

学编程，怎么学？ 用 AI，怎么用？

计算机通修类课程与 AI 使用

李亦凡

24 级管理学院统计系



自我介绍

李亦凡

24 级管理学院
统计系

其他履历与分享资料
见个人主页

lunarkn1ght.github.io

人工智能数学原理与算法 A

本学期助教

“知之道”极客中心

参与人工智能领域的研究

大创项目

研究智能体在量化因子挖掘中的应用

同一个 Hello World, 五种写法

写法各不相同, 实现的功能却一样

Java

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

C / C++

```
#include <stdio.h>  
int main() {  
    printf("Hello, World!");  
    return 0;  
}
```

Python

```
print("Hello World")
```

R

```
cat("Hello, World!")
```

JavaScript

```
console.log("Hello, World!");
```

学习编程, 要抓住自己想实现的程序逻辑

编程思维：拆分问题

把一句模糊的人话，拆分成一组
明确、可执行、可检查的步骤

先弄清楚要做什么

需要哪些信息？
按什么规则处理？
希望得到什么结果？

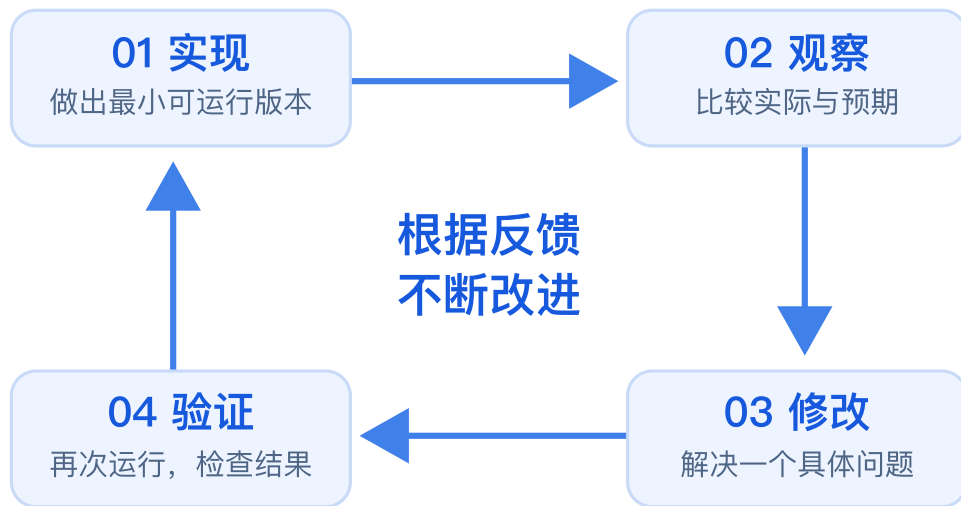
再想清楚怎样执行

第一步做什么，下一步做什么？
哪些操作需要重复？
没有符合条件的结果怎么办？

循环和条件判断，都可以对应到现实问题里的具体步骤

工程思维：迭代演进

接受第一版并不完美，再根据反馈不断把它改对



向 AI 求助前

我原本希望得到什么？

实际出现了什么？

报错信息是什么？

AI 可以帮助定位和修改，最后仍然要由我们**运行、检查和验证**

在练习中锻炼思维

一门语言不是一下就能学会的
两种思维也需要在动手中慢慢培养

基础题与刷题

熟悉语法，练习局部代码
从循环、数组等基础内容开始

洛谷 / 力扣

自己想用小项目

构思一个小玩具，借助 AI 逐步迭代
一个数据处理工具，也有价值

需求拆解 / 运行验证 / 根据反馈修改

没必要上来就做高大上的东西，先把一个小问题解决好

AI-A/B：内容很多，应该怎么学？

统计学习、神经网络、图神经网络、Transformer、强化学习……

如果我们也跟着东拼西凑地学，效果肯定不好

解决什么问题？

模型为什么会出现

输入输出是什么？

它究竟在处理什么

为什么这样设计？

关键步骤起什么作用

“结合上面的叙述，给后来者一个警醒：速通这门课的想法是不切实际的。每个章节的概念尽量还是平时背一背，考前只花一两天背诵整本书压力太大。计算题还是有迹可循的，课程涉及的算法并不困难。”

——LUNARKNIGHT 评课

数学知识，真的能用起来

微积分、线性代数、概率论，经过建模和编程
竟然能构建出“智能”的模型

作为学生的感受

原来之前学的数学是真实有用的！

庞大的人工智能，也可以拆开来理解

作为助教的期待

每个模块为什么要这么设计？

为什么要经过这样的处理？

它们都解决了什么问题？

在编写实验的过程中，也去想一想每一步背后的理由

助教的一点吐槽：电脑基本功

不需要一次性学会，可以在日常使用过程中逐步掌握

类别	建议掌握的内容
文件管理	文件夹、路径与扩展名；查找、重命名、移动文件；压缩与解压
常用操作	复制、粘贴、撤销、保存、查找等快捷键；截图、分屏与窗口切换
文档与 PDF	标题、段落与分页排版；导出 PDF 并检查格式
电子表格	简单公式、求和、排序与筛选
软件与存储	区分文件、软件与操作系统；理解内存与磁盘、本地与云端存储的区别
备份与恢复	重要文件备份、版本保留；回收站与版本历史的使用
计算机硬件组成	了解 CPU、内存、硬盘、显卡、主板与常见外设的基本作用
Markdown（按需）	标题、列表、链接、引用与代码块
Git（按需）	查看修改、提交记录、对比版本；协作时学习分支与合并

小麻烦，也值得解决

AI 作为工具

能大幅降低

我们日常工作的成本

这里的成本，主要是**时间和精力**

整理材料、修改格式、尝试方案.....

省下来的精力，可以投入更有价值的事情



从日常的小麻烦里发现需求

最近有没有哪件事，反复花时间处理，却总觉得麻烦？

群里的通知 明明看过了，还是会漏掉要交的材料

笔记与文件 记了不少，复习时却找不到要看的文件

重复的格式 每次处理一份表格，都要再改一遍相同的格式

AI 作为工具，就一定在具体的场景来使用

借助 AI，把需求变成可以行动的结果

拿整理文件这件小事来说

先说清楚自己的需求

哪些内容找起来最麻烦？

希望按什么方式整理？

哪些内容需要保留原样？

再看结果是否符合预期

整理后，能找到想要的文件吗？

原有内容和链接还正常吗？

哪里不合适，需要继续修改？

我们决定要求，AI 帮忙执行，再由我们[检查和改进](#)

借助 AI 学习

可以反复问，也可以根据自己的需求
让 AI 帮我们**查漏补缺**

01 先自己尝试

找到具体卡住的位置

03 验证回答

代码自己运行，概念对照教材

看懂了一份答案，还要检验自己能不能用出来

02 让 AI 提示和解释

指出问题，补充自己需要的例子

04 关掉答案再做一次

用一道类似题，检查能否独立完成

AI 使用边界

AI 帮我写完了，可是过两天
我又看不懂这些东西了，甚至完全不会

别产生“我已经会了”的错觉

善用 AI，引导 AI
而不是把 AI 当许愿机

关键步骤能不能解释？
结果有没有价值，能不能判断？

我的体会：“AI 的表现能力和使用者能力成正比”
自己的知识和判断力，会影响如何提出要求、评审结果

AI 仍然会犯错

自己可以不做每一步具体内容
但一定要**评审 AI 做的东西**

运行、核对，必要时继续查证

把琐碎的工作省下来

AI 是工具

编程思维

把模糊的问题，拆成可执行的步骤

工程思维

根据反馈不断修改，把结果验证清楚

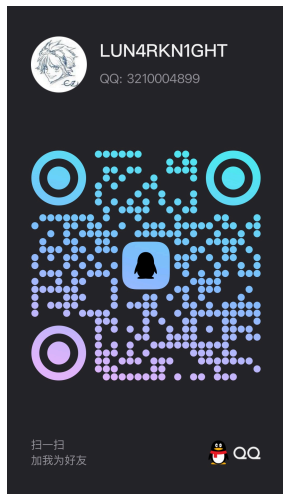
人机协作

借助 AI 降低成本，自己保留理解和判断

把时间和精力，投入**更有价值的事情**，展现自己的才华

谢谢大家，欢迎交流！

有问题，我们可以接着聊



QQ 联系方式



USTC-Agent 交流小队



个人主页 lunarkn1ght.github.io

个人仓库 github.com/LUNARKN1GHT

AI 笔记 / 助教讲义 [USTC-AI-Notes](#)

评课社区 [AI-A](#) [程设 A](#) [数据结构 A](#)