

# Teaching Practice of Python Programming Course in Big Data Era

Qing Liu<sup>a,\*</sup>

*School of Statistics and Information, Shanghai University of International Business and Economics, Shanghai, 201620*

**Abstract.** The era of big data poses new challenges and higher requirements for students majoring in economics and management. The programming course should also adapt to the requirements of the era of big data. From the five aspects of content reorganization, comparative teaching, case teaching, WeChat help, and practical teaching, the teaching practice methods of this course are expounded. Through the study of programming courses, students' professional qualities have been improved, and their competitiveness has been strengthened.

## 1 Introduction

The teaching reform of computational thinking needs to understand the temporal nature of computational thinking, that is, computational thinking is a scientific thinking based on computer technology. Teaching reform must conform to and adapt to the development stage of computer technology. [1] Nowadays, we are in an era where new knowledge and new technologies are emerging continuously. The technologies of cloud computing, Internet of Things, data mining, machine learning, and deep learning are applied to all walks of life. On March 17, 2016, according to The Outline of the Thirteenth Five-Year Plan for National Economic and Social Development of the People's Republic of China, Section 27 "Implementation of National Big Data Strategy" in Chapter 6, the expanded network economic space, puts forward: Big data is used as a basic strategic resource to fully implement actions to promote big data development. Report of the McKinsey Global Institute [2] also points out that effective capture, analysis, visualization, application of big data to insight and achieve business objectives will help enterprises stand out from the fierce competition. The emergence of big data will definitely subvert the traditional way of data management. It will bring revolutionary changes to data sources, data processing methods and data thinking. [3] Changes in data management methods will inevitably lead to revolutionary changes in the way talents are trained.

In the context of such big data, teaching related knowledge such as data acquisition, data analysis, and machine learning becomes indispensable. And as the basis of these knowledge - programming courses for

computational thinking must redesign the teaching content and teaching programs.

## 2 Python programming and computational thinking

In the context of big data, Matlab, SPSS, SAS, Python, R, etc. are good analysis tools among many analysis tools. Python has passed C# and Visual Basic.NET for the fourth place in TIOBE's October 2018 programming language rankings. Python is a high-level programming language with interpreted, object-oriented, dynamic data types. It is simple, easy to learn, portable, extensible, and embeddable. In addition to the large standard library, there is a very rich third-party library. In July 2008, in the article "Computational Thinking and Thinking About Computing", Professor Zhou pointed out: The essence of computational thinking is abstraction. Deeper computational thinking will help us not only to model more and more complex systems, but also to analyse the massive amounts of data we collect and generate. [4] Python's programming model is also very compatible with human thinking habits, which helps us to integrate computational thinking into the teaching of Python programming.

## 3 Teaching practice of python programming

### 3.1 Content reorganization and difficult adjustment to be in line with big data era

In the past programming courses, we always start with

<sup>a</sup> Corresponding author: [author@e-mail.org](mailto:author@e-mail.org)

\*Fund project: 2018 School-level Curriculum Construction Project of Shanghai University of International Business and Economics

About the author: Liu Qing (1973-), Female, Wuhan, Hubei, Associate Professor, Master, Research direction: database technology, data analysis.

constants, variables and expressions, and then explain the three structures of sequence, selection and loop. In the process of explaining these three structures, classical algorithms such as the greatest common divisor, completeness, and prime numbers are often used to organize teaching, with the loop is extremely in-depth. In fact, this does not match the language features of Python. We don't need to go too deep into the loop. Students only need to know how to use the loop, understand the similarities and differences between for loop and while loop, the differences between break and continue, and they can compile simple loops by themselves. There is no need to study complex algorithms. Because the Python community has a very shareable community of Python enthusiasts who upload a lot of packages, we just take them to use according to the introduction. Moreover, too much research on algorithms takes up a lot of time, which may lead to more important follow-up content without time to explain. After finishing the three structures, the contents of common data structures such as lists, tuples, dictionaries, collections, strings and so on are followed. In the past, all the methods were explained in great detail, while useful comprehensions and generators were not involved. Now we will add list comprehension, dictionary comprehension, set comprehension, generator and so on. We will first mention various methods, and then introduce them in detail when we use them; When we talk about strings, we often only talk about character formatting methods, escape characters and common functions, but the useful string module, JSON module and regular expression are not involved. And this part is an indispensable part of learning data acquisition, web crawler and so on. We should include these contents in strings; After talking about common data structures, it comes to functions, files, and object-oriented content. In the past, only text files were discussed in the file, but the binary file was not mentioned. In the file section we add explanations for the openpyxl, struct, pickle, marshal, and shelve modules.

### 3.2 Comparative teaching, from the surface to the essence

In the programming course, some concepts are similar but not identical, and students need to figure out the similarities and differences such as iterable objects and iterators. On the surface, objects that are traversed by for... in... are iterable objects. Then lists, tuples, strings, and dictionaries are all iterable objects. In fact, if an object implements the `__iter__()` method, then this object is an iterable object. So how do we determine whether the list object is an iterable object? Through `help()`, as shown in Figure 1, we find that the `__iter__()` method is implemented in the list class, so the list object is an iterable object. Similarly, we can judge whether other objects are iterable objects in this way. We can also judge by the built-in function `isinstance()`, as shown in Figure 2. For iterators, the `__iter__()` method and `__next__()` method are implemented, and the object that can continuously return the next value through `__next__()` method is called iterator. Iterators can also access the next

element through the built-in function `next()`. In the same way, we know that lists, tuples, strings, and dictionaries are iterable objects rather than iterators. Furthermore, they can get an iterator through the `iter()` function. According to this principle, we can also judge that enumerate object and zip object are both iterable objects and iterators.

```
>>> help(list)
Help on class list in module builtins:

class list(object)
| list(iterable=(), /)
|
| ..:::
|
| __iter__(self, /)
|     Implement iter(self).
```

Figure 1. Checking the list class

```
>>> from collections.abc import Iterable
>>> isinstance(['abc',1,8.5],Iterable)
True
>>> isinstance(('abc',1,8.5),Iterable)
True
>>> isinstance({'1':'one',2:'two'},Iterable)
True
>>> isinstance({'one','two'},Iterable)
True
>>> isinstance('abcdefg',Iterable)
True
>>> isinstance(range(10),Iterable)
True
>>> isinstance(123,Iterable)
False
```

Figure 2 Judging by built-in function `isinstance()`

### 3.3 Case teaching, training students' computational thinking ability

Case teaching is based on case analysis, with the solution of reasons as the main line, and the teaching content is gradually expanded around the discussion and analysis of cases. Through the analysis of the case, the students' curiosity is stimulated, the enthusiasm of the students is mobilized, and the students actively think and learn.<sup>[5]</sup> Computational thinking is to help students simulate the thinking of computers to think about ways and methods to solve real problems, and to improve students' ability to analyze and solve problems.<sup>[6]</sup> In case teaching, we use a series of methods of computational thinking (problem reduction, separation of concerns, simulation, heuristic reasoning, recursive thinking, backstepping, etc.) to design a series of cases, including crawler series and file database series. Crawler is a script or program to automatically capture data in network according to certain rules and we mainly explain web-based crawling. In the crawler series, requests and urllib modules are frequently used in the data acquisition phase; When parsing and matching data, regular expressions, BeautifulSoup, and

lxml all have their own characteristics. When saving data, the knowledge of files and databases will be used. We design a case to crawl the film review data from the Douban net. Taking "Wolf Warriors" as an example, we can crawl film review information, analyze film reviews, make word clouds, and analyze the reasons for good and bad film reviews. We use three analytical methods to analyze, focusing on lxml, first crawling a page, then expanding to multiple pages, and then storing the crawled data into a csv file or a MySQL database. This allows students to follow the example and crawl the data to suit their needs. Of course, it's just a simple start. If we encounter a website with a strong anti-crawler mechanism, we need to learn something else by ourselves. In the file database series, we take student management as a case, first store the entered data in a file, and perform the operations of adding, deleting, and searching in the file; After that, the file system is transformed into a database. Finally, it is implemented in an object-oriented way. Through these three levels of transformation learning, students have made clear the differences between files and databases, modular programming and object-oriented programming.

### **3.4 Building micro-video and pushing the understanding of knowledge points and case ideas through WeChat**

The Internet era is in the ascendant, and the we-media era has emerged. The popularization of mobile Internet has brought a new change to the behavior and habits of young college students [7]. Social software such as WeChat, micro-blog and so on are very popular and almost everyone pays close attention. We use this feature to promote micro-video and background knowledge and theoretical knowledge of the case through WeChat to enhance students' understanding of relevant knowledge. For example, we make micro-videos of three knowledge points: comprehension, generator, and yield statement. Students further learn that tuples do not have comprehensions but have generator through micro-video. The result of generator is a generator object, which has the characteristics of lazy evaluation and is suitable for big data processing. When we talk about functions, we combine the characteristics of generator objects with functions containing yield statements, and explain the similarities and differences between yield statements and return statements: The similarity is all are used to return a value from a function; The difference is that the return statement ends as soon as the function is executed. Each time the yield statement is executed and a value is returned, the execution of the subsequent code is suspended, and the execution is resumed the next time through the generator object's `__next__()` method, the built-in function `next()`, the loop, and so on. When learning files and database cases, we will push the organization of the files, coding, the storage structure of the database, SQL statements, common database products, paradigms and other related knowledge. In this way, students can also use sporadic time to strengthen the study of knowledge, deepen their understanding, and gain

a lot of useful knowledge over time.

### **3.5 Practice teaching to further enhance students' computational thinking ability**

From the point of view of modern learning psychology theory, the development of thinking is an emotional process, and it is difficult to realize the transfer of students' thinking ability without "preaching" education in the real situation [8]. After the students have passed the case teaching in the real situation, their thinking ability is strengthened, and they can use what they have learned to complete the practice project. Computational thinking has a strong ability to innovate, and the highest goal of developing computational thinking is to innovate [9]. We divide the students into small groups of 3 to 5 people to complete the practice projects together. For example students can crawl user reviews from an online store, analyze the reasons for praise and bad reviews, and get suggestions to help companies improve their services; Students can download data on house prices from the official website of the National Bureau of Statistics of China, analyze the average house price, and select multiple variables such as gross domestic product and national consumption level for multivariate regression analysis. In the process of completing the practice project, students are likely to get different and better methods by consulting materials and combining with what they have learned. After the completion of the project, students' self-evaluation and teachers' comments not only affirm what they have done well, but also point out the shortcomings, so that students can better improve and develop. At the same time, the good project of this session can be regarded as the case of the next session.

## **4 Conclusion**

Computer programming is the basic course to learn to solve problems using computers. Whether computer programming can be used to solve problems is not only the proficiency of language, but also the ability of computational thinking<sup>[10]</sup>. In universities, through programming courses, students' thinking ability is exercised, and students' basic professional ability and sustainable development ability are also trained. Practice has proved that through the study of this course, the level of students using the knowledge of programming to solve professional problems has been significantly improved, the importance of teamwork has been understood, and the competitiveness has been enhanced. At the same time, in the process of teaching this course, teachers have perfected their own knowledge structure, and their teaching level has also been constantly improved.

## **References**

1. Song Tian, Huang Tianyu, Li Xin. Python Language: Ideal Choice for Teaching Reform of Programming Course [J]. China University Teaching, 2016, 2: 42-47.

2. [http://cdn.oreillystatic.com/en/assets/1/event/71/Big%20Data\\_%20The%20Next%20Frontier%20Presentation.pdf](http://cdn.oreillystatic.com/en/assets/1/event/71/Big%20Data_%20The%20Next%20Frontier%20Presentation.pdf)
3. Meng Xiaofeng, Ci Xiang. Big Data Management: Concepts, Techniques Challenges [J]. Journal of Computer Research and Development, 2013, 50 (1): 146-169.
4. Wing JM. Computational Thinking and Thinking About Computing [J]. Philosophical Transactions of the Royal Society, 2008, 366: 3717-3725.
5. Zhang Dianlong, He Dongguang, Jia Lijuan. Design of Teaching Plan for Cultivating Computational Thinking Ability in Programming Course [J]. Computer Education, 2015 (16): 78-80.
6. Wang Yaping. Teaching Reform Practice of "Python Programming" Course from the Perspective of Computational Thinking [J]. Computer Knowledge and Technology, 2018, 14 (02): 143-144.
7. Lu Ning. Analysis on the Thought of "Ideological and Political Courses" and "Course Ideological and Political" in University [J]. University Education, 2018. 1: 122-124.
8. Li Feng, Wang Jiqing. Computational Thinking: An Intrinsic Value of Information Technology Courses [J]. China Educational Technology, 2013, (8): 19-23.
9. Mou Qin, Tan Liang. Research and Progress in Computational Thinking[J]. Computer Science, 2011, (3): 10-15.
10. Yang Xiaohua, Liu Jie, Liu Zhiming, Xu Zhuoran. Discussion on the Teaching Method of Programming Design Integrating with Computational Thinking Training [J]. Journal of Higher Education, 2015 (19): 106-109.