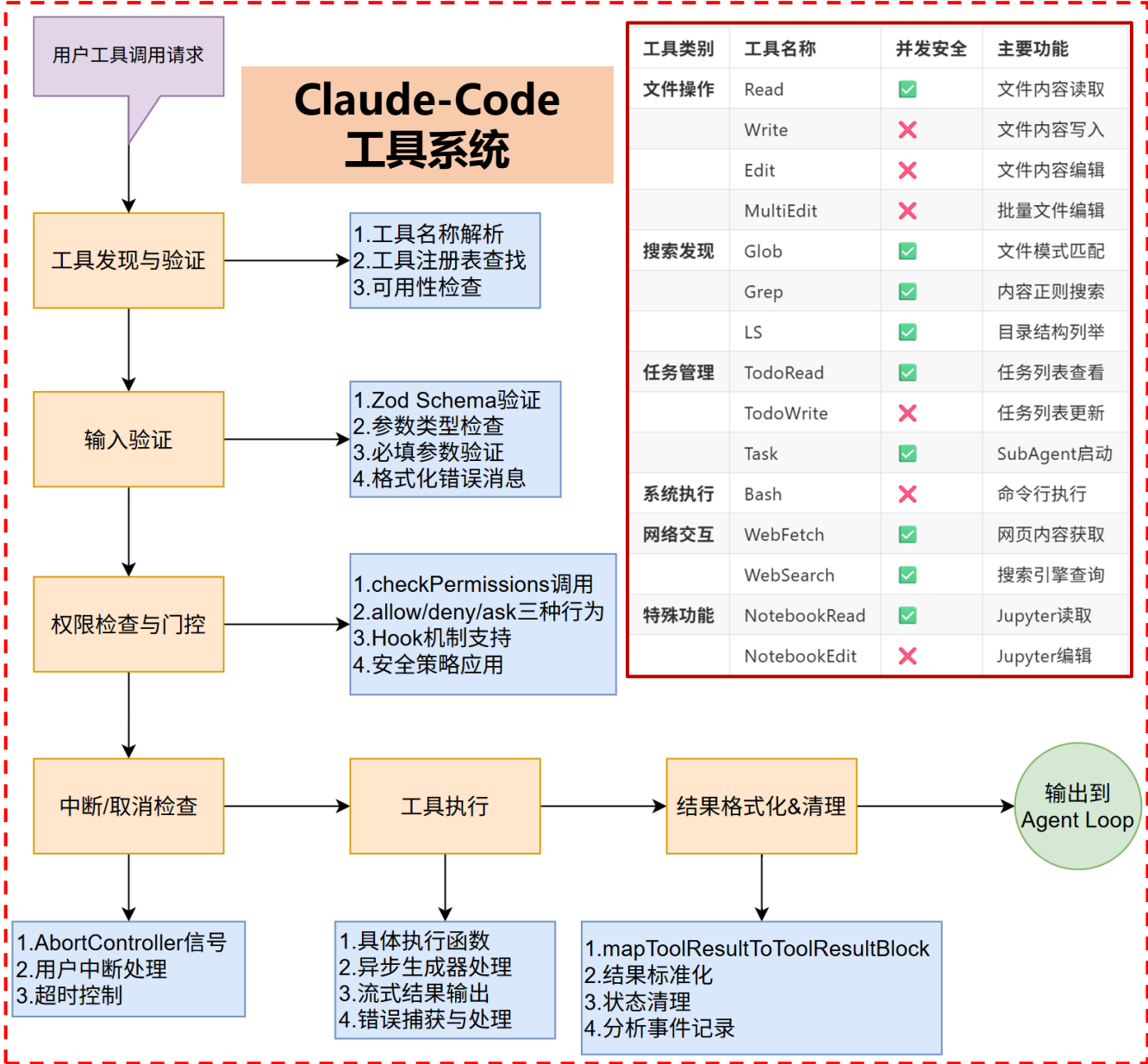


Claude Code-工具系统

- 工具的有效性验证与权限管理
- 工具/SubAgent的执行状态管理和异步监控
- 工具串行/并行的管理
- 工具运行状态管理
- 工具运行结果处理



工具系统的全流程管理



Claude Code-任务管理

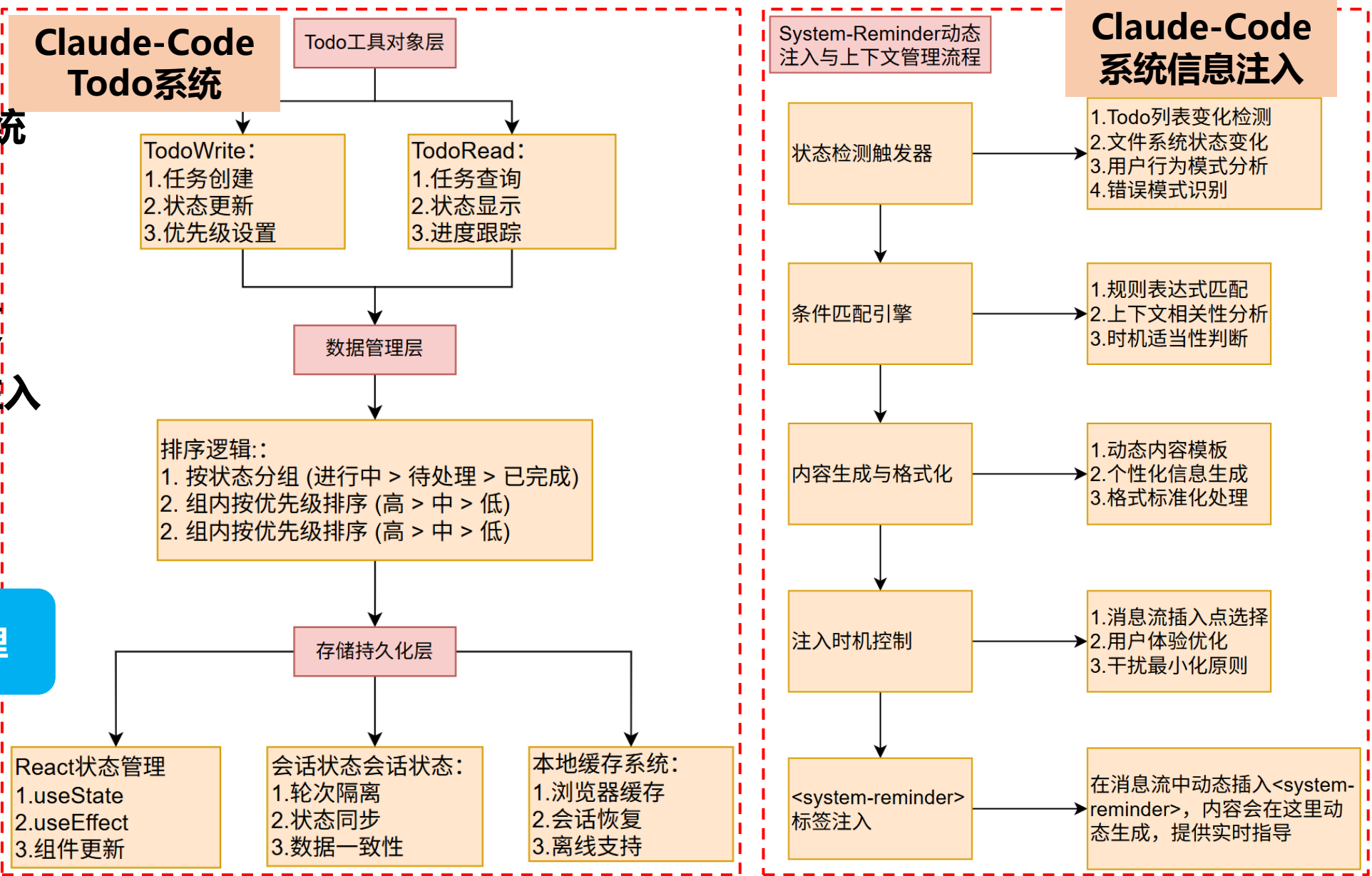


Claude Code-任务系统

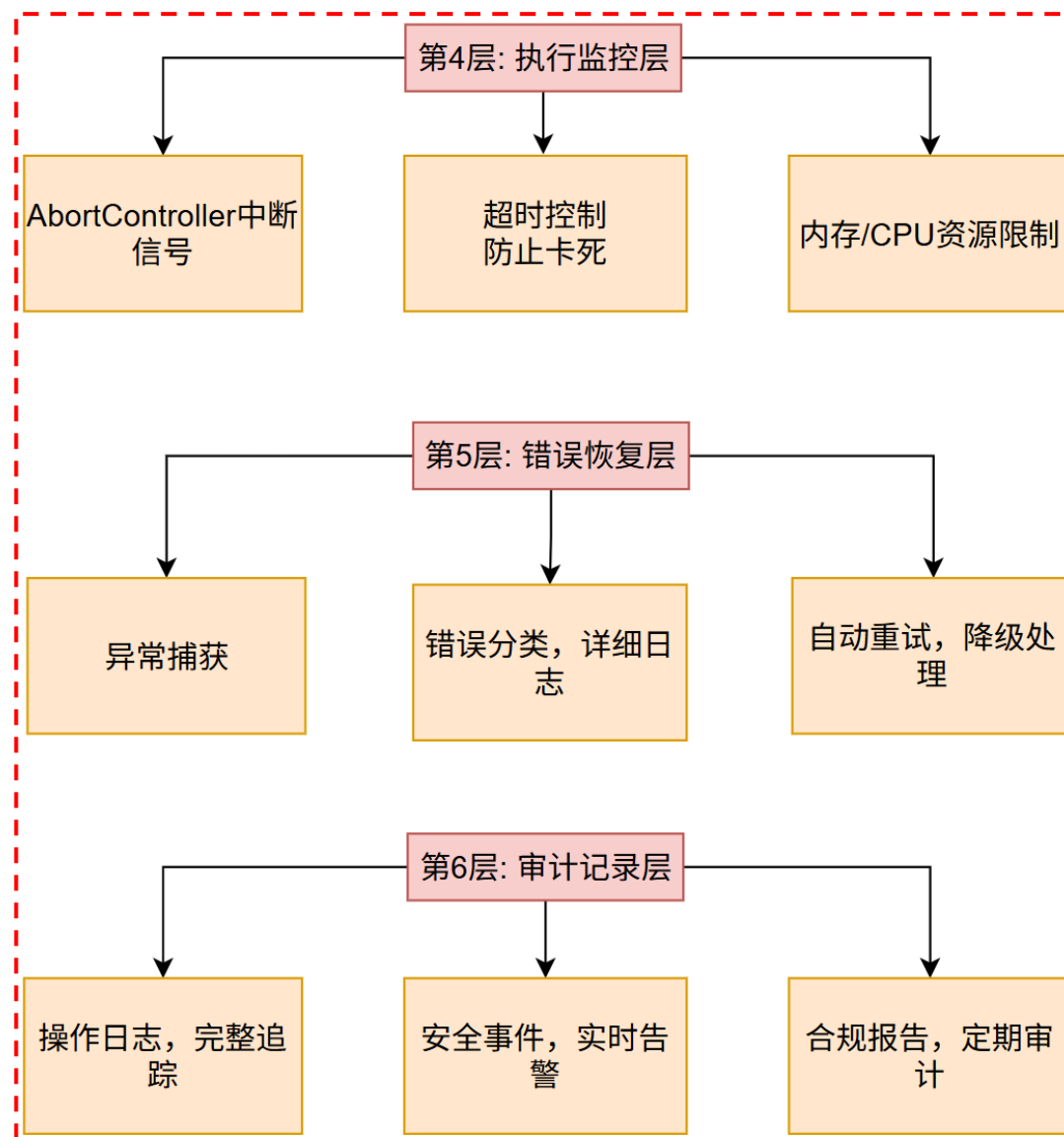
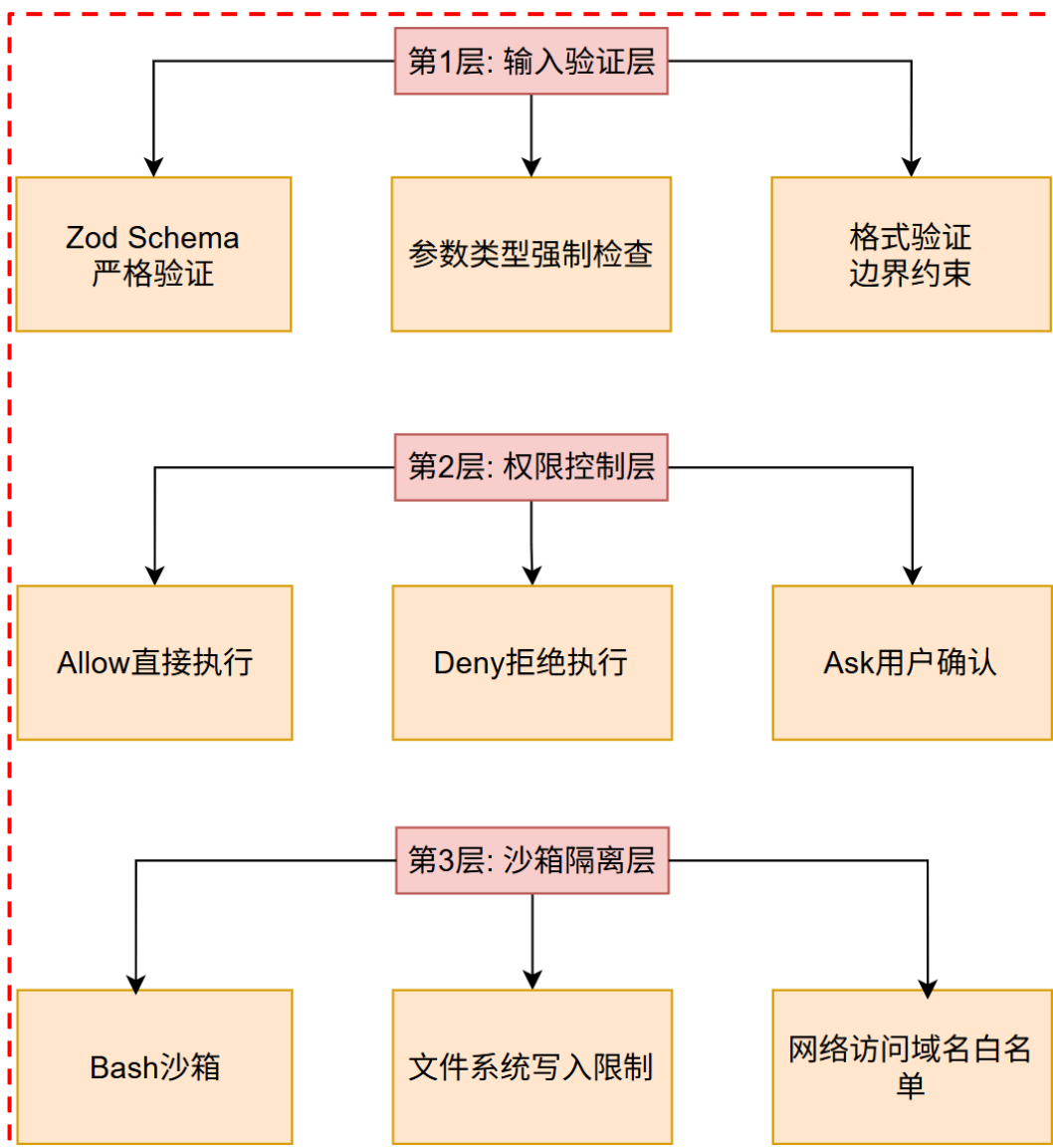
- 任务分类管理
- 任务轮次隔离
- 数据缓存和恢复
- 系统提示信息注入



任务系统的全流程管理

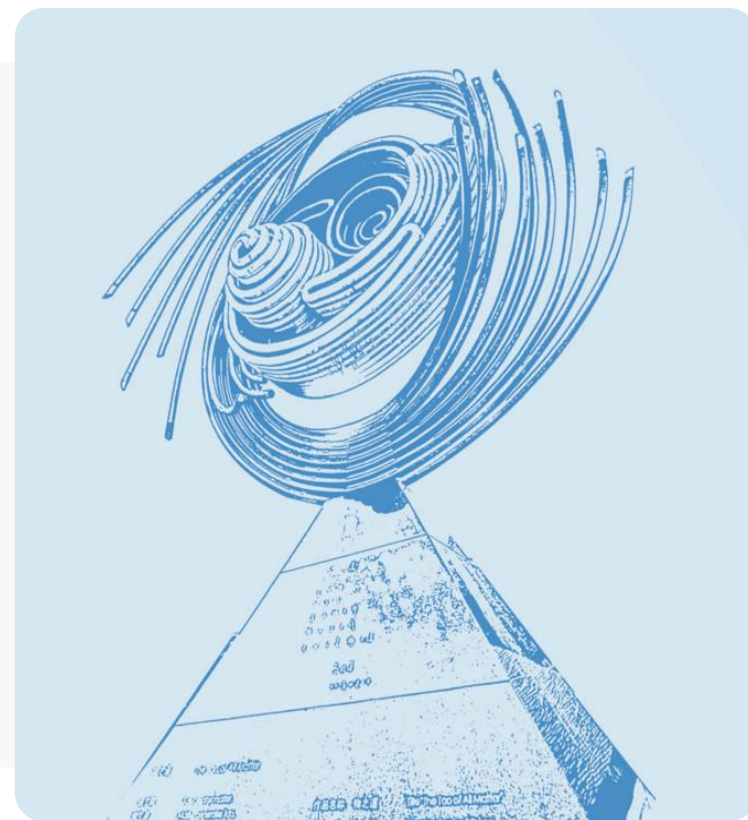


Claude Code-Security



Part 4

Claude Code 的使用技巧



使用技巧 - 模型切换



Claude Code 目前支持 Claude Opus 与 Claude Sonnet 4 两个模型的灵活切换，使用 /model 命令进行切换：

```
Select Model
Switch between Claude models. Applies to this session and future Claude
Code sessions. For custom model names, specify with --model.

>1. Default (recommended) Use the default model (currently Sonnet 4) · ✓
   $3/$15 per Mtok
2. Opus Opus 4.1 for complex tasks · $15/$75 per Mtok
3. Opus Plan Mode Use Opus 4.1 in plan mode, Sonnet 4 otherwise

Enter to confirm · Esc to exit
```

/model <model_name>自定义模型

HepAI API支持模型：

openai/gpt-5, \$1.25/\$10

openai/gpt-5-mini, \$0.25/\$2

aliyun/qwen3-coder, ¥ 4/ ¥ 16

deekseek-ai/deepseek-v3, ¥ 2/ ¥ 8

具体更多模型：

<https://aiapi.ihep.ac.cn>

红框圈出的文字提示可以看出模型的计费规则（MTok = 百万令牌tokens.）

以currently Sonnet 4为例，输入百万tokens 花费3美元，模型输出百万tokens花费15美元。

对于Claude来说，一个token大约代表3.5个英文字符



```
===== Previous Conversation Compacted =====
```

```
• Compact summary (ctrl+r to expand)
```

```
> /compact
```

```
  | Compacted. ctrl+r to see full summary
```

```
> |
```

```
? for shortcuts
```

```
Auto-updating...
```

Claude Code 提供了一个 `/compact` 压缩命令:

它会清除对话历史记录,但保留上下文中的摘要。你在让Claude code 进行一个项目任务时,当对话历史变得很长时,使用 `/compact` 可以压缩,确保 Claude code没有忘记整个项目的相关内容的同时,帮助用户减少tokens的花费,虽然 Claude Code 默认在上下文超过 95% 容量时自动压缩(可通过 `/config` 开启/关闭自动压缩)

使用技巧 - 记忆文件介绍



记忆类型	文件位置	用途说明	使用示例
项目记忆（共享）	./CLAUDE.md	项目团队共享的指令	项目架构、编码规范、常用工作流程
用户记忆（全局）	~/.claude/CLAUDE.md	用于所有项目的个人偏好设置	代码风格偏好、个人工具快捷方式

1

Claude Code 提供两种记忆位置，都有不同用途：

2

可以在多个位置放置 CLAUDE.md 文件，Claude Code 会递归读取这些文件，从当前工作目录开始，向上递归到根目录，读取找到的任何 CLAUDE.md 文件。

使用技巧 - 记忆文件介绍



在会话期间使用
/memory 斜杠命令，
可以在系统编辑器中
打开记忆文件：

Memory Files

Learn more: <https://docs.anthropic.com/en/docs/claude-code/memory>

Select memory to edit:

- > 1. Project memory Saved in ./CLAUDE.md
- 2. User memory Saved in ~/.claude/CLAUDE.md

Memory file does not exist yet. [Enter] to create ./CLAUDE.md.

最基础的就是可以通过在第二个用户级记忆
文件中写入：**每次请用中文回答我：**

这样设置记忆后,后续所有
项目的交互就都是中文回
答的了

使用技巧 – 注入本地文件



在使用Claude修改代码之前，先让 Claude 理解你的代码，可以在输入框使用@键触发本地文件选择，TAB键快速补全，选择本地文件

```
> @backend内是我的后端代码，  
@frontend/是我前端代码请阅读并理解，便于你后续进行修改
```

```
frontend/  
frontend/node_modules/content-type/  
frontend/node_modules/es5-ext/math/fround/  
frontend/node_modules/content-disposition/  
frontend/node_modules/undici-types/content-type.d.ts  
frontend/node_modules/clipboardy/node_modules/get-stream/source/contents.js  
frontend/node_modules/core-js/stable/object/from-entries.js  
frontend/node_modules/table/node_modules/ajv/lib/refs/json-schema-2020-12/meta/  
a/content.json  
frontend/node_modules/table/node_modules/ajv/lib/refs/json-schema-2019-09/meta/  
a/content.json  
frontend/node_modules/core-js/stable/math/fround.js
```

使用技巧 – 根据图片实现网页



Claude Code 可以发送图片并进行处理，在命令行中，把图片和提示词发过去，让它更好的理解你的意图。

```
> @屏幕截图 2025-08-18 171917.png 请根据这个图片的设计模型设计网页
```

使用技巧 – 恢复历史会话



```
(dioplas) muhongyuan@MuJiu:~/projects/dioplas$ claude --resume
```

	Modified	Created	# Messages	Git Branch	Summary
1.	1h ago	3h ago	165	-	you as, PAYG), ...
> 2.	3h ago	3h ago	2	-	
3.	3h ago	3h ago	6	-	

如果你已经进入了 Claude Code 会话，想恢复到之前的哪个历史会话，可以在对话框使用 `/resume` 命令恢复历史会话：

Claude Code 提供两个选项来恢复之前的对话：

`claude --resume` 或者 `claude -r`: 显示历史对话选择器；

`claude --continue` 或者 `claude -c`: 自动继续最近的对话，无需任何提示。

这两个带参数的命令需要在「非交互模式」下进行，也就是在终端输入

使用技巧 – 编辑模式



在 Claude Code 中，可以通过按下 shift + tab 键，切换编辑模：**一般模式**，**自动接受编辑模式**，**计划模式**

「自动接受编辑」功能 (auto-accept edits on)：

开启此功能后，Claude Code 会自动创建、编辑文件，而不是每次都需要我们手动确认。

```
>   
  
▶▶ accept edits on (shift+tab to cycle) 
```

「计划模式」功能 (plan mode on)：

在项目前期需要规划功能的时候可以用到这个模式，它会自动给到计划方案，并进行可行方案的优劣，然后底部你是否执行：

```
>   
  
|| plan mode on (shift+tab to cycle) 
```



添加 MCP 服务器

比如我来添加一个浏览器自动化操作的 MCP 服务器:
<https://github.com/modelcontextprotocol/servers-archived/tree/main/src/puppeteer>

基本语法

claude mcp add <name> <command> [args...]

例如

```
claude mcp add puppeteer npx -- -y @modelcontextprotocol/server-puppeteer
```

```
(dioptas) muhongyuan@MuJiu:~/projects/dioptas$ claude mcp add puppeteer npx -- -y @modelcontextprotocol/server-puppeteer
Added stdio MCP server puppeteer with command: npx -y @modelcontextprotocol/server-puppeteer to local config
File modified: /home/muhongyuan/.claude.json [project: /home/muhongyuan/projects/dioptas]
```

JSON 方式

claude mcp add-json <name> '<json>'

例如

```
claude mcp add-json -s user puppeteer '{
  "command": "npx",
  "args": ["-y", "@modelcontextprotocol/server-puppeteer"]
}'
```

使用 -s user 标志, 可以将 MCP 服务器添加到全局配置 (可以在 ~/.claude.json 文件中查看), 而不是只针对某个项目, 默认不填为 local 即当前项目。

```
(dioptas) muhongyuan@MuJiu:~/projects/dioptas$ claude mcp add-json -s user puppeteer '{
  "command": "npx",
  "args": ["-y", "@modelcontextprotocol/server-puppeteer"]
}'
Added stdio MCP server puppeteer to user config
```



管理 MCP 服务器

使用以下命令管理 MCP 服务器：

列出所有已配置的 MCP 服务器 `claude mcp list`

```
• (dioptas) muhongyuan@MuJiu:~/projects/dioptas$ claude mcp list
Checking MCP server health...

puppeteer: npx -y @modelcontextprotocol/server-puppeteer - ✓ Connected
```

获取指定的 MCP 服务器信息 `claude mcp get`

XX

```
• (dioptas) muhongyuan@MuJiu:~/projects/dioptas$ claude mcp get puppeteer
puppeteer:
  Scope: Local config (private to you in this project)
  Status: ✓ Connected
  Type: stdio
  Command: npx
  Args: -y @modelcontextprotocol/server-puppeteer
  Environment:
```

删除指定的 MCP 服务器 `claude mcp remove "xxx" -s local` 或 `claude mcp remove "xxx" -s user`

```
To remove from a specific scope, use:
claude mcp remove "puppeteer" -s local
claude mcp remove "puppeteer" -s user
```



Manage MCP servers

> 1. puppeteer ✓ connected · Enter to view details

MCP Config locations (by scope):

- User config (available in all your projects):
 - /home/muhongyuan/.claude.json
- Project config (shared via .mcp.json):
 - /home/muhongyuan/projects/dioptas/.mcp.json (file does not exist)
- Local config (private to you in this project):
 - /home/muhongyuan/.claude.json [project: /home/muhongyuan/projects/dioptas]

For help configuring MCP servers, see:

<https://docs.anthropic.com/en/docs/claude-code/mcp>

Esc to exit

01

进入 claude 对话后使用 /mcp 命令来查看 MCP 服务器相关信息:

02

代表刚才的浏览器自动化操作的 MCP 服务器:可以被Claude调用了



刚才的浏览器自动化操作的 MCP 服务器能够在真实的浏览器环境中与网页交互、截取屏幕截图和执行 JavaScript。调用尝试一下

- ✓ 可以看到，不用我指定使用，Claude会自动判断要完成我当前的任务可以使用哪些已经装载的MCP工具去实现

用户输入：

打开网址<https://www.bilibili.com/>，帮我分析下该网址首页的布局

```
> 打开网址https://www.bilibili.com/，帮我分析下该网址首页的布局

• I'll help you analyze the layout of Bilibili's homepage. Let me navigate to the website and take a screenshot to examine its structure.

• Update Todos
  [ ] Navigate to Bilibili homepage
  [ ] Take screenshot of the homepage
  [ ] Analyze and describe the layout structure

• Update Todos
  [ ] Navigate to Bilibili homepage
  [ ] Take screenshot of the homepage
  [ ] Analyze and describe the layout structure

• puppeteer - puppeteer_navigate (MCP)(url: "https://www.bilibili.com/")

Tool use

puppeteer - puppeteer_navigate(url: "https://www.bilibili.com/") (MCP)
Navigate to a URL

Do you want to proceed?
> 1. Yes
2. Yes, and don't ask again for puppeteer - puppeteer_navigate commands in /home/muhongyuan/projects/dioptas
3. No, and tell Claude what to do differently (esc)
```

运行结果实例：

```
• puppeteer - puppeteer_screenshot (MCP)(name: "bilibili_homepage", width: 1400, height: 900)
  | Screenshot 'bilibili_homepage' taken at 1400x900
  | [Image]
```

- Update Todos
 - ▣ ~~Navigate to Bilibili homepage~~
 - ▣ ~~Take screenshot of the homepage~~
 - ▣ Analyze and describe the layout structure
- 根据截图，Bilibili首页的布局结构如下：
 - 顶部区域
 - 头部导航栏：包含Bilibili Logo、搜索框、用户头像、消息、收藏、历史、创作中心等功能入口
 - 主导航菜单：首页、番剧、直播、游戏、知识、数码、生活、美食等分类标签
 - 主要内容区域
 - 左侧主推荐区：大尺寸的"AI开学季"推广banner，采用轮播图形式，底部有指示器
 - 右侧视频网格：推荐视频缩略图网格布局，每个视频卡片包含：
 - 视频封面图
 - 播放量、弹幕数、时长信息
 - 视频标题
 - UP主信息和发布时间
 - 下方内容区域
 - 更多推荐视频：继续展示推荐视频的网格布局
 - 每行大约4-5个视频卡片
 - 右上角功能区
 - 用户登录入口和相关功能按钮
 - 下载客户端提示

整体采用蓝白色调，布局清晰，以视频内容为核心，突出推荐算法驱动的个性化内容展示。

可以在红框圈出位置看到Claude自动调用了MCP访问浏览器和对浏览器页面进行截图的功能，读取截图后完成了我给的任务。



- 选择合适的模型
- 上下文压缩
- 设置记忆文件
- 注入本地文件
- 恢复历史会话
- 编辑模式
- 使用MCP

Part 5

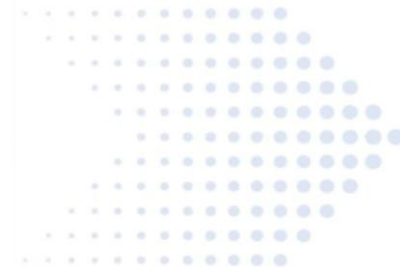
总结与展望





- **AI编程助手：Claude Code的基本原理和灵活应用**
- 自动编程辅助的综述
 - 发展历程
 - 一页ppt列出来
 - 多个工具（每个1-2页）
 - （早期）Eclipse
 - （经典）IntelliSense，代码补全、语法提示、语法高亮、代码折叠、错误检查etc
 - （现代）Github copilot, Fitten, Cursor, Claude Code
 - 一页横向对比图
- Claude Code的示例和使用
 - 一页简介
 - 多个ClaudeCode应用案例展现其能力
 - 使用方式（通过aiapi）
- Claude Code的基本原理
 - 参考Claude Code逆向工程分析：https://github.com/shareAI-lab/analysis_claude_code
- Claude Code的使用技巧
 - Claude设置文件/Agent/MCP
 - 上下文、费用、/clear、/compact
 - 遵循代码开发模式
 - 不同模型切换
 - ...

总结与展望



计算机终将取代算盘，汽车终将取代马车：

学会计算机、学会开车是重要的个人技能，学会正确使用AI亦如此。
未来十年，AI 不会取代人，能正确使用AI的人将取代不会AI的人

机 遇：

编码不再困难，未来更需要的是人是心中有产品蓝图、会用开发工具的人。

“双刃剑”：

最惨就业季! AI吞噬入门级岗位

.....
**Thanks for
Listening**





Claude Code记忆体

- 使用短中长进行记忆分区
- 通过大模型和提示词进行长记忆压缩
- 长文件、多文件通过关键信息压缩和多次读取进行信息压缩和上下文注入



智能体/多智能体系统的记忆层设计要点

学习的点:

- 智能体对话设计中，进行记忆结构化分区和信息压缩，以节省token和精炼信息
- 对多文件，如项目文件，可进行文件数量、内容的精炼压缩，以让智能体能快速理解项目内容
- 智能体可以使用RAG+长期记忆进行记忆持久化保存
- 可以在对话过程中实时学习、反思、保存系统的处理过程，进行实时学习和反馈



执行并发调度算法核心逻辑:

- 1. 并发安全工具可以同时执行 (最大10个)
- 2. 非并发安全工具必须串行执行
- 3. 混合场景下优先执行并发安全工具
- 4. 使用Promise.race实现抢占式调度
- 5. 动态调整并发度基于系统负载

并发冲突检测算法:

- └ 文件写入冲突检测 (Write/Edit工具)
- └ 系统资源竞争检测 (Bash工具)
- └ 状态修改冲突检测 (TodoWrite工具)
- └ 网络请求频率限制 (WebFetch/WebSearch工具)

