Agent Skills 极简入门宝典

陈　放　张雁飞　著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

引言：为什么你需要 Skills

用过 ChatGPT、Claude 等聊天机器人的人均有过这种经历。

每次打开对话，你均要从头讲一遍需求。

“帮我写一份周报，要包含本周完成的工作、遇到的问题、下周计划，用简洁的列表格式。”

等工具执行完成，你很满意。

下周，你又要写周报了。

打开新对话框，需要再讲一遍需求。

“帮我写一份周报，要包含本周完成的工作、遇到的问题、下周计划，用简洁的列表格式。”

这就是传统聊天机器人存在的三大局限：

- 对话无状态：新对话无法继承历史上下文。
- 知识难沉淀：经验散落在不同对话中，难以复用。
- 提示词重复输入：相同指令需反复编写。

此前虽涌现出诸多新概念，如 MCP、Subagent 等面向不同 Agent 架构的设计模式，但均没有彻底解决这个问题，而 Skills 就是为此而生的。

本文要讲的就是：**让 AI 工具掌握你沉淀的专属工作方法，以后重复的工作，一句话就能搞定。**

先说清楚：这并不是高深莫测的技术。

它本质上是一种实用的规范化模式：将高频重复操作封装为结构化文件，AI 依据该文件自动执行。不仅省时省力，还不容易出错。

下面让我们一起来探索。

第 1 章

Agent Skill 到底是什么

先给出概要总结：**Skills 就像是给工具写的“使用说明书”**。

正如前文所说，用户将经验写成文件，下次直接使用，无须从头提要求。

1.1 传统方式面临的三个问题

问题一：每次均要从头说起。

每周五下午，你要写周报。每次均要告诉工具：

- 从聊天记录里找出这周做了什么事情。
- 从邮件里找出参加了哪些会议。
- 从项目管理软件中找出任务的完成情况。
- 写成需要的格式，简洁版或详细版。

每周均提一遍需求，每次均要写几分钟。这样操作十几遍之后，体验很差。

问题二：知识是分散的。

你的经验在头脑里，在聊天记录里，在各种文档里。下次想用时，

却找不到。

问题三：重复工作太多了。

同样的事情，每周都做，每月都做。每次均是复制内容，再稍微修改。半小时过去后，还在做这些琐碎之事。

1.2　Skill 带来的三个变化

如图 1-1 所示，与提示词相比，Skills 有三个变化。

图 1-1

1. 变化一：知识被封装起来了

将经验写成文件，存在固定的地方。下次需要使用时，用一句话指令调出来。Skills 的优点是不用从头写要求，无须翻找聊天记录，更不用查找文档。

2. 变化二：可以重复使用

写一次，用无数次。无须每次均复制文件，不用每次修改内容。

3. 变化三：可带工具

不仅包含经验，还能附带脚本、配置和 API 密钥等。工具需要什么，就安装什么。用的时候，便拿出来使用。

1.3 打个比方

好比一位新入职的员工。

当没有 Skills 时：

用户每次均问："这个怎么做？那个怎么做？"用户每次均要从头教智能体，它学得快，但忘得也快。

当有了 Skills 之后：

你给他一本标准化操作手册，智能体自己翻看手册，并执行任务。而且手册还能不断改进，当它遇到新问题，你将手册更新，Skills 就知道如何做。

总之：

Skills =结构化经验（文本定义）+ 工具（脚本、配置/API 集成）= 可复用、可维护的自动化方案

1.4 还有哪些类似的工具

很多人混淆了 Skills、Command、Subagent 等工具，这三类能力均受 Claude Code 原生支持。表 1-1 为不同工具的差别。

表 1-1

工具名称	特　点	通俗解释	典型场景
Skills	自动识别的技能包	特定任务的“岗位手册”	审查代码、处理 PDF
Command	斜杠命令	高频操作的“快捷键”	审查代码
Subagent	独立工作的分身	派出去干活的“外包人员”	调研竞品、搜索资料
MCP	外部服务连接器	连接外部工具的“插头”	操作 Notion、查找数据库

在什么场景下使用：

- **面对高频任务，使用 Command：**每天用十几次，输入 `/review` 比提需求还快。

- **低频任务用 Skills**：偶尔使用，免去记命令，随便说句话即可执行。
- **复杂任务用 Subagent**：如调研、重构，会产生大量内容，但要注意会污染当前对话。
- **操作外部服务用 MCP**：如要操作 Notion、GitHub 等外部工具。

1.6　Skills 的结构

一个典型的 Skills 文件夹结构通常如下（具体字段可根据需求调整）。

```
  skill-name/
├── SKILL.md          # 核心：定义什么时候用、怎么做
├── scripts/          # 脚本（Python、Bash 等）
├── references/       # 详细文档（按需加载）
└── config.yaml       # 配置（可选）
```

下面重点介绍。

1. SKILL.md：核心配置文件

SKILL.md 是 Skills 的灵魂，它定义了两个关键问题。

（1）何时用。文件开头的这段：

```
---
name: weekly-report
description: 当用户说"写周报"、"周报"、"weekly report"或需要总结本周工作时使用
---
```

其中的 description 是关键。Claude 依靠它来判断何时应触发该 Skill。它的质量决定了 Skill 能否被触发。

- 优秀的 description：当用户说“写周报”、“周报”、“weekly report”或需要总结本周工作时使用。
- 不推荐的 description：周报。

（2）怎么做。具体的执行逻辑如下。

```
你是周报生成专家。

## 工作流程

1. 从聊天记录中提取本周完成的工作
2. 从邮件中提取参加的会议信息
```

```
3．从项目管理软件中提取任务完成情况
4．按以下格式输出：
   - 本周完成工作
   - 进行中的工作
   - 遇到的问题

## 输出格式

使用简洁的列表格式，每项不超过一行。
```

写清楚具体要求，Claude 就会按此执行。

2. scripts/：脚本文件夹

编写辅助脚本，比如：

- fetch_github_issues.py：从 GitHub 拉取 issue。
- format_markdown.py：格式化 Markdown。
- send_email.sh：发送邮件。

这些脚本可以简化复杂的操作，让 Skills 更强大。但是，多数 Skills 仅需 SKILL.md 即可实现功能，无须额外脚本。

3. references/：详细文档

该文件夹采用了一种渐进式披露的设计模式。

- 对于简单任务：只加载 SKILL.md，速度快。
- 对于复杂任务：自动加载 references/中的详细文档，以提高输出质量。

比如在 references/api_reference.md 里可放置：

- 完整的 API 文档。
- 处理复杂场景的方法。
- 遇到错误后的应对指南。

这样既保证了日常使用的速度，又保证了完成复杂任务的质量。

4. config.yaml：配置文件（可选）

编写配置信息，比如：

```
api_key: your-key-here
default_format: markdown
max_length: 1000
```

但大部分时候不需要此文件。

下面是一个完整的例子。

将上面配置整合起来，展示一个完整的 Skills。

```
weekly-report/
├── SKILL.md
├── scripts/
│   └── fetch_jira_tasks.py
├── references/
│   ├── jira_api.md
│   └── email_format_guide.md
└── config.yaml
```

SKILL.md 内容：

```
---
name: weekly-report
description: 当用户说"写周报"、"周报"、"weekly report"或需要总结本周工作时使用
---

你是周报生成专家。

## 工作流程

1. 使用 `scripts/fetch_jira_tasks.py` 获取本周完成的任务
2. 从用户的描述中补充其他工作内容
3. 按标准格式输出

## 输出格式

### 本周完成
- [任务 1] - 简短描述
- [任务 2] - 简短描述

### 进行中
- [任务 3] - 进度描述

### 遇到的问题
- [问题描述] - 解决方案或下一步计划
```

该 Skills 可完成以下任务：从 Jira 获取任务、汇总周报内容、按规范格式输出。

总结一下：

- SKILL.md 是必需文件，用于定义触发条件与执行逻辑。
- scripts/存放辅助脚本，大部分时候不需要。
- references/存放详细文档，遇到复杂任务时才加载。

- config.yaml 存放配置，大部分时候不需要。

绝大多数 Skills 仅需 SKILL.md 即可满足需求。

1.7 支持 Skills 功能的平台（表 1-2）

表 1-2

平　台	能不能用	备　注
Claude Code	支持	官方工具，功能最全
VS Code	支持	需要安装插件
Cursor	支持	原生支持
Coze	支持	字节跳动出品
Antigravity	支持	谷歌出品
Trae	支持	国产平台
JetBrains	部分支持	插件开发中

目前，主流 AI 编程工具大多支持 Skills 功能或其类似能力，如图 1-2 所示。除此之外，因为 Deepseek、智谱、Kimi 等大模型兼容 Anthropic 的 Claude Code，因此这些模型在接入层面也可间接支持 Skills 相关能力。

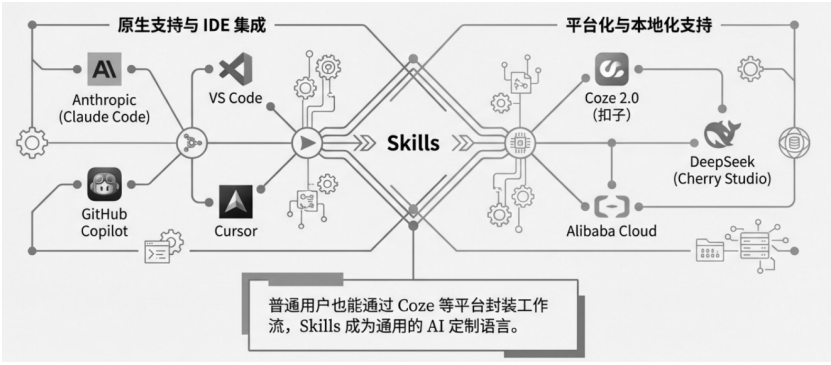

图 1-2

第 2 章

手把手教你创建第一个 Skills

本章会带你从零开始，创建一个完全可用的 Skills——**“微信文章格式化器”**。

2.1 实战场景

问题：经常要将普通文本转换成微信公众号格式（加粗、添加标题、设置引用等）。

传统做法：手动加星号、调整格式、反复预览——每次要 10 分钟。

有了 Skill 之后： 用一句话即可搞定——只需几秒钟。

2.2 方法一：用 Skill Creator

无须写任何代码，完全用自然语言即可实现。

Step 1：安装 Skill Creator

在 Claude Code 中直接说（通过 Anthropic 官方提供的 Skill Creator 工具实现）：

```
帮我安装 skill-creator，地址是：
https://git***hub.com/anthropics/skills/tree/main/skills/skill-creator。
```

Step 2：用自然语言描述需求

```
用户：创建一个 skill，能将普通文本转换成微信公众号格式。
```

Skill Creator 会自动执行下面的任务：

- 询问细节（需要哪些格式、触发词是什么）。
- 创建文件夹结构。
- 生成 SKILL.md 配置文件。
- 创建脚本模板。

完整对话示例：

```
用户：创建一个 skill，能将普通文本转换成微信公众号格式。

Claude：好的！请回答几个问题：

1. 这个 skills 的名称叫什么？（建议：wechat-formatter）
答：wechat-formatter。

2. 触发该 Skills 的关键词有哪些？
答：微信格式、公众号格式、转发微信文章。

3. 需要支持哪些格式？
答：标题（#）、加粗（**）、引用（>）、列表（-）。

Claude：Skills 创建完成！
      位置：~/.claude/skills/wechat-formatter
```

Step 3：测试 Skills

```
你：帮我将这段文字转成微信格式

我是标题
这是加粗的文字
这是普通文字

Claude：# 我是标题
       **这是加粗的文字**
       这是普通文字
```

2.3　方法二：手动创建（完全控制）

如果想完全掌控 Skills 的行为，手动创建是最好的方式。

Step 1：创建文件夹结构

```
# 进入技能目录
cd ~/.claude/skills

# 创建新 skills 文件夹
mkdir wechat-formatter
cd wechat-formatter

# 创建必要的文件夹
mkdir scripts referenCES
```

Step 2：编写 SKILL.md（核心配置）

```
cat > SKILL.md << 'EOF'
---
name: wechat-formatter
description: 当用户说"微信格式"、"公众号格式"、"转微信文章"或需要将普通文本转换为适合微信公众号发布的格式时使用
---

你是一位专注于微信公众号内容格式化的 AI 助手。

## 核心功能

将普通文本转换为微信公众号支持的 Markdown 格式：
- 标题：`# 标题`
- 加粗：`**文字**`
- 引用：`> 引用内容`
- 列表：`- 列表项`
- 分割线：`---`

## 处理规则

- 自动去除连续空行（保留一个空行）
- 保持原文段落结构
- 不改变链接和图片格式
```

```
## 输出格式

直接输出格式化后的 Markdown，不需要额外解释。
EOF
```

Step 3：测试 Skills

重启 Claude Code：

```
exit
claude
```

测试：

```
你：将这段文字转成微信格式：

AI 技术发展趋势
人工智能正在改变世界
机器学习是重要分支

Claude：# AI 技术发展趋势

        人工智能正在改变世界

        - 机器学习是重要分支
```

完美运行！

2.4 实用技巧：Skills 的调试方法

1. 检查 description 是否准确

若 Skills 未被触发，通常是因为 description 字段描述不够准确或覆盖不足。

- 好的 description：当用户说“审查代码”、“review”、“帮我看看代码”、“检查 bug”、“这段代码有问题吗”时使用。
- 差的 description：代码审查。

2. 测试触发条件

用不同的提示词测试：

- “帮我审查一下代码”。
- “这段代码有没有问题”。
- “review 一下”。

看哪种说法能触发 description，哪种不能。

3. 查看日志

Claude Code 会在控制台上显示加载了哪些 Skills。如果未加载，请检查路径和文件名。

2.5　如何安装 Skills

方法一：通过自然语言指令安装（推荐新手）

直接在 Claude Code 中说：

```
帮我安装 skill，地址是[GitHub URL]
```

方法二：手动安装（最可控）

```
# 进入技能目录
cd ~/.claude/skills

# 克隆 Skill 仓库
git clone [GitHub URL] [skill-name]

# 重启 Claude Code
```

方法三：通过插件市场安装（最便捷）

```
# 注册市场
/plugin marketplace add anthropics/skills

# 搜索并安装
/plugin marketplace search [关键词]
/plugin install [skill-name]
```

2.6　常见问题

问：Skills 已创建，但未生效？

答：重启 Claude Code。Skills 只在启动时加载。

问：怎么知道 Skills 被触发了？

答：控制台将输出“Using skill: xxx”。

问：一个任务能触发多个 Skills 吗？

答：不会。Claude 将选择相关性最高的 Skills。

问：Skills 能调用其他 Skills 吗？

答：不能。各 Skills 相互隔离，不支持跨 Skills 调用。

问：能把 Skills 分享给团队吗？

答：可以。将 Skills 对应的文件夹提交至 Git 仓库，团队成员克隆后将其置于~/.claude/skills/即可。

第 3 章

这些 Skills 值得收藏

社区已经涌现了大量优秀的 Skills。以下精选几款最实用的技能，并附上快速安装指南。

3.1 官方必备 Skills（Anthropic 出品）

3.1.1 Skill Creator：可自动生成技能的 Skills

Skill Creator 是一个能自动创建其他 Skills 的 Skills，堪称“元技能”。

它能做什么：

- 无需编程基础，仅需描述需求，即可自动生成完整的技能目录结构。
- 自动编写 SKILL.md 的触发词和描述。
- 自动生成脚本模板。

快速安装：

```
帮我安装 skills，地址是：
https://git***hub.com/anthropics/skills/tree/main/skills/skill-creator
```

3.1.2 PPTX Generator：一键生成 PowerPoint 演示文稿

从 HTML 直接生成专 PowerPoint 演示文稿。

能帮你做什么：

- Markdown/HTML 转 PPTX。
- 自动套用预设模板。
- 批量生成幻灯片。
- 支持中文字体和排版。

快速安装：

```
请安装 skills，地址是：
https://git***hub.com/anthropics/skills/tree/main/skills/pptx
```

3.2 社区爆款 Skills

3.2.1 Superpowers：全栈开发工作流

由开发者 obra 创建的完整软件开发流程 Skills。

能做什么：

- 自动生成需求文档（PRD）。
- 编写代码并自动测试。
- 代码审查和优化建议。
- 在 Git 提交信息生成。
- 生成完整的 CI/CD 流程。

适合人群：独立开发者、小型开发团队、创业者。

快速安装：

```
帮我安装 skills，地址是 https://git***hub.com/obra/superpowers.git
~/.claude/skills/superpowers
```

3.2.2 X Article Publisher——推特创作者神器

快速创作和发布 X（Twitter）文章的 Skills。

能做什么：

- 生成推文串（thread）。
- 自动控制字数并适配目标语气风格。
- 优化标题和钩子。
- 生成相关话题标签。

适合人群： 内容创作者、运营人员、KOL。

快速安装：

```
帮我安装 skills，地址是 https://git***hub.com/wshuyi/x-article-publisher-skill.git ~/.claude/skills/x-article-publisher
```

3.2.3　NotebookLM Bridge——连接 Google NotebookLM

在 Claude Code 中直接使用 NotebookLM 的知识库功能。

能做什么：

- 检索你的 NotebookLM 中创建的知识库笔记。
- 直接将 PDF 上传到 NotebookLM。
- 基于知识库回答问题。
- 多源知识整合。

适合人群：研究人员、学生、知识管理爱好者。

快速安装：

```
帮我安装 skills，地址是 https://git***hub.com/PleasePrompto/notebooklm-skill.git ~/.claude/skills/notebooklm
```

3.2.4　Obsidian Skills：笔记软件增强

由 Obsidian 创始人亲自编写的 Skills。

能做什么：

- 生成 Obsidian 兼容的 Markdown。
- 自动添加标签和元数据。
- 生成 ObsidianCanvas 白板。
- 完整保留原始 Markdown 格式。

适合人群：Obsidian 用户、知识管理爱好者。

快速安装：

```
帮我安装skills,地址是 https://git***hub.com/kepano/obsidian- skills
```

注意，该仓库包含多个子技能，但仍支持通过自然语言指令统一安装。

3.3　实用工具类 Skills

3.3.1　PDF Toolkit：PDF 全功能处理工具

由 Anthropic 官方出品的 PDF 处理工具。

能做什么：

- 合并多个 PDF 文件。
- 拆分 PDF 为多个文件。
- 将 PDF 转换为其他格式。
- 提取 PDF 中的文本和图片。
- 压缩 PDF 文件大小。

适合人群：行政与文秘人员、文档工程师、科研人员。

快速安装：

```
帮我安装 skills，地址是 https://git***hub.com/anthropics/skills/tree/main/skills/pdf
```

3.3.2 WebApp Testing：代码审查专家

由 Anthropic 官方出品的代码质量检查工具。

能做什么：

- 自动识别常见安全漏洞（XSS、SQL 注入等）。
- 性能问题分析。
- 代码规范合规检查。
- 最佳实践建议。
- 自动生成测试报告。

适合人群：开发者、QA 工程师、技术负责人。

快速安装：

```
帮我安装 skills，地址是 https://git***hub.com/anthropics/skills/tree/main/skills/webapp-testing
```

3.4 推荐安装清单

（1）如果你是新手入门

- skill-creator：学会创建自己的 Skills。
- pptx：快速生成演示文稿。
- webapp-test：提升代码质量。

（2）如果你是内容创作者

- x-article-publisher：社交媒体内容创作。
- obsidian-skills：笔记管理。

- pdf：文档处理。

（3）如果你是开发者

- superpowers：全栈工作流。
- skill-creator：快速开发自定义技能。
- notebooklm-skill：基于 NotebookLM 的智能技术文档分析与问答。

记住：不要一次安装太多的 SKills，建议控制在 3 ~ 5 个以内，多了反而会混乱。