Claude Skills
整体总结



Skill = 说明书 + 代码 + 知识 + 素材

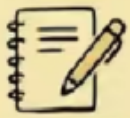
什么是Skills? 🤔

Skill = 标准化的文件包

⇩

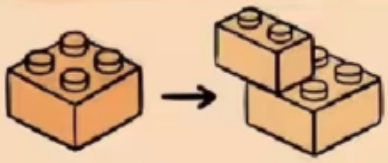
打包Agent完成任务所需的一切

Box 1: 员工手册



告诉新人该怎么做

Box 2: 乐高说明书



配件→组装→完成图

Box 3: 技能包



即插即用，开箱就会

```
my-skill/
├── SKILL.md
├── {/} scripts/
├── references/
└── assets/
```

核心特点：模块化 | 可复用 | 按需加载





Skills解决了什么问题？

❌没有Skills的痛点

Problems!

- 上下文爆炸
- Token成本高
- 信息稀释
- 提示词难维护
- 经验无法复用

引入Skills

✅有了Skills之后

Solutions!

- 分层按需加载
- 节省Token
- 模块化管理
- 沉淀可复用
- 执行更可控

核心价值：

从"一次性对话"→"可复用能力"

从"巨型Prompt"→"技能包解耦"

如何写好Skill？

开头三问

- 它帮我干什么？
- 什么时候出场？
- 与当前场景有何关系？

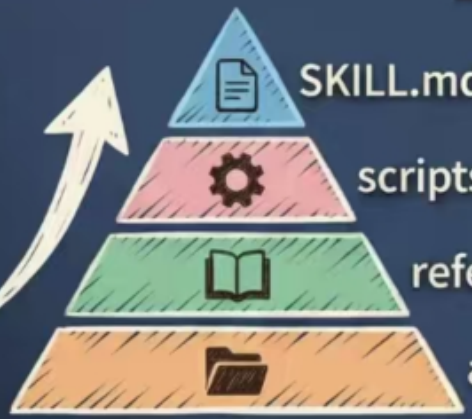
结构选择

- 流程型
- 任务菜单型
- 规范型
- 能力清单型

写作要点

- ☑ 名称+描述要清晰
- ☑ 判断条件要明确
- ☐ 控制在~500行
- ☐ 失败后怎么办？
- ☐ 写清分支与异常
- ☐ 确定性部分→脚本

金字塔架构



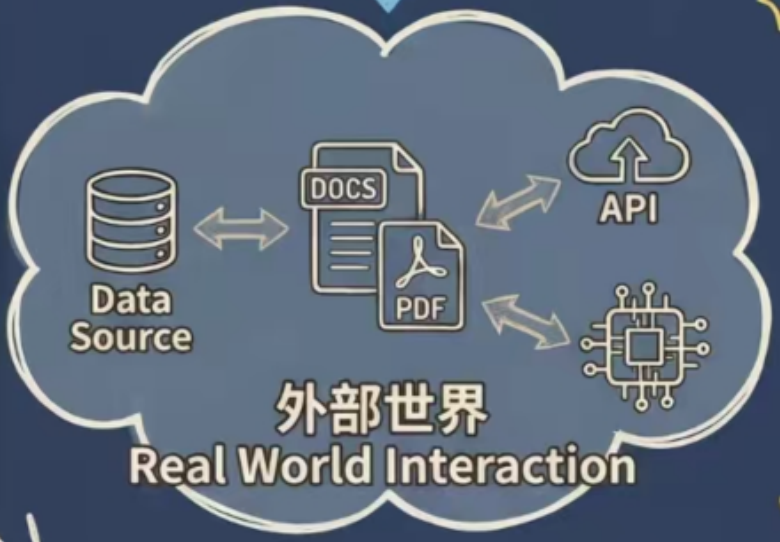
渐进式披露→按需加载

☆ 能脚本就脚本，别让AI即兴发挥！

Skills vs 其他工具







工具	作用	比喻
MCP	手脚问题	USB-C
Skills	流程方法	SOP手册
Rules	行为边界	纪律条例
RAG	记忆扩展	图书馆
Commands	快捷操作	按钮



一句话:MCP负责连接，Skills负责驱动

正交、可组合