Scholars Journal of Arts, Humanities and Social Sciences
OPEN ACCESS
Abbreviated Key Title: Sch J Arts Humanit Soc Sci
ISSN 2347-9493 (Print) | ISSN 2347-5374 (Online)
Journal homepage: https://saspublishers.com

Research on Python Language Teaching Based on Case

Taisong Xiong[1*], Yuanyuan Huang[1]

[1]Chengdu University of Information Technology, Chengdu, 610225, P.R. China

DOI: 10.36347/sjahss.2021.v09i10.005 | **Received:** 03.09.2021 | **Accepted:** 09.10.2021 | **Published:** 21.10.2021

***Corresponding author:** Taisong Xiong

Abstract **Review Article**

Python language is a more and more widely used programming language. It becomes an inevitable choice to chose Python language as an undergraduate programming language teaching. Aiming at the current Python language teaching, focusing on basic grammar explanations, lack of case-based and comprehensive application of knowledge points. We propose a case-based python teaching plan, and these cases include comprehensively Python grammar knowledge to effectively enhance students' learning interest and learning effect.

Keywords: Python language, programming language, grammar, teaching plan

1. INTRODUCTION

Python language is becoming more and more popular among researchers and programmers due to its simplicity, openness, and legibility. Python language has been widely used in many fields such as machine learning and data analysis. At the same time, numerous third-party libraries provide support for Python applications. Therefore, Python is taught in many universities at china and other countries. Teaching a programming language is first to enable students to master the basic grammar and application characteristics of the language, the similarities and differences with other programming languages, and then to enable students to comprehensively apply the programming language to solve practical problems. The model of explaining the basic grammar knowledge points and simply listing the grammar knowledge points makes the lecture very boring. It also results in students' low acceptance and bad teaching effect. If there is no comprehensive training topic, the knowledge acquired by the student will be very messy after teaching whole grammar knowledge points. After learning, they don’t know how to use the knowledge points to resolve the realize problems. The teaching goals don't reach. Therefore, for programming languages, case-based teaching is used to allow students to learn and master basic grammatical knowledge in vivid case-based teaching. At the same time, they can clearly realize that the knowledge they have learned can be applied to the corresponding places, which can improve students' enthusiasm and understanding.

2. SPECIFIC CASES

Three cases of knowledge points are given in this chapter, through which students can master the corresponding knowledge points.

2.1 The programming case of input() function

When talking about the usage of input() function, we use the comprehensive case shown below, so that students can not only learn and master the usage of input() function, but also review and deepen their previous knowledge. The implementation of this case requires the input of three data items: name, gender and age of two people, in which the values of name and gender are string types and the values of age are integers. These three data items form a dictionary, which is placed in a list variable.

```
peoples=[] # Initialization list Line 1
for num in range(2): # Line 2
name=input('please input the name') # Line 3
sex=input('please input the sex') # Line 4
age=int(input('please input the age')) # Line 5
people={'name':name, 'sex':sex, 'age':age } # Line 6
peoples. append (people) # Line 7
print(peoples) # Line 8
```

After inputting the information of two people, the for loop ends, the value of the list - peoples is printed, and the following results can be obtained.

```
[{'name': 'zhang san', 'sex': 'man', 'age': 20}, {'name': 'wang wu', 'sex': 'female', 'age': 18}]
```

The function of input() function is to wait for users' input. Press enter to indicate the end of users' inputs, and then the return value of function is the input data of user and its type is string. Therefore, when the age value is obtained as an integer in the program, the int() function is used to explicitly convert the string to an integer. Although this program has only 8 lines codes, it contains 7 different basic Python usages. The function of the first line is to initialize the list variables. The second line represents the for loop, and the third and fourth lines represent the value of the name and gender entered by the user from the input() function and assign them to the corresponding two variables respectively. The int() function is adopted to explicitly convert the value obtained from the input function to the integer type in the fifth line. The three input values are composed into a dictionary element to represent a person in the sixth line. A dictionary element is inserted into the last position of the peoples list in the seventh line. The print function is used to print the values of the list peoples in the line 8.

2.2 The programming case of class object

Python is an object-oriented programming language. Class and object programming are an important part of the object-oriented programming language, so it is also an extremely important part of Python programming learning. Since students have already mastered most of the basic grammatical knowledge of Python when explaining Python programming, the cases used in the teaching of classes programming will review the knowledge learned before. This case mainly realizes the basic information of a class and the studies, hobbies and other information of the students in the class. This case has designed two classes Class_records and Student_record. The class Class_records is used to describe the basic information of one class. It mainly includes five fields such as class_name, faculty, instructor, students, filename, among which students is a list type, and its elements are object instances of class Student_record. Class_records defines five methods: __init__(), init_classinfo(), add_students(), print_class_information(), print_students_information(). The initialization function __init__() is defined as follows:

```
def __init__(self,filename):
self.class_name=''
self.faculty=''
self.instructor=''
self.students=[]
self.filename=filename
```

Among them, filename is the name of the file storing class information which uses the csv format , and then read the csv file in the init_classinfo() method and obtain the field values to realize the assignment of the fields in Class_records. Through the realization of this method, the csv file operation is reviewed. The member method add_students (self, filename) realizes the addition of student records by accessing the student information stored in the file.

Class Student_record is used to represent the information of students. It mainly includes six fields such as name, age, sex, stuno, hobbies, records, among which hobbies is a list type, which means that students can have multiple hobbies; records is a dictionary type, which used for recording subjects and their corresponding grades. When reading student hobbies information and student subject record information, two functions read_hobbies(filename) and read_records(filename) are defined to read records from the file, and the functions return variable values of list type and dictionary type. Then the returned values are assigned to the member variables hobbies and records.

Three different Python formatting output methods are used to implement In the function of formatting and outputting student information. Therefore, the students can have a more extensive and in-depth grasp of the use of print() function. It can also expand students' knowledge and horizons.

1. Use print(f"{'name='}{self.name} {'age='}{self.age} {'stuno='}{self.stuno}") in the method print_student(self) to output information.
2. Use print(" {}".format(self.hobbies[i]),end="") in print_hobbies(self) to output multiple hobbies without wrapping.
3. Use print(' %s= %d'%(k,v), end="") in the print_records(self) method to output subject and grade information.

2.3 Visual programming case

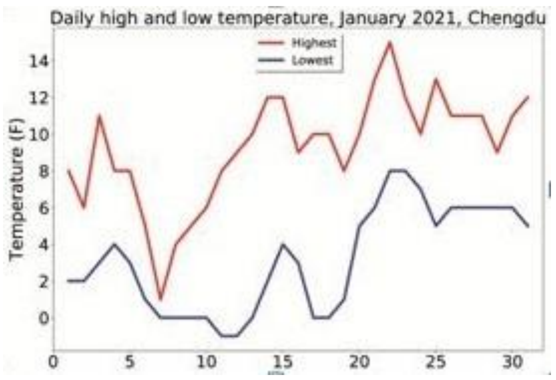

Figure 1: The curves of daily high and low temperature

Displaying data and its internal relationships graphically is an important function of data visualization. Python not only has important applications in data analysis, it can also play an important role in visualization. Python visual programming is an important part of it. At the same time, the visual programming application involves multiple programming techniques of python. A good visual programming case can make students master the skills proficiently which is an important goal of the teaching program. We carefully designed and implemented a case of drawing the daily minimum and maximum temperature curves in Chengdu in January 2021. The result is shown in Figure 1.

The temperature data in this case is stored in a csv file. Define the function read_csv(filename) to read the highest and lowest temperature values and return numpy array variables.

```
def read_csv(filename):
highs=[]
lows=[]
with open(filename) as fd:
jsreader=csv.reader(fd)
for row in jsreader:
highs.append(int(row[0]))
lows.append(int(row[1]))
return np.array(highs), np.array(lows)
```

A class draw_plt is defined for drawing images, which contains two member variables plt and xdim. Define the member method plt config (self) to set the image drawing format, draw_pic (self, highs, lows) to draw the curve and keep the image. The definition and implementation of the class is given as follows:

```
class draw_plt:
def __init__(self,plt,xrange):
self.plt=plt
self.xdim=np.arange(0, xrange) + 1

def plt_config(self):
self.plt.title ("Daily high and low temperature, January 2021, Chengdu", fontsize=30)
self.plt.xlabel('', fontsize=16)
self.plt.ylabel('Temperature (F)', fontsize=30)
self.plt.tick_params(axis='both', which='major', labelsize=28)
self.plt.xlim(0, 32)

def draw_pic(self,highs,lows):
plt.plot(self.xdim, highs, '-r', linewidth=5)
plt.plot(self.xdim, lows, '-b', linewidth=5)
plt.legend(('Highest', 'Lowest'), loc='upper center', shadow=True, fontsize=20)
plt.savefig('figure.eps')
plt.show()
```

A function visualization() is defined to implement testing the function read_csv(filename) and the instantiation of the class draw_plt and the class method. The definition of the function visualization() is shown as follows.

```
def visualization():
filename = 'cd.json'
highs,lows=read_csv(filename)
figure(figsize=(15, 10))
pltclass=draw_plt(plt,len(highs))
pltclass.plt_config()
pltclass.draw_pic(highs,lows)
```

3. SUMMARY

We design the flexible application of Python's multiple grammars and knowledge points through 3 comprehensive cases. Students can master Python knowledge points flexibly and comprehensively through case-based teaching. In the future teaching, these cases should be gradually improved, and the Python grammar and actual cases should be effectively combined, so that students can more efficiently, comprehensively and accurately master the knowledge points of Python. Furthermore, they can use these knowledge point more flexibly.

REFERENCE

- Eric, M. (2016). Python Crash Course A Hands-on, Project-Based Introduction to Programming. William Pollock. 2016.
- Guanghui, Z. (2017). Python programming cross integration case teaching for new engineering. *Computer education*, 8, 23-27.
- Tian, H., Tianyu, H., & Xin, L. (2016). Python language: an ideal choice for the teaching reform of programming course. *China University Teaching*, 2, 42-47.
- Honghong, G., Nuo Y., &Hong Z. (2021). Construction of basic course of Python Programming for new engineering. *Modern computer*, 7, 118-121.
- Xu, Z. (2020). Research on Python Programming teaching reform based on OBE concept. *Education modernization*, 7(33), 44-47.