

the-craft-of-selfteaching

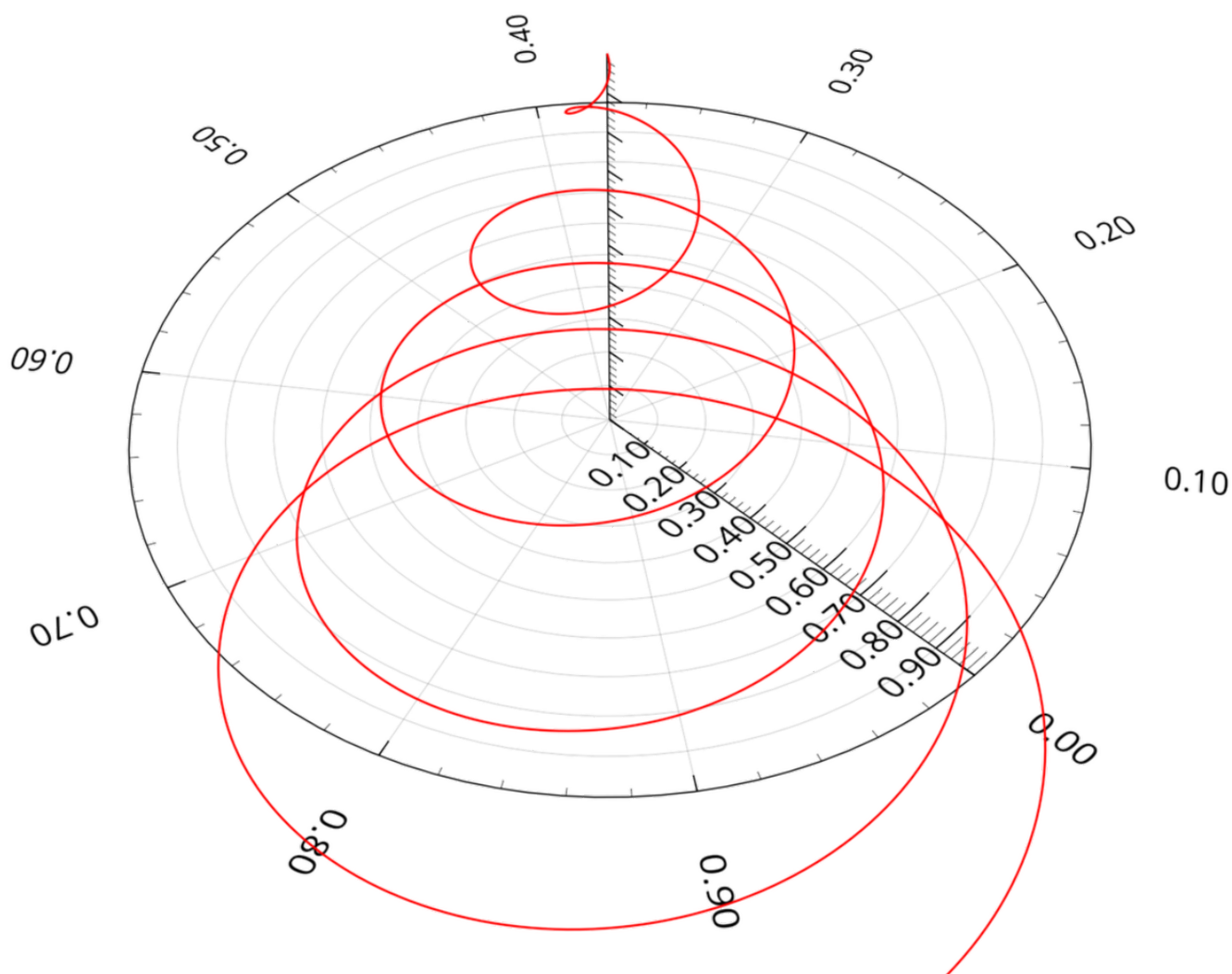
One has no future if one couldn't teach himself^[1].

自学是门手艺

没有自学能力的人没有未来

作者：李笑来

特别感谢霍炬（@virushuo）、洪强宁（@hongqn）两位良师诤友在此书写作过程中给予我的巨大帮助！



pseudo-code of selfteaching in Python

```
def teach_yourself(anything):  
    while not create(something):  
        learn()  
        practice()  
    return teach_yourself(another)
```

```
teach_yourself(coding)
```

请先行阅读 [T-appendix.jupyter-installation-and-setup](#) 以便在本地安装 [Jupyterlab](#) 而后用更好的体验阅读本书。

有兴趣帮忙的朋友，请先行阅读 [如何使用 Pull Request](#) 为这本书校对。

2019 年 3 月 23 日，新增 Markdown 版本：

<https://github.com/selfteaching/the-craft-of-selfteaching/tree/master/markdown>

目录

- [01.preface](#)（前言）
- [02.proof-of-work](#)（如何证明你真的读过这本书？）
- [Part.1.A.better.teachyourself](#)（为什么一定要掌握自学能力？）
- [Part.1.B.why.start.from.learning.coding](#)（为什么把编程当作自学的入口？）
- [Part.1.C.must.learn.sth.only.by.reading](#)（只靠阅读习得新技能）
- [Part.1.D.preparation.for.reading](#)（开始阅读前的一些准备）
- [Part.1.E.1.entrance](#)（入口）
- [Part.1.E.2.values-and-their-operators](#)（值及其相应的运算）
- [Part.1.E.3.controlflow](#)（流程控制）
- [Part.1.E.4.functions](#)（函数）
- [Part.1.E.5.strings](#)（字符串）
- [Part.1.E.6.containers](#)（数据容器）
- [Part.1.E.7.files](#)（文件）
- [Part.1.F.deal-with-forward-references](#)（如何从容应对含有过多“过早引用”的知识？）
- [Part.1.G.The-Python-Tutorial-local](#)（官方教程：**The Python Tutorial**）
- [Part.2.A.clumsy-and-patience](#)（笨拙与耐心）
- [Part.2.B.deliberate-practicing](#)（刻意练习）
- [Part.2.C.why-start-from-writing-functions](#)（为什么从函数开始？）
- [Part.2.D.1-args](#)（关于参数（上））
- [Part.2.D.2-aargs](#)（关于参数（下））
- [Part.2.D.3-lambda](#)（化名与匿名）
- [Part.2.D.4-recursion](#)（递归函数）
- [Part.2.D.5-docstrings](#)（函数的文档）
- [Part.2.D.6-modules](#)（保存到文件的函数）
- [Part.2.D.7-tdd](#)（测试驱动的开发）
- [Part.2.D.8-main](#)（可执行的 **Python** 文件）
- [Part.2.E.deliberate-thinking](#)（刻意思考）
- [Part.3.A.conquering-difficulties](#)（战胜难点）
- [Part.3.B.1.classes-1](#)（类 —— 面向对象编程）
- [Part.3.B.2.classes-2](#)（类 —— **Python** 的实现）
- [Part.3.B.3.decorator-iterator-generator](#)（函数工具）
- [Part.3.B.4.regex](#)（正则表达式）
- [Part.3.B.5.bnf-ebnf-pebnf](#)（**BNF** 以及 **EBNF**）

- [Part.3.C.breaking-good-and-bad](#)（拆解）
- [Part.3.D.indispensable-illusion](#)（刚需幻觉）
- [Part.3.E.to-be-thorough](#)（全面 —— 自学的境界）
- [Part.3.F.social-selfteaching](#)（自学者的社交）
- [Part.3.G.the-golden-age-and-google](#)（这是自学者的黄金时代）
- [Part.3.H.prevent-focus-drifting](#)（避免注意力漂移）
- [Q.good-communication](#)（如何成为优秀沟通者）
- [R.finale](#)（自学者的终点）
- [S.whats-next](#)（下一步干什么？）
- [T-appendix.editor.vscode](#)（**Visual Studio Code** 的安装与配置）
- [T-appendix.git-introduction](#)（**Git** 简介）
- [T-appendix.jupyter-installation-and-setup](#)（**Jupyterlab** 的安装与配置）
- [T-appendix.symbols](#)（这些符号都代表什么？）

关于 [.ipynb](#) 文件转换为 [.md](#) 文件的备注：

需提前安装 nbconvert 插件，Terminal 下执行：

```
$ jupyter nbconvert --to markdown *.ipynb
```

而后将所有 [.md](#) 文件移到 markdown/ 目录之下 — 除 README.md 文件之外

README.md 文件复制一份到 markdown/ 目录之下，而后编辑为当前文件

需使用 VSCode 批量 Find and Replace：

将所有 (Part.1.A.better.teachyourself_files/ 替换为 (../images/

将所有 (Part.1.E.6.containers_files/ 替换为 (../images/

将所有 ```\n\n 替换为 ```\n

将所有 \n\n``` 替换为 \n```

将所有 .ipynb) -> .md

将所有 (images -> (../images)

Part.1.E.3.controlflow.md 文件中有过长的 output 需要编辑

Part.1.E.7.files.md 文件中有过长的 output 需要编辑

本书的版权协议为 [CC-BY-NC-ND license](#)。



署名-非商业性使用-禁止演绎 3.0 未本地化版本 (CC BY-NC-ND 3.0)

脚注

[1]: ‘Themselves’ or ‘themselves’? -- Oxford Dictionary

[↑Back to Content↑](#)