Universe Scientific Publishing

ORIGINAL ARTICLE

Application of Jupyter Notebook in the Teaching of Programming Courses (Python)

Muyi Chen

School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang 110159, Liaoning, China.

Abstract: In the teaching of computer language and programming courses, PowerPoint (PPT) still dominate the classroom teaching process. Teachers usually employ PPT courseware to teach grammar and related content, and screenshots of code segments and running results are given in PPT. The disadvantage of this method is that students cannot see the actual program execution process and the program design process, so it is not easy for the students to understand the teaching content and review after class, it is even impossible for the students to understand how to write programs from scratch. Aiming at the problems of using PPT to teach programming courses in domestic colleges and universities, a new software named "Jupyter Notebook" is used to reform the teaching method. The method of teaching reform is introduced, and the benefits of the new approach is described. Compared to traditional PPT, the advantage of Jupyter Notebook at different learning stages is summarized. This method is very suitable for the teaching of programming courses such as Python.

Keywords: Jupyter Notebook; Programming Teaching; Python

Computer language programming courses in domestic colleges and universities mainly teach traditional programming languages such as C, JAVA, C+ + , etc. The PPT software and related teaching method is generally used in teaching. But this method has the following shortcomings:

(1)The teaching content in PPT is separated from the code running and debugging operation, which increases the difficulty of understanding for beginners.

(2)In the basic grammar teaching process, the sample code is usually scattered as small code blocks that are not related to each other. If we directly show the source code file and pass it to the students with PPT, the students may make mistakes or encounter errors when directly executing the code. Besides, different grammar knowledge and important techniques are mixed together in the source code, the output results of are also mixed together, which is not convenient for observation and understanding for the students. If we pack multiple files together, each includes an individual teaching unit, and pass them to students, many students will not bother to open the files one by one and give up the experiment. Even the highly motivated students may just run the code file once and think that the learning is complete after the results are obtained successfully.

(3)When students review the contents, they often only look at the PPT and do not do programming practices by themselves, they probably have the illusion that they understand all the contents, but in fact they have not. Only those students with a strong willingness to learn actively will copy the code in the PPT and paste them into the programming environment for experimentation. Most of them just run the program once and get the results and then stop learning.

In order to solve the above problems, we explored the use of Jupyter Notebook to reform Python language teaching.

1. Reform teaching methods using Jupyter Notebook

Jupyter Notebook is a web-based application that can be used to take notes. It is simple and easy to learn. It integrates

doi: 10.18686/ahe.v5i2.3309

functions such as text, pictures, and interactive code execution. By organizing these elements and content into the same web document, users can build flexible powerful notes.

Jupyter Notebook uses the programming method proposed by the world's top computer scientist Donald Knuth——literary programming: it allows people to develop programs according to their own thinking logic. To put it simply, we have transitioned from writing code that the machine can understand to explaining to people how to make the machine realize our ideas. In addition to the code, there are more narrative texts, diagrams and other content.

During teaching a programming language, a teacher must not only be a good programmer, but also a good writer. Through storytelling, the teaching process is more vivid and concrete. Jupyter Notebook is an efficient tool that integrates programming and writing, which can be greatly improve and enrich the student's learning experience.

In teaching, Jupyter Notebook integrates teaching content (including text and pictures, etc.) and code segments that can be interactively executed in real time into one document, and performs live demonstrations. You can also add experiments, run the code and observe the results at any time according to the students' reaction in classroom to deepen their understanding.

2. Interactive teaching by Jupyter Notebook

(1) Jupyter Notebook integrate all required teaching elements (titles, texts, diagrams, code blocks that can be interactively executed in real time, and running results) in the same page, which is clear and intuitive, convenient for experiments, reduces the difficulty of understanding, and helps to enhance students' interest in learning and positivity.

(2)It is convenient for dynamic exploration in time and recording in class, teacher can initiate new explorations conveniently, and the teaching contents can be better reviewed after class.

For important knowledge points, teachers can directly enter the code snippetsin Jupyter Notebook and execute it in the classroom. On one hand, students have personal experience and see the design process and the execution process of the code segment. On the other hand, this dynamic adaption also slows down the teaching and learning process, introduces a more flexible classroom rhythm, and leaves enough time for students to think, observe, experience, and remember. In addition, the current theoretical research and practice of the brain's learning process have shown that if multiple senses can be mobilized at the same time, it will be more conducive to learning and memory students hear the keyboard tapping, see the dancing characters on the screen, observe the program running and debugging, the classroom experience is enriched, and it is more conducive to their learning, memory and understanding.

(3) Students can also easily observe and understand how complex programs are built by simple code segments step by step, instead of just getting a final source code file and wondering how, which is very helpful for students to truly understanding the method of writing programs.

In traditional teaching of programming courses, small code fragments and long complete programs are provided for students, but the students lack the knowledge and experience of how to construct complete programs from small code fragments. They do not know how could the transition happen? They don't know how to accomplish this and soon give up, so they lack the ability to solve practical problems.

(4) In order to avoid the common situation during learning, that students only looking at the courseware without hands-on practice, the Jupyter Notebook interactive execution software is adopted, so that students can interactively execute corresponding code segments on the same webpage in real time while watching text and pictures to understand the teaching content. Observations, experiences and experiments are conveniently made.

3. Teaching reform methods and effects

Compared with PPT teaching, great results have been achieved by adopting Jupyter Notebook.

3.1 Teaching on class

3.1.1 Adopt Jupyter Notebook

Students can get a complete story. The code blocks and the text and diagrams required by the teaching content are organized in the same page, and the code can be executed interactively in real time, which is clear and intuitive. The text, pictures, each intermediate step code, and the execution result appear on the page in the form of a markdown note block or executable code block.

Students can follow the courseware documents to see the entire thinking evolution process and the program design process from scratch, they can also experiment directly by themselves, see the intermediate execution results of each step, understand the program's step-by-step construction process from simple to complex, observe how the program evolves from prototype to perfection, from error to correction, that is, the whole programming iterative process. By this teaching method, what students see is not just a

result, but a complete story.

It is convenient to add more sample code blocks to the courseware based on the students' classroom responses and questions they asked, and run them interactively in real time and save them in the courseware to create a better learning experience for students.

3.1.2 Adopting PPT

It is suitable for displaying text and charts, but the code cannot be executed interactively in real time. The text and picture-based teaching content is separated from the executable code in different documents, which causes difficulties for the students to understand and separates teaching and experimentation.

3.2 Self-study after class

3.2.1 Adopting Jupyter Notebook

It's far better and necessary to do practices than just reading books and code. After class, students read the explanations based on text and pictures, and directly execute the corresponding code blocks interactively, which will help the learning process, make it more smoothly and more deeply.

Students can explore independently. When problems are found in the learning process, they can easily add new code blocks to experiment and save the experiments directly on the same page. Jupyter Notebook greatly lower the threshold of exploratory learning and greatly improve the learning effect.

3.2.2 Adopting PPT

Students probably just read the teaching contents without any effective practice: Students often only read the text and cannot directly execute the code interactively. The understanding is not solid and it is easy to forget.

3.3 Practice after class

3.3.1 Adopting Jupyter Notebook

Jupyter Notebook make programming easier. Students can ask questions and implement their ideas at any time according to their own thinking logic, they can start interactive experiments with small code blocks and basic functions, and get the results and visualize it. They can gradually improve the design, build larger and more complex code blocks and debug. All of this can be done on the same page.

Saving all the experiments is also easy. Multiple ideas and multiple code blocks are integrated in the same page, saved can be executed at any time, explained at any time, verified at any time, and observed at any time. Jupyter Notebook greatly helps the students, providing them with a better learning and practical experience.

3.3.2 Adopting PPT

It can only save ideas in text or image format, and cannot debug and run programs using the same PPT document conveniently.

During this teaching reform, the results show that the introduction of Jupyter Notebook in teaching instead of traditional PowerPoint courseware has lowered the threshold for students to practice, programming and experiments are more convenient. The students have higher enthusiasm. For knowledge points they do not understand well, the students can explore and design experiments on their own. We evaluated the classroom teaching, after-school review and practice, the effects of teaching and learning have been significantly improved, students have a deeper understanding of grammar knowledge than before when using PowerPoint, and their ability to solve practical problems has also been significantly improved.

References

1. Zhao G. Cross-integration case teaching of Python programming for new engineering subjects. Computer Education 2017; (8): 23-27.
2. Qin K, Liu G. Apreliminary study of university teaching reform oriented to Python application. Computer Education 2017; (9): 21-25.
3. Song T, Huang T, Li X. Python language: an ideal choice for teaching reform of programming courses. China University Teaching 2016; (2): 15-20.
4. Zhang L, Jin Y, Zhang J. MOOC-based "Python plays with data" flipped classroom practice and research. Industry and Information Technology Education 2017; (3): 70-75.