00 的 Skill Checklist

🗄️ author：00 + ChatGPT
website：www.tophci.com
公众号：零反思
update：2025-12-062025-12-06

🧱 A. 定位与元信息（Header 层）

- [ ] `name` 是具体的**动宾结构**（`processing-pdfs` / `mcp-builder` / `algorithmic-art`）。
- [ ] `description` 说清楚三件事：
  - [ ] 它**做什么**（帮模型干成哪一类真实工作）。
  - [ ] **何时使用 / 如何触发**（典型场景 / 文件类型 / 关键词）。
  - [ ] **不做什么**（边界、排除项，而不是泛泛地"什么都能做"）。

🧩 B. 结构与信息架构（Pattern 层）

- [ ] skill 明确采用了一个主结构模式：
  - 流程型：`Overview → Workflow / Phases → Step 1,2,3...`
  - 菜单型：`Overview → Quick Start / Common Tasks → Task groups`
  - 规范型：`Core Principles → Guidelines → Specific Rules → Examples`
  - 能力清单型：`Role / Goals → Capability Map → Capability Items → Combos`
- [ ] 标题层级清晰，主干 1–2 层就可以跑通任务。
- [ ] `SKILL.md` 只放**骨干信息**（流程/规范/决策规则/示范样例），大块细节放在 `references/`，用一句话 + 链接引用。
- [ ] 禁止套娃，避免 `SKILL.md` → advanced.md → details.md 的结构。

🧬 C. 思维模型与内容质量（How to think）

> skill 不是只告诉“要做什么”，而是教模型“先怎么想，再怎么做”。

- [ ] 在开头给出高级的 framing，把任务从“具体操作”提升到“方法论 / 流派”层，比如把设计变成“visual movements”，把 server 变成“让 LLM 真正完成任务的工程系统”。
- [ ] 在动手之前要求模型先**思考/规划**，比如先写哲学/设计论再写代码，先做需求分析 / 约束分析 / tradeoff 决策再实现。
- [ ] 对关键决策写出**可复用维度**：
  - 例如算法类：噪声函数、粒子行为、场、时间、参数、涌现；
  - 设计类：排版、色彩、动效、留白、信息密度；
  - 工程类：覆盖率、错误处理、分页、可复现、性能。
- [ ] skill 中同时包含：
  - [ ] 正向指导（应该怎么做）；
  - [ ] 负向禁忌（绝不要怎样，比如紫渐变 + Inter + 全居中式 AI slop）。

🗃️ D. 资源与模板组织（Scripts / References / Assets）

- [ ] 可重复使用的代码逻辑放在 `scripts/`。
- [ ] 大块的规范 / API / schema / 文档放在 `references/`。
- [ ] 模板 / wireframe / HTML / PPT / JSON 样例等放在 `assets/`。
- [ ] `SKILL.md` 里清楚说明哪些是固定模板 / UI 框架，哪些是可自由发挥的变量。
- [ ] 对于前端 / artifact 类 skill，清楚写出初始化脚本（如 `init-artifact.sh`）、打包脚本（如 `bundle-artifact.sh`）、输出文件名（如 `bundle.html`）。

🛠️ E. 工程性与工具设计

- [ ] 工具 / 接口都有**结构化输入/输出 schema**（Zod / Pydantic / JSON schema），字段有类型、约束 / 枚举、说明、示例等。
- [ ] 工具命名清晰（`domain_action_object`），方便模型通过名称猜用途。
- [ ] 有合理的注解：`readOnly` / `destructive` / `idempotent` / `openWorld` 等（按具体框架）。
- [ ] 错误信息写明发生了什么，给出下一步建议（重试条件、改参数、换工具或找人类）。
- [ ] 有针对上下文窗口的设计：
  - [ ] 大结果分页 / 限制条数；
  - [ ] 优先返回结构化数据 + 必要文本解释。
- [ ] 对 determinism / reproducibility 有要求，例如使用 seed / 固定参数（在需要可复现的任务中，尤其是生成类）。

🎨 F. 审美与创意质量

- [ ] skill 区分**哲学/设计理念**和**具体实现**两个层次：
  - [ ] 先写流派/哲学/设计哲学（4–6 段，覆盖关键维度）；
  - [ ] 再输出成图片/代码/文档/布局。
- [ ] 要求概念是“深埋在结构里的种子”，而不是直接写在画面上，例如把主题抽象成色彩比例、动线、参数范围。
- [ ] 有反 AI-slop 的强约束：
  - [ ] 禁止默认字体组合（Inter/Roboto/Arial 滥用）；
  - [ ] 禁止默认紫渐变、全居中卡片墙；
  - [ ] 禁止所有元素统一圆角、一刀切阴影。
- [ ] 明确要求**精修阶段不新增、只“打磨”**，例如对齐、排布、层次、色彩细节；
- [ ] 对视觉/交互有可检查项：
  - [ ] 不 overlap、不出界、有安全边距；
  - [ ] 动效少而精；

- [ ] 色板有限且有主副色之分。

📊 G. 示例 / 用例 / 反例（Few-shot & Anti-shot）

- [ ] 有 2–5 个高质量“示范条目”：
  - [ ] 创意类：每条是 “哲学描述 + 算法/视觉结构翻译”。
  - [ ] 工程类：每条是 “用户目标 → 调用序列 / 工具设计范例”。
- [ ] 有明确的**反例**（bad patterns），说明反例是什么样的，为什么不接受它。
- [ ] 示例中覆盖标准场景、边界场景、带冲突/权衡的场景（比如：优先安全 vs 优先效果）。

✅ H. 自检与评估（Evaluation & Second Pass）

- [ ] skill 明确写了“**成功的定义**”：
  - 对创意类：外观质感、概念是否嵌入、可复现性。
  - 对工程类：能否完成任务、错误处理是否合理、性能与可维护性。
- [ ] skill 设计了一些**评测任务 / 测试集**：
  - [ ] 复杂真实、可验证、有唯一答案或明确判定标准。
  - [ ] 记录方式结构化（比如 XML / JSON 测试描述）。
- [ ] 在输出前，有一小节“自检 checklist”：
  - [ ] 是否遵守核心原则 / 规范？
  - [ ] 是否踩到任何禁区（审美/安全/上下文浪费）？
  - [ ] 是否还有可以删减 / 合并 / 精简的部分？

🧭 I. 边界、版本与演化（Scope & Evolution）

- [ ] 明确 skill 的适用范围（domain / 项目 / 团队）。
- [ ] 清晰写出当前版本的假设与限制（data、工具、框架）。
- [ ] 没有让模型承担自己做不到的职责，而是对高风险场景要求人类介入。

更多 AI 研究和技巧，请访问 www.tophci.com 或公众号「零反思」