

# Spec-Kit 核心 workflow 指南

## Spec-Kit 核心 workflow 指南

简介

5 分钟快速上手

核心命令参考手册

1. `/speckit.constitution`
2. `/speckit.specify`
3. `/speckit.clarify`
4. `/speckit.plan`
5. `/speckit.tasks`
6. `/speckit.implement`

最佳实践与避坑指南

环境设置

FAQ

## 简介

---

Spec-Kit 是一个基于 **规范驱动开发 (Spec-Driven Development)** 理念的项目治理工具集。它通过一系列结构化的斜杠命令，引导开发者从需求分析、技术设计到任务拆解，最后实现自动化执行，确保代码与文档的高度一致性。

## 5 分钟快速上手

---

假设你要添加一个“用户登录”功能：

1. **定义需求**: `/speckit.specify 用户登录` -> 生成 `spec.md` 。
  2. **消除模糊**: `/speckit.clarify` -> 交互式回答问题，完善 `spec.md` 。
  3. **技术规划**: `/speckit.plan` -> 生成 `plan.md` , `data-model.md` 等。
  4. **拆解任务**: `/speckit.tasks` -> 生成可执行的任务列表 `tasks.md` 。
  5. **开始实施**: `/speckit.implement` -> 自动化执行 `tasks.md` 中的任务。
- 

## 核心命令参考手册

---

### 1. `/speckit.constitution`

**用途**: 确立项目章程和核心原则。

**场景**: 项目初始化或原则变更（如强制双格式文档）。

### 2. `/speckit.specify`

**用途**: 从自然语言描述创建功能规范（`spec.md`）。

**建议**: 描述要包含“谁”、“做什么”以及“为什么”。

### 3. `/speckit.clarify`

**用途:** 扫描规范中的模糊点并交互式解决。

**原则:** 在开始规划前，确保核心需求无歧义。

### 4. `/speckit.plan`

**用途:** 进行技术设计和架构规划。

**产物:** `plan.md` , `data-model.md` , `research.md` 等。

### 5. `/speckit.tasks`

**用途:** 将设计转化为具体的、可执行的任务列表 ( `tasks.md` )。

**特点:** 支持并行任务标记 `[P]` 和用户故事关联 `[US1]` 。

### 6. `/speckit.implement`

**用途:** 按照 `tasks.md` 的计划自动执行代码编写和验证。

---

## 最佳实践与避坑指南

---

- **独立测试:** 每个用户故事应能独立运行和测试，避免庞大的“全能型”提交。
- **迭代澄清:** 不要急于写代码， `/speckit.clarify` 能帮你省下大量返工时间。
- **双格式同步:** 确保所有文档都包含 Markdown 和 PDF。

## 环境设置

---

如果 PDF 生成失败，请确保已安装以下依赖：

```
pip install playwright markdown
playwright install chromium
```

## FAQ

---

### Q: 为什么生成的 PDF 没更新?

A: 检查是否触发了自动同步逻辑，或者手动运行 `python generate_learning_pdf.py <file.md>`。

### Q: 如何自定义模板?

A: 修改 `.specify/templates/` 目录下的对应文件。