# 1. 学写代码的方式

对于从未接触过Python，或者对Python代码有些了解的初学者，究竟如何进入到Python代码的世界里，并能够应用于自身的专业解决实际的问题？

* 调查目前为止所有的学习方式，主要包括：

1. 教材类。对Python语言系统的阐述，从数据类型结构至基本语句、及函数和类，到标准库和扩展库，最后为高级的专项，例如机器学习、深度学习等。此类方法多出现在系统性阐述的教材（图书）中。该类方法通常适用于具有较强自修能力的学习者，可以从头阅读教材，逐行的敲入代码跟随练习；
2. 在线视频课程。该类视频课程教学者通常会选择一本教材，或者自行编写教学大纲、教案进行讲解。此类的教学途径通常适合于自修能力相对较弱的学习者，可以从教学者的讲解中，直接获取信息，减少阅读教材信息再提取加工的过程；
3. 在线代码交互练习。可以跟随章节进度，对照案例或者提示在线敲写代码，并实时反馈结果。这种学习的方式，因为互动信息反馈，具有游戏性，初学者往往倾向于优先选择此类途径；
4. 表格形式的卡片。此类方式可以理解为对系统阐述内容的教材类重要内容的提取，转换为表格卡片的形式，这么处理的好处是去除一些目前不必要，日后需要再了解的内容，将更多精力用于重点核心内容上；
5. 练习形式。类似表格卡片形式，将传统系统阐述的内容分解为几十，甚至更多的练习小节，结合文字阐释，强调练习的重要性；
6. 自由组合上述的教学途径。通过使用不同教学手段，可以更综合的调用学习者的学习兴趣，提高学习效率和学习效果。

* 调查目前为止Python教学的主要示例内容，主要包括：金融类、网页类、计算机（软件开发类和嵌入式系统）、数据分析类（含金融类，地理信息和大数据等）、算法类、游戏类，艺术类等。

根据当前Python教学途径，及示例内容，对于城乡规划、风景园林、生态规划（林业、水利，湿地、草原、海洋）及建筑等归属自然资源部的相关专业，在开展Python编程语言学习，解决本专业的问题时，示例的内容最好能够更贴合专业内容，而学习方式上则可以任意组合。

## 1.1 示例数据以专业相关为主

如果能尽量以本专业（规划设计）相关的数据作为Python学习的数据内容，以处理本专业问题为导向学习代码的基本核心，这将能够更好的引导，学习Python基础知识后，不知道如何应用Python处理专业问题的弊病。同时，Python基础核心部分内容是为没有或者基础较弱的读者提供阅读《城市空间数据分析方法》的先导知识补充。

在示例数据选择上优先考虑《城市空间数据分析方法》基础实验和专项研究部分的内容，如果数据无法满足要求，或者不能很好的解释待说明的知识点，将会调整选择的数据，仍然尽量保持为专业相关数据。

## 1.2 结合表格卡片形式的知识点内容

基础核心除了作为先导知识补充之外，仍需要解决两个问题，其一，可以作为知识点查询工具，当遗忘或寻找解决问题的方法时，可以从这些卡片中搜索定位；其二，知识点内容可以不断拓展更新，这不仅包括基础知识点，还会增加标准库及扩展库部分，任何有价值的算法或解决问题的逻辑，这将有助于不断积累实践中会用到的各类知识内容，也避免同类问题重复解决。

表格卡片的形式，能让读者可以根据卡片表格形式，增加自己关注的知识点，不断积累。通常，一组内容组成一张卡片，这种将知识切分成微内容（碎片）的方式，能够让初学者每完成一张卡片，就会产生一定的成就感，有意愿不断的学习每张卡片，并可以根据自己的时间弹性调整学习时段；根据自己掌握的水平，挑选适合的卡片；根据要解决的问题，搜索查询卡片内容。

卡片完成的数量，可以作为衡量代码水平（完成度）的因素之一。每完成部分卡片，就该部分内容可以增加小节测试题，作为衡量代码水平的又一因素。综合卡片数量和测验，制定可以衡量学习者编码水平的分数级别，这有助于帮助初学者粗略了解自己的代码水平，及同届中同学间的水平差异，激励初学者不断学习，提高代码书写水平。

## 1.3 搜索与英文

在代码世界中，通常用google搜索引擎，能够较为方便准确的定位到待搜索的答案，从而快速解决问题，节约时间。同时，Python编程语言为英语书写，因此在写作该部分内容时，对于主要的词汇会同时给出英文，方便英文搜索时使用，并接轨英文这一国际通用语言。

## 1.4 完成度自测

| 等级 | ID | 卡片名称 | 得分（是否完成） | 备注 |
| --- | --- | --- | --- | --- |
| Ⅰ 级 | 1 | PCS\_1 善用print()，基础运算，变量及赋值 | 10（ ） |  |
|  | 2 | PCS\_2 数据结构-list\_tuple\_dict\_set | 10（ ） |  |
|  | 3 | PCS\_3 数据结构-string | 10（ ） |  |
|  | 4 | PCS\_4 基本语句-if\_for\_while\_comprehension | 10（ ） |  |
|  | 5 | PCS\_5 函数-def\_scope\_args | 10（ ） |  |
|  | 6 | PCS\_6 函数-recursion\_lambda\_generator | 10（ ） |  |
|  | 7 | PCS\_7 模块与包及发布-module\_package\_PypI | 10（ ） |  |
|  | 8 | PCS\_8 (OOP)类Classes-定义，继承，**init**()构造方法，私有变量/方法 | 10（ ） |  |
|  | 9 | PCS-9 (OOP)类Classes-Decorators(装饰器)\_Slot | 10（ ） |  |
|  | 10 | PCS-10 异常-Errors and Exceptions | 10（ ） |  |
| 总计 |  |  | 100（ ） |  |

注：Python 基础核心 以表格卡片的形式写作，主要受到 [Coffee Break Python](https://coffeebreakpython.com/)[1]一书，及配套在线交互学习[finxter-puzzle training](https://app.finxter.com/learn/computer/science/)①的影响。该作者在书中阐述了基于谜题学习方法（Puzzle-based Learning）对学习者的有效性，从Overcome the Knowledge Gap, Embrace the Eureka Moment, Divide and Conquer, Improve From Immediate Feedback, Measure Your Skills, Individualized Learning, Small is Beautiful, Active Beats Passive Learning, 及Make Code a First-class Citizen等多个方法阐述了其价值。就Python 基础核心而言，采纳了表格卡片的形式，并借鉴了评级方式，卡片的内容则就规划设计专业和作为《城市空间数据分析方法》的先导知识补充，以及合理性做出了调整。

注释（Notes）：

① finxter-puzzle training，（<https://app.finxter.com/learn/computer/science/>）。

参考文献（References）:

[1] Mayer, C. Coffee break python: 50 workouts to kickstart your rapid code understanding in python[M]. September 16, 2018.

[2] Shaw, Z. A. Learn python3 the hard way: A very simple introduction to the terrifyingly beautiful world of computers and code[M]. Addison-Wesley Professional,June 26, 2017.

[3] Shaw, Z. A. Learn more python the hard way: The next step for new python programmers[M]. Addison-Wesley Professional, September 1, 2017.

[4] Publishing, A. Python programming for beginners: The ultimate guide for beginners to learn python programming[M].October 26, 2022.

[5] Rao, B. N. Learning python[M]. CyberPlus Infotech Pvt. Ltd, February 14, 2021.

[6] Snowden, J. Python for beginners A practical guide for the people who want to learn python the right and simple way[M]. Independently published, December 2, 2020.

[7] 包瑞清; 学习PYTHON—做个有编程能力的设计师, 江苏凤凰科学技术出版社, 2015