



## IS PYTHON AN APPROPRIATE PROGRAMMING LANGUAGE FOR TEACHING PROGRAMMING IN SECONDARY SCHOOLS?

Eva Mészárosová

Department of Informatics Education, Faculty of Mathematics, Physics and Informatics,  
Comenius University, Mlynská dolina 1, Bratislava, Slovakia  
eva.meszarosova@fmph.uniba.sk

### Abstract

A variety of programming languages are used to teach fundamentals of programming in secondary schools in Slovakia. Nowadays, we observe a new trend, the Python language gaining ground. In our paper we evaluate the interviews, in which we asked teachers with years of pedagogical experience, what the reasons for selecting a particular programming language where. By analysing the responses we learn about their experience with teaching programming and create a list of the important elements in the selection of the most suitable programming language for secondary school students. We will seek an answer for the question whether the Python programming language is appropriate for all secondary school students.

### Keywords

teaching programming, programming language, Python, secondary school

### Introduction

Nowadays a huge amount of programming languages is available, so an often raised question is, which is the best programming language is for teaching programming fundamentals for all students in secondary schools. The current trend shows that more and more universities choose the Python language for teaching basic programming courses. These incorporate world-famous institutions such as Michigan State University (CSE, 2014), Massachusetts Institute of Technology (Guttag, 2013), New York University, Carnegie Mellon University and the University of Cambridge. At the Faculty of Mathematics, Physics and Informatics of Comenius University in Bratislava there is currently an ongoing introductory course of programming in Python for the first year students of Applied Informatics (Blaho, 2014). Should secondary schools also react on this trend of teaching in Python? Is python an appropriate programming language for all of the secondary school students?

Secondary schools in England, the Czech Republic and Hungary also run introductory courses for programming and some of them have already switched to teaching programming in Python. Python is currently used for teaching programming in some secondary schools in Slovakia as well. However, this teaching takes place either completely without textbooks and methodologies, or by methodologies developed by the teachers.

## The Python programming language

The Python programming language was created by Guido Van Rossum in 1989. It is an interpreter programming language developed as an open source project. Python supports object-oriented programming, procedural and also functional programming. It is a cross-platform language, which means that programs written in Python run under many operating systems including Microsoft Windows, Linux and Unix systems like Mac OS X, with almost complete support of the standard and third-party libraries, by simply copying the source code of the program. (Summerfield, 2010)

Attractive features of the language Python are the easy readability of the code, its clean syntax and that it is more intuitive than the other languages. The following features make Python suitable for users:

- punctuation characters are not used to surround code blocks or separate commands, but some of them are used to indicate the following block of code, or to index data structures
- whitespace has a special meaning – it indicates the following block of code by which the user is encouraged to write a code that is easy to read
- it supports many ways to structure the program, it is up to the user to pick the best one – this feature allows the teachers to choose a sequence of topics most suitable for their goals

One of the benefits of the Python language is also the number of existing libraries available for free, but their use is not mandatory. Beginners have no need to know the libraries, their functions and features to learn the basics of programming. Many basic programming courses integrated the libraries into the curriculum gradually. The most important libraries for teachers could be the `Tkinter` graphical library, the `Random` library to generate random values, and the turtle graphics library called `Turtle`, which allows the students to create attractive and interesting programs.

Another benefit for the students is the fact that Python is not intended exclusively for educational purposes, but they may later use it during their professional career, as it is used in many areas of information technologies. It is used for example in web development, network administration, computer games programming, data processing and a number of programs has an integrated support for Python scripts (e.g. Blender, Photoshop). (Briggs, 2012)

Nevertheless, working with variables in Python is atypical. When assigning a value to a variable, the reference to the value is assigned. Further assignments change the associated

reference. Variable types are not declared and during runtime the type of the assigned value can change.

## Comparison of Python and Pascal (Delphi)

According to Stoffová and Czakóová (2012), programming was taught at secondary schools (grammar schools) particularly in the Pascal programming language, in the Delphi and Lazarus development environments. At secondary technical schools they focused on languages designed for creating applications and websites like C, C ++ or C#. The survey described below also shows that Pascal (Delphi and Lazarus environments) is widespread in schools, and the majority of survey respondents currently teach programming in this language.

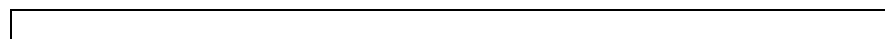
To compare the syntax of Python and Pascal I have chose an example from a science textbook for secondary schools: Programming in Delphi and Lazarus (Blaho, 2012), which teachers in the survey identified as the most used material for teaching programming. I used two short parts of a program for visual representation of Python's syntax and how it differs from the syntax of Pascal.

In the Pascal programming language the keywords `begin` and `end` are used to indicate a block of code. In Python a block of code is indicated by whitespaces (usually about 4 spaces). Statement terminators are not used in Python either, as statements are newline terminated, but a semicolon may be used alternatively. A colon is used to indicate a block of statements following branching or looping statements as well as function definitions.

```
import tkinter
g = tkinter.Canvas()
g.pack()

r = 10
y = 30
for i in range(1,11):
    x = 30
    for j in range(1,i):
        g.create_oval(x-r, y-r, x+r, y+r, fill='red')
        x += 2 * r
    y += 2 * r
```

**Tab. 1:** Example of the program written in Python

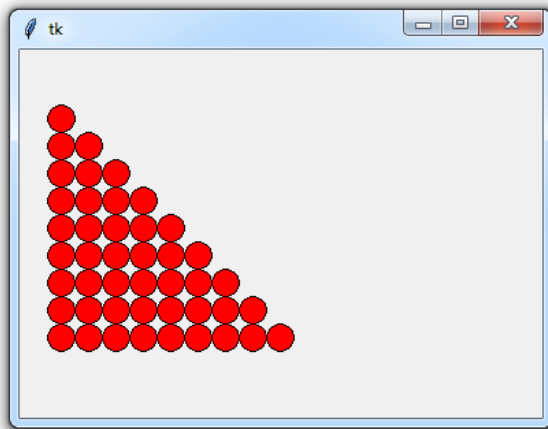


```

procedure TForm1.ButtonClick(Sender:TObject);
var
  x, y, r, i, j : Integer;
begin
  Image1.Canvas.Brush.Color := clRed;
  r := 10;
  y := 30;
  for i := 1 to 10 do begin
    x := 30;
    for j := 1 to i do begin
      Image1.Canvas.Ellipse(x-r, y-r, x+r, y+r);
      x := x + 2 * r;
    end;
    y := y + 2 * r;
  end;
end;

```

**Tab. 2:** Example of the program written in Pascal (Delphi)



**Fig. 1:** Example for nested loops (drawn by the program above)

## Methods of research – pilot study

The topic of my dissertation is the Python programming language as part of computer science education. This article is based on my pilot study in which I am getting familiar with this topic on a theoretical level. I acquired the data described in this paper by collecting and studying literature on the topic.

### Survey

To determine which language is appropriate for teaching programming fundamentals in secondary schools, I needed the opinions of experienced teachers. I implemented a survey in

which I asked computer science teachers in secondary schools for their opinions on programming languages. The survey questions were answered by 33 teachers from Slovakia, who teaches programming at secondary (grammar) schools or secondary technical schools in big cities. The country was approximately equally represented with minor variations. These teachers have 13 years of computer science teaching experience on the average. Here is a summary of their responses:

*What features should an ideal programming language for teaching programming in secondary schools have?*

In their answer teachers mostly emphasized the importance of simple, well understood and intuitive language syntax. According to them, the language should be clearly and logically structured and illustrative, so that the teaching could focus more on the development of algorithmic thinking than on teaching the rules of the programming language.

For the pupils, positive motivation is important and therefore a relatively quick feedback associated with graphical output from the written program as well. It is also necessary that the language promotes right programming habits, supports object-oriented programming and has strong motivational properties. Many teachers mentioned the need for easy debugging and syntax highlighting in programming environments. They also emphasized the importance of a methodology and traditions in computer science education.

*What features of programming languages limit or complicate the process of teaching programming?*

Most teachers identified as the most limiting feature of a language its difficult and strict syntax with long statements. As negative features they identified also abstractness, clumsiness and poor clarity of a language, which has too complicated and illogical structures. According to one teacher: „... when pupils learn to program, they're learning a new foreign language. They must learn not only the right words, but also the right syntax – where to put dots, semicolons, which parentheses to use ... and of course the order of these elements is also very important. If I wanted to simplify the answer to this question: the less such elements a programming language includes, the more attention could be paid to the 'logic' of programming“.

Teachers' responses indicate that a poor quality debugger is also limiting and they identified as a negative feature also when the language allows to run code with semantic errors. The need to add a type to variable declarations, pointers, challenging environments and keywords in English were also identified as complicating features, since some students are not learning English. The lack of textbooks also limits teachers and schools mostly prefer freeware or open source software.

*Which textbooks, books and materials do you use for teaching programming?*

Most of the teachers mentioned the textbook of programming in Delphi - *Informatika pre stredné školy: Programovanie v Delphi a Lazaruse* (Blaho, 2012). Beside this textbook they use their own materials as well as other ones from the internet, their school, national education

projects and the Faculty of Mathematics, Physics and Informatics of Comenius University in Bratislava. 22 out of 33 teachers are teaching programming in the Pascal language (Delphi and Lazarus environments), the other teachers are teaching in Turbo Pascal (8 teachers), C or C++ (7), Java (2) or Python (1). The sum of the taught languages is more than 33 because many of the teachers, especially the ones at secondary technical schools, are teaching programming in more than one language.

The survey also shows that 12 teachers out of 33 are planning to switch to another programming language in 5 years and 6 of them stated that their language of choice is going to be Python. I also asked teachers for their opinion on the Python programming language. Approximately half of the asked teachers knows this language, but most of them categorized their knowledge as weak – they saw some program code or heard feedback about the language, though they never wrote any program in it. Teachers in the survey marked Python as a simple to understand clear programming language with wide range of usage. According to them, Python is practical, quick, easy to read, applicable and “*super to teach and learn*”. There were also answers like it is a “*one among many*” language and “*it’s similar in structure to Pascal*”.

### **Interview with selected teachers**

I decided to take interviews with two teachers, who have many years of experience in teaching computer science and programming and with whom we talked more about their views on the ideal programming language and the limiting features of programming languages.

The interviews were conducted in the context of a qualitative research using the method data collection of a structured interview. The research sample consisted of two computer science teachers. I started the research by selecting respondents, whom I asked questions from the survey during the interview. I recorded the interviews and later transcribed them and analysed the obtained data.

The respondents were asked two questions:

- 1) What features should an ideal programming language for teaching programming in secondary schools have?
- 2) What features of programming languages limit or complicate the process of teaching programming?

The first respondent was a computer science teacher with 17 years of experience, who teaches at a school for highly gifted pupils. He is teaching programming in Python already the 4th year already and in the interview he shared his experiences and opinion on why he chose this programming language.

He described his views on the above questions, by dividing the programming languages into three groups, while for the languages he described the features that are ideal or limiting for a first programming language for the pupils.

For the Pascal programming language he emphasized mainly the negative features, like that there are two types of loops (repeat and while) and that Pascal is actually not often used in practice.

For the C language, in which the pupils learn the programming fundamentals, he mentioned as positive aspects its speed, and that the language is an interface between high and low level languages. As a negative aspect he considered the freedom of formatting the code. He knows from his experience that pupils do not format the code written in C well, which is consequently difficult to read.

We had a long conversation about the Python language. The respondent emphasized the positive aspects of Python:

- + Indentation
- + Minimal overhead
- + Intuitive
- + Does not burden pupils with formalities (such as Form.create, etc.)
- + Scripting language, OOP, functional – universal language supporting different ways of programming
- + Interface – useful in practice

The only disadvantage, as the respondent believes, is that the types of variables are not declared, and the type of the value can change during runtime. According to the respondent, this can be quickly explained to the pupils and “*it does not cause such a mess as it could*”.

The second respondent was a computer science teacher with 15 years of experience and received a dissertation in Technology Enhanced Learning. He teaches programming in Pascal using Delphi at a secondary (grammar) school. The respondent answered to the above questions extensively, while he justified his point of view with examples from his pedagogical experience. According to him, the ideal programming language should have the following features:

Positive:

- + Routing – for the student to have an idea of the program he writes
- + Error notification
- + Libraries – a set of libraries that are not mandatory to use
- + Develops critical thinking – finding an optimal solution for the problem
- + Graphics – motivation for the pupils
- + Examples having a natural context – interactive games, not pure math
- + A language close to practice
- + Gradual uncovering of the curriculum (language options – the degree of abstraction)

Negatives:

- Examples with mathematical context
- Definition of the variables in an another part of the code
- Lot of code needed to make the program do something meaningful
- License for development environment
- Untyped variables
- Complicated environment

## Choosing an appropriate programming language

The selection of the first programming language is very important in terms of obtaining the correct programming habits by pupils as well as the acquirement of appropriate algorithms. The first programming language will affect the success of students in using and creating algorithms and proposing solutions to problems, see (Atteq et al., 2014). The national education curriculum includes the thematic field *Procedures, problem solving, algorithmic thinking*. The standard content of the course in this field includes in addition to algorithms, also problem analysis and program debugging, also programming language features like syntax, program execution and logical errors. According to the standards of performance, pupils should be able to solve problems using algorithms and be able to write them using the commands of the programming language. They should also understand the complete programs, analyze the problem, propose an algorithm to solve a problem and verify the correctness of the algorithm, see (ŠPÚ, 2008).

Many experts compare and evaluate programming languages in their works. Mareš in his paper (Mareš 2015) emphasized the following requirements for the first programming language:

- 1) **directness or linearity** – easy to code without the need for terms and structures that the user does not know yet
- 2) **transparency** – easy readability of the code, even for someone who can program but does not know the language.
- 3) **resistance** – detect errors when executing the program, easy debugging
- 4) **abstraction** – to choose the level of abstraction by the pupils' skills
- 5) **development environment** – quality development environment corresponding to the complexity of the educational curriculum.
- 6) **history of language and practicality** – it is good if the language in which programming is taught is used in practice in the world and is close to the other languages to facilitate subsequent learning of other programming languages
- 7) **continuity** – it is worth to continue the programming language they encountered in primary school, and also to take in account that many students will meet with other languages in their further studies
- 8) **availability** – licenses, development environment should be available for free to all pupils on all commonly used devices and operating systems



- 9) **literature** – methodologies, books and textbooks are needed for educational purposes for the language taught at the school
- 10) **teachers** – it is necessary that teachers of computer science know the programming language used to the necessary level to be able to teach the curriculum and also to find errors in pupils' programs fast enough

The above mentioned criteria for the choice of the first programming language for courses of programming fundamentals are broadly consistent with the criteria by other professionals from the world. Zelle (2015) Grandelle et al.(2006), Ateeq et al.(2014), Krpan and Bilobrk (2011) and many experts are dealing with this subject in their research. In these publications they compare different programming languages with Python, such as Java and C ++, while in their conclusions Python is considered to be the most suitable language for introductory programming courses. Furthermore, they mention their positive results from programming courses conducted in Python.

## Conclusion

*„Programming is hard, but we should strive to make it no harder than it needs to be“*

Zelle John M.

In this paper I described the various opinions of teachers and experts on the required features of the first programming language in secondary schools. Most of the reviews seem to agree that we want to focus on teaching algorithms, developing problem-solving skills, and we want students to learn particular skills, which they will later use in their study and practice. Therefore we are not focusing on teaching a programming language, but to teach programming fundamentals. A language, which is suitable for this purpose must allow students and teachers to concentrate on algorithms and acquired competencies without encumbering them with the language syntax and a complicated development environment. From the survey and interviews we found out what are the demands of teachers on the first programming language.

Although the Python programming language is not fulfilling all the requirements of teachers and specialists, but compared to other languages it satisfies the most of them. The problem with the Python programming language is mainly the lack of methodologies and textbooks for secondary schools in Slovak language. Therefore in the course of my research, I will try to answer the following question: what topics, algorithms and competencies should we teach in secondary schools and how to choose the sequence of topics and the methodology.

## References

CSE 231: *Introduction to programming 1*. Department of Computer Science, College of Engineering, Michigan State University, 2014, Available from: <http://cse.msu.edu/~cse231>

GUTTAG, John V. *Introduction to Computation and Programming Using Python* (Spring 2013 Edition). Massachusetts Institute of Technology. Cambridge: Press, 55 Hayward Street, MA 02142. ISBN 978-0-262-51963-2.

BLAHO, Andrej. Programovanie v Pythone [online]. [cit. 2015-02-03]. Available from: <http://python.input.sk/>.

SUMMERFIELD, Mark. *Python 3, Výukový kurz*. Computer Press a. s., Brno, 2010. ISBN 970-80-251-2737-7.

BRIGGS, Anthony. *Hello! Python*. Manning Publications Co., Shelter Island, NY, 2012. ISBN 978-19-351-8208-5.

STOFFOVÁ, Veronika and Krisztina CZAKÓOVÁ. Propedeutika programovania a nová školská reforma. In *Journal of Technology and Information Education, Časopis pro technickou a informační výchovu*, 3/2012, vol. 4, issue 3. ISSN 1803-537X.

BLAHO, Andrej. *Informatika pre stredné školy: Programovanie v Delphi a Lazaruse*. 2. vyd. Prešov: Polygraf print, spol. s.r.o, 2012. ISBN 978-80-10-02308-0.

ATTEQ, Muhammad, Hina HABIB, Adnan UMER and Muzammil UI REHMAN. C++ or python? Which One to Begin With: A Learners Perspective. In *2014 International Conference on Teaching and learning in Computing and Engineering, IEEE*. ISBN 978-1-4799-3592-5/14.

ŠPÚ – Štátny pedagogický ústav. *Štátny vzdelávací program, Informatika – príloha ISCED 3A*. Bratislava, 2008.

MAREŠ, Martin. *Tri otázky k výuce programování* [online]. [cit. 2015-03-03]. Available from: <http://mj.ucw.cz/papers/proglang.html>

ZELLE, John M. Python as a First Language. Department of Mathematics, Computer Science, and Physics Wartburg College [online]. [cit. 2015-02-03]. Available from: <http://mcs.wartburg.edu/zelle/python/python-first.html>.

GRANDELL, Linda, Mia PELTOMÄKI, Ralph-Johan BACK and Tapio SALAKOSKI. *Why complicate things? Introducing programming in high school using Python*. Proceedings of the 8th Australasian Conference on Computing Education, pp. 71–80, January 16–19, 2006, Hobart, Australia.

KRPAN, Divna and Ivan BILOBRK. *Introductory programming languages in higher education*. MIPRO, 2011 Proceedings of the 34<sup>th</sup> International Convention. IEEE, 2011.