

WHY PYTHON IS A GOOD PROGRAMMING LANGUAGE FOR LEARNING TO PROGRAM

Nebojša Vasiljević, nebojsa.vasiljevic@mtt.gov.rs, Republic of Serbia,
Ministry of Foreign and Internal Trade and Telecommunications

The choice of a programming language for learning to program highly depends on circumstances, so we can't give a general answer which choice is the best, but there are programming languages that stand out with their characteristics as appropriate for learning to program.

Programming is often learned for the purpose of achieving a more specific goal, which then influences the choice of a programming language. For example, if your motivation to learn programming is Android application development, you will learn to program in Java,

and if you suddenly realize the need to deal with macros in Microsoft Word or Excel, then you will make your first steps in programming with Visual Basic. In such cases, the question of choice of a programming language obviously doesn't even arise.

First steps in programming can be achieved with tools based on composition of graphical elements or on a specially designed programming language. In such cases, we actually don't choose a programming language, but a tool as a whole and some tools of this type, such as Scratch¹ can be a really good choice for the first steps

¹ <http://scratch.mit.edu/>

in programming, especially in the younger age. However, sooner or later in learning to program we have to approach conventional programming languages, and here we come back to the question from the subject we are dealing with in this paper.

If you want to focus on understanding ideas and acquiring programming skills rather than mastering the programming language itself, and at the same time you want to choose a programming language that is well represented in practice, you can hardly make a better choice than Python ².

If you want to familiarize yourself with the Python programming language before you continue reading this paper or at some point later, we recommend you to take a look at the Python interactive tutorials on Codecademy ³.

Below we will highlight and explain certain features of the Python programming language that make it a good choice for learning to program.

1. It ranks in the 10 most used programming languages

За учење програмирања треба изабрати jeFor learning to program, you should choose a programming language that has taken root in practice and is widely represented. Tool and library support, presence on various platforms, experience of colleagues, available literature and the reasonable expectation that the language will not be forgotten in the next five or ten years are important considerations when you are choosing a programming language.

Of course, we don't claim that the threshold for the selection of a programming language should be sharply the top 10, but the fact is that Python ranks in the top 10 with almost all ranking methodologies.⁴

² <http://www.python.org/>

³ <http://www.codecademy.com/tracks/python>

⁴ An article *Measuring programming language*

Here we have deliberately avoided emphasizing the argument that you should learn a programming language which you will probably use later in professional work. The focus of learning programming shouldn't be on mastering a programming language, but on the concepts that can be expressed in various programming languages, including the one you have chosen. Even one who would never be a professional programmer benefits from understanding the basic concepts of programming.

2. Scripting language with dynamic typing and automatic memory management

Some people consider the fact that Python is a scripting language with dynamic typing an advantage, while others see it as a disadvantage, but for an introduction to programming, advantage is prevalent.

First of all, the simplest examples are really simple, without «extra code» that is explained such as «It must stand as it is, we will learn later what it is for.» A first example of a Python program could be:

```
print('Hello World!')
```

contrary to an equivalent program in, for example, C++ programming language:

```
#include <iostream.h>
int main()
{
    cout << «Hello World!»;
}
```

You can run Python code in the interactive

popularity on Wikipedia (http://en.wikipedia.org/wiki/Measuring_programming_language_popularity) states five sources of measuring programming language popularity and at the time this paper was written (May 2013) all five sources list Python among the top 10 of the most popular programming languages.

console (also known as Python Shell), execute simple examples instantly and see the results. This could be extremely practical for the first steps, or even later when you want to try or check something.

Python programs are stored in text files with the extension ".py". There is no compilation, no additional files produced during compilation, no necessity to define a project for each simple example, and the like.

Python, as an interpreted language, doesn't reach the efficiency of compiled languages, such as C++, or languages like Java or C# that use just-in-time compilation, but Python is efficient enough for many applications, including learning to program⁵.

The fact that variables need not be declared and don't have a static type makes programming easier for beginners and relieves the need to understand often complex syntax of type specification. Let's take the Python program example:

```
def sum(x, y):
    return x + y

a = sum(42, 47)
b = sum('Maja', 'Marina')
print(a, b)
```

It is relatively easy to explain and understand why the program outputs:

```
89 MajaMarina
```

⁵ Some alternative Python implementations such as PyPy (pypy.org) or Cython (www.cython.org) support certain methods of compilation. PyPy supports just-in-time compilation and can be used as a replacement for the standard implementation (also known as CPython) with a solid level of compatibility, whereas Cython translates Python code into C programming language and is commonly used to translate a module and use it as a native CPython extension.

Note that we didn't have to say anything about polymorphism, generic types, templates, etc. Also note that each value carries its own type, so that the operation "+" was appropriately applied to integers and strings. Python doesn't distinguish between primitive types and objects, and there are not many variations of similar types. We don't need to think whether to use a float or double (for the C programming language you can dedicate a whole lesson to this subject) or whether to use an int, long, Integer, Long or BigInteger (in Java).

Python has automatic memory management, without explicit memory allocation and deallocation and without the need to use pointers explicitly.

3. Clear and easy to understand syntax

The syntax of the Python programming language is a clear and easy to understand, and we will illustrate it on the program example that determines the number of occurrences of a given character in a string:

```
def prebroj(character, string):
    counter = 0
    for k in string:
        if character == k:
            counter = counter + 1
    return counter

print(prebroj('a', 'aparət'))
```

Previous program will print the number 3, because the character 'a' appears three times in the string 'aparət'.

Python uses the characteristic grouping by indentation, which is simple to read, there are no additional brackets or keywords, but it takes some getting used to when you write code.

4. Strings, lists, tuples, dictionaries, and sets are built-in types with an easy to use syntax

Strings and basic collection types, such as lists of variable length, tuples, dictionaries (associative arrays) and sets are built-in types well integrated in the type system and the language syntax, with all needed operators and predefined functions.

This enables us to use the basic collection types for problem solving from the beginning when appropriate, instead of introducing them later in advanced lessons, after learning how to implement them or because they are in special libraries that should be used in a specific way.

Nowadays, you can regularly learn how to use sorting to solve a problem before you learn sorting algorithms, or how to use dictionaries before learning hash table implementations, etc. This is the same as when people were beginning to learn higher programming languages, before they learned assembler and implementation of language processors.

For learning to program, basic collection types should be treated as basic building blocks used in problem solving and programming language should allow their use without assumption of understanding of any additional advanced concept (patterns, generic type, etc). Here is an example in which the variable marks associates a list of marks to each student:

```
marks = {}
marks['Ivan'] = [3, 5, 4]
marks['Ana'] = [5, 4, 5, 5]
```

and this can be written in shorter form:

```
marks = {'Ivan': [3, 5, 4],
         'Ana': [5, 4, 5, 5]}
```

Similar code in Java would look like this:

```
Map<String, List<Integer>> marks
=
```

```
new HashMap<String,
List<Integer>>();
marks.put("Ivan", Arrays.
asList(3, 5, 4));
marks.put("Ana", Arrays.
asList(5, 4, 5, 5));
```

In this example, *Arrays.asList(3,5,4)* provides a fixed length list, and if you want a variable length list, then you should use *new ArrayList<Integer>(Arrays.asList(3,5,4))*. Java has a good support for the basic collection types, but you need to master certain advanced concepts of the programming language before you use this support with understanding.

Native support for the basic collection types gives us a simple way to construct data structures that we need in elementary algorithms by composing basic collection types and other built-in types, without new type definitions. For example, the tree of arithmetic expression $5 * (2 + 3)$ could be constructed as follows:

```
expression = ['*', 5, ['+', 2,
3]]
```

Of course, Python supports the definition of new types, that is classes, which we will show later.

5. A Good Calculator Replacement

If you use Python in interactive mode (Python shell), it can replace your calculator. For example:

```
>>> 2+2
4
```

„>>>“ is a prompt, it stands in front of what we type, and the next line is the result.

Calculator is often used in solving problems such as: "If a man walks a distance $s = 4.5\text{m}$ in $t = 3.5\text{s}$, what is his average speed?" The formula for velocity is $v = s / t$, so when you put numbers in place and enter it in the "calculator", you will get:

```
>>> 4.5/3.5
1.2857142857142858
```

Python is "smarter" than a plain calculator, so we can write:

```
>>> t=3.5
>>> s=4.5
>>> s/t
1.2857142857142858
```

If we have a serial of five measurements of the time a man ran 100m, we can calculate five velocities in this way:

```
>>> measurements = [15.3, 11.7,
21.9, 13.2, 14.6]
>>> s = 100
>>> [s/t for t in measurements]
[6.5359477124183005,
8.547008547008547,
4.566210045662101,
7.575757575757576,
6.8493150684931505]
```

We can introduce particular elements of programming as a calculator extended features, and practice them on the kind of problems that students generally solve using a calculator, which is an effective method to bring a student up to the first line of code with the minimal conceptual burden.

6. Supports multiple assignment

If you want to swap values of variables a and b, in Python you can write:

```
a, b = b, a
```

Without multiple assignment, we would have to use three assignments and an auxiliary variable. Also a function can return multiple values (actually returns a tuple), for instance *divmod* calculates both integer quotient and remainder:

```
d, m = divmod(p, q)
```

Euclidean algorithm for the greatest common divisor⁶ could be coded in Python like this:

```
def nzd(a,b):
    while b!=0:
        a, b = b, a % b
    return a
```

Multiple assignment is a detail that makes it easier to express your idea and to focus on the essence of an algorithm.

7. Mathematically consistent

From the fifth grade of primary school in Serbia, pupils are generally familiar with the concept of division remainder which is equal to zero if the numerator is divisible by denominator and is greater than zero and less than the denominator otherwise. In many programming languages, including Python, character "%" is used to denote division remainder operation. Thus $5 \% 3 = 2$. Python provides a mathematically correct result even when the numerator is negative, but in many other programming languages it is not the case (such as C, C++, C# or Java).

For instance, in Python, expression $-1 \% 3$ results to 2, as opposed to some other programming languages where the same expression results to -1. So, you can feel free to write:

```
position = (position +
displacement) % width
```

and you don't need protection against the negative values of displacement, like you need in some other programming languages:

⁶ Euclidean algorithm is described in the section Algorithms in on-line materials of prof. Cvetana Krstev (in Serbian), http://poincare.matf.bg.ac.rs/~cvetana/Nastava/Materijal/Algoritmi_1213.pdf

```
position = (position +
displacement + width) % width
```

It may sound like a detail, but modular arithmetic (calculations with division remainders) is an area of mathematics that is extensively used in programming from the early beginning and it is not convenient when you need to explain to students why something differs from what they had learned in math and how they should overcome it.

Python also introduces a special operator for integer division `//`⁷. Modular arithmetic in Python is extended to floating-point numbers, so the problem „How many boards of the size X can you cut from a piece of wood of the size Y, and how much wood remains, where the sizes do not have to be integers", can be solved as follows:

```
print(X // Y, X % Y)
```

Inequalities such as $a < b < c$ often appear in mathematics. This kind of inequality can also be written as $a < b$ and $b < c$. Most programming languages do not support the first way of writing multiple inequalities, but only the other, expanded way. In Python you are free to write:

```
if a < b < c:
    left, middle, right = a, b, c
```

Python also supports big integers, for instance Python will correctly calculate the value

$$\frac{2013^{1000}}{2013^{999}}:$$

```
>>> print(2013**1000//2013**999)
2013
```

Internally, Python makes a difference between

⁷ From the 3.0 version of Python, operator `//` provides floating point result even for integer operands, but in the earlier versions, operators `//` and `./` provide the same result when operands are integers.

plain and big integers, but programmer doesn't need to care about that.

Mathematical consistency of Python helps us to build on what a student has already learned in math and be less focused on the programming language tricks, which lets us devote more attention to the problem we are solving and to the programming concepts.

8. Supports nested functions

The Pascal programming language supports nested functions (a function defined within another function), but despite the fact that Pascal has influenced almost all subsequent programming languages in terms of the elements of structured and procedural programming, many programming languages do not support nested functions. Nested functions can be useful in recursive algorithms, providing an elegant possibility to place a recursive function within the main function, as seen in the example of binary search⁸:

```
def search(list, targetValue):
    def searchRec(left, right):
        if right < left:
            return -1
        mid = (left + right)//2
        if targetValue>list[mid]:
            return searchRec(mid + 1, right)
        elif targetValue<list[mid]:
            return searchRec(left,mid-1)
        else:
            return mid
    return searchRec(0,len(list) - 1)
```

If the function `searchRec` was not nested into the function `search`, it would be necessary to add

⁸ Binary search algorithm is described in the section Arrays in on-line materials of prof. Cvetana Krstev (in Serbian) , http://poincare.matf.bg.ac.rs/~cvetana/Nastava/Materijal/NizoviListe_1213.pdf

parameters list and targetValue to the function searchRec, which would reduce the code clarity. Take into account that there may be more than two values we need to access from a recursive function all the time. With a help of object-oriented programming, we can define a class containing a recursive method, and store this kind of values in instance variables, but this is less elegant and forces us to introduce object-oriented programming concepts just to be able to express a recursive algorithm.

9. Python supports both procedural and object-oriented and functional programming

Python allows you to involve only concepts that you need, and not to bother with others. You can easily program purely procedural, without any class or method definition, what we could see in previous examples. In Java and C# programming languages you have to define at least one class in your program and you can't define a function other than a static method.

It does not mean that Python does not support object-oriented programming; moreover, every value is an object. For example, the operation $a+b$ always comes down to the method call `a.__add__(b)`, including for the built-in types, which means that the following two lines of code calculate the sum of the numbers 7 and 9:

```
>>> a, b = 7, 9
>>> a.__add__(b)
16
```

In Python we define a method as a function inside the class, where the first parameter is reserved for the object itself (instance of the class) on which the method is called, and instance variables are created dynamically during some methods execution, usually in the constructor:

```
import math
class point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def distance(self, t):
        return math.sqrt((self.x-t.x)**2
+ (self.y-t.y)**2)

a = point(1,4)
b = point(5,2)
print(a.distance(b))
```

Previous program outputs 4.47213595499958

Python also supports the basic elements of functional programming, such as first-order functions, lambda expressions, and closures⁹. Python is suitable to demonstrate elements of functional programming and to use them to solve a problem, although it is not entirely designed as a functional language. Here is a typical functional programming example, where a function is applied to all elements of a list, which results to a new list:

```
>>> list(map(lambda x: 2*x,
[1,2,3]))
[2, 4, 6]
```

Python has special syntax for expressions like the one in the previous example. The syntax is similar to the mathematical set-builder notation, such as $\{2x \mid x \in \{1, 2, 3\}\}$, so it is understandable even if you haven't heard anything about functional programming before

```
>>> [2*x for x in [1,2,3]]
[2, 4, 6]
```

The concepts of object-oriented programming are not necessary for someone's first steps in

9 [http://en.wikipedia.org/wiki/Closure_\(computer_science\)](http://en.wikipedia.org/wiki/Closure_(computer_science))

programming and can be left for later stages of learning programming. Python allows you to start first steps in programming with the procedural paradigm, but to be able to use some functional programming constructs that will not burden the students and will just make your examples clearer. You can include object-oriented programming when you consider it is appropriate.

10. Supports classic text I/O and basic 2-dimensional graphics

Examples and problems that we use in programming lessons often suppose simple user interaction: user needs to enter some input values and program should output some results. Plain old textual input and output is the easiest way to do that. The Python standard library properly supports classic text I/O, which can be illustrated by the example:

```
a = float(input("Unesi dužinu komada drveta: "))
b = float(input("Unesi dužinu daske: "))
print("{0:.0f} dasaka i komad od {1:.2f}".format(a//b, a%b))
```

Execution of the previous program on the console will look like this:

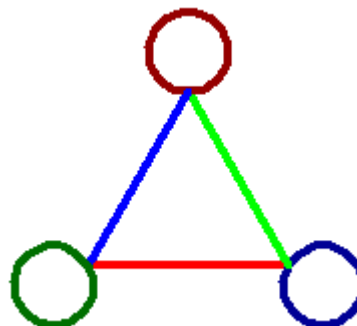
```
Unesi dužinu komada drveta: 4
Unesi dužinu daske: 0.7
5 dasaka i komad od 0.50
```

Besides text-based user interaction, in programming lessons it is also convenient to use examples with graphics. The Python standard library has support for the so-called „turtle graphics” known from the LOGO programming language. This method of graphics programming is based on the metaphor of moving object that leaves a trail. In the original LOGO implementation this object was displayed as a

turtle character, which inspired the name of the method. The value of turtle graphics lays in the fact that without the knowledge of the Cartesian coordinate system and trigonometric functions, by using simple programming constructs, you can draw interesting geometric shapes. For example:

```
import turtle
koja = turtle.Turtle()
koja.pensize(4)
for boja1,boja2 in
[ („red”, „darkblue”),
 („green”, „darkred”),
 („blue”, „darkgreen”) ]:
    koja.color(boja1)
    koja.forward(100)
    koja.right(120)
    koja.color(boja2)
    koja.circle(20)
    koja.left(240)
```

results to:



For learning programming, PyGame library¹⁰ is also often used, which has additional support for 2-dimensional (2D) drawing and 2D games.

We would also note that the program in the Python programming language does not have to determine whether it will be the console or graphical, but program can simply input some

¹⁰ <http://www.pygame.org/>

values from the console, and then draw something. When you draw with the turtle module, a separate window will be opened automatically.

11. Rich standard library, great selection of additional libraries and tools

Rich libraries widen the range of problems that can be solved with relatively simple programming. Students can easily draw charts, three-dimensional (3D) scenes, edit images or manage robots.

Standard Python installation comes with IDLE integrated development environment, which is good enough for an introduction to programming, but you can also select some more advanced development environment. Python support is present in virtually all well-known development environments that support multiple programming languages, including two generally best-known development environments: Microsoft Visual Studio

through the Python Tools for Visual Studio¹¹ plugin, and Eclipse through the PyDev¹² plugin. If you want to use a more compact development environment designed particularly for Python, you can try PyScripter¹³ or Spyder¹⁴. There are also commercial development environments specialized for Python like JetBrains PyCharm¹⁵.

If you need an enhanced interactive Python shell, integrated with a plotting library, you will find IPython¹⁶ useful. On the left side in Figure 1, the IPython Qt Console¹⁷ is shown, and in the figure we can see the example from the section A Good Calculator Replacement with an additional line of code that plots the chart.

Regarding the graphical user interface (GUI) programming, standard Python installation comes with the Tkinter library, but you can also find several multiplatform GUI libraries supported, such as wxWidgets¹⁸ and Qt¹⁹.

If you need an easy to use 3D graphics library, there is VPython²⁰.

Python is integrated into many applications, such as the excellent 3D modelling software Blender²¹, which is free and open source. There are several web programming frameworks based on Python, and the best-known is Django²².

To be able to use an additional library you

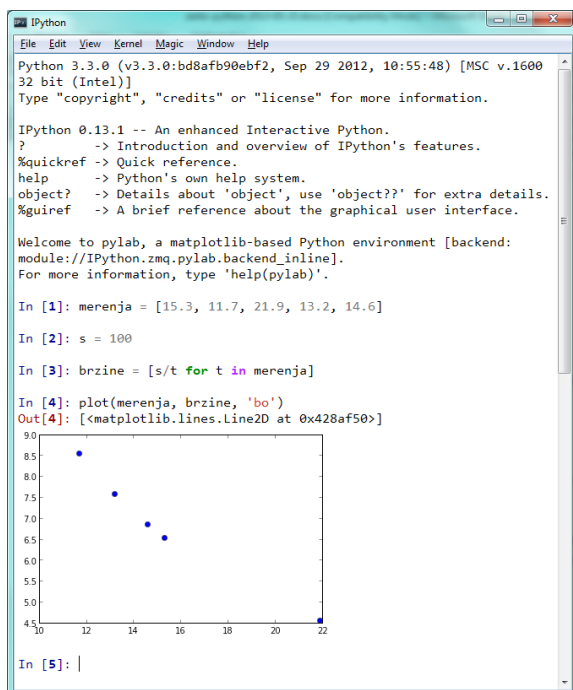


Figure 1 . IPython Qt Console

- 11 <http://pytools.codeplex.com/>
- 12 <http://pydev.org/>
- 13 <https://code.google.com/p/pyscripter/>
- 14 <https://code.google.com/p/spyderlib/>
- 15 <http://www.jetbrains.com/pycharm/>
- 16 <http://ipython.org/>
- 17 <http://ipython.org/ipython-doc/dev/interactive/qtconsole.html>
- 18 <http://www.wxwidgets.org/>, Python support: <http://wxpython.org/>
- 19 <http://qt-project.org/>, Python support: <http://qt-project.org/wiki/PySide> or <http://www.riverbankcomputing.com/software/pyqt/>
- 20 <http://www.vpython.org/>
- 21 <http://www.blender.org/>
- 22 <https://www.djangoproject.com/>

usually have to install the library support over your existing Python installation. Even though installation procedure is usually simple, you may not like to install many libraries one by one, and then you can select one of the independent Python distributions such as WinPython²³ or Python(x, y)²⁴. If you are a Linux user, check out which Python libraries your Linux distribution already encompasses.

12. Widely used for an introduction to programming

Python is a popular choice as the first programming language, both for the very young ages and for the introductory courses at universities. As an example we mention the course Introduction to Computer Science and Programming²⁵ of the Massachusetts Institute of Technology. Video recordings of all lectures and all supporting materials for the course are freely available. List of universities that use Python can be found on the web page The Python Wiki - Schools using Python²⁶.

Python is selected by the authors of many introductory books aimed at children who want to learn to program, such as Python for Kids²⁷ or the “Hello World! Computer Programming for Kids and Other Beginners”²⁸, as well as free books in electronic format such as Invent Your Own Computer Games with Python²⁹.

If you prefer to learn to program through the game programming, you can use the book Arcade Games with Python and Pygame³⁰ with

23 <https://code.google.com/p/winpython/>

24 <https://code.google.com/p/pythonxy/>

25 <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00-introduction-to-computer-science-and-programming-fall-2008/>

26 <http://wiki.python.org/moin/SchoolsUsingPython>

27 <http://jasonrbriggs.com/python-for-kids/>

28 <http://www.manning.com/sande/>

29 <http://inventwithpython.com/>

30 <http://programarcadegames.com/>

freely accessible HTML version.

Finally, we should mention the book Think Python: How to Think Like a Computer Scientist³¹, available under Creative Commons Attribution-Non Commercial licence which means that you can modify and customize it, as long as you credit the original author and don't use the derived work for commercial purposes.

13. Integrates well into interactive online content

Besides the fact that standard Python implementation supports many operating systems, there is an implementation that runs inside the web browser. On the website CodeSculptor³² you can execute Python code in your web browser. Rice University³³ uses CodeSculptor in its course An Introduction to Interactive Programming in Python, which is available online on the Coursera³⁴ platform.

Thanks to this and other technologies that allow the integration of Python with online systems, Python has become one of the best-supported programming languages in interactive online educational resources.

If you have not already done so, take a look at the Python lessons of Codecademy³⁵. Those online tutorials allow you to type code examples and answers to questions, and to execute the code directly in your web browser (Figure 2). Just go to the site, you don't have to install anything nor to copy or retype code examples, you only need to follow the instructions, supplement examples that appear directly in the editor and provide answers, and the system will correct you if something is not done properly.

We should also point out the edition How

31 <http://www.greenteapress.com/thinkpython/>

32 <http://www.codesculptor.org/>

33 <http://www.rice.edu/>

34 <https://www.coursera.org/>

35 <http://www.codecademy.com/tracks/python>

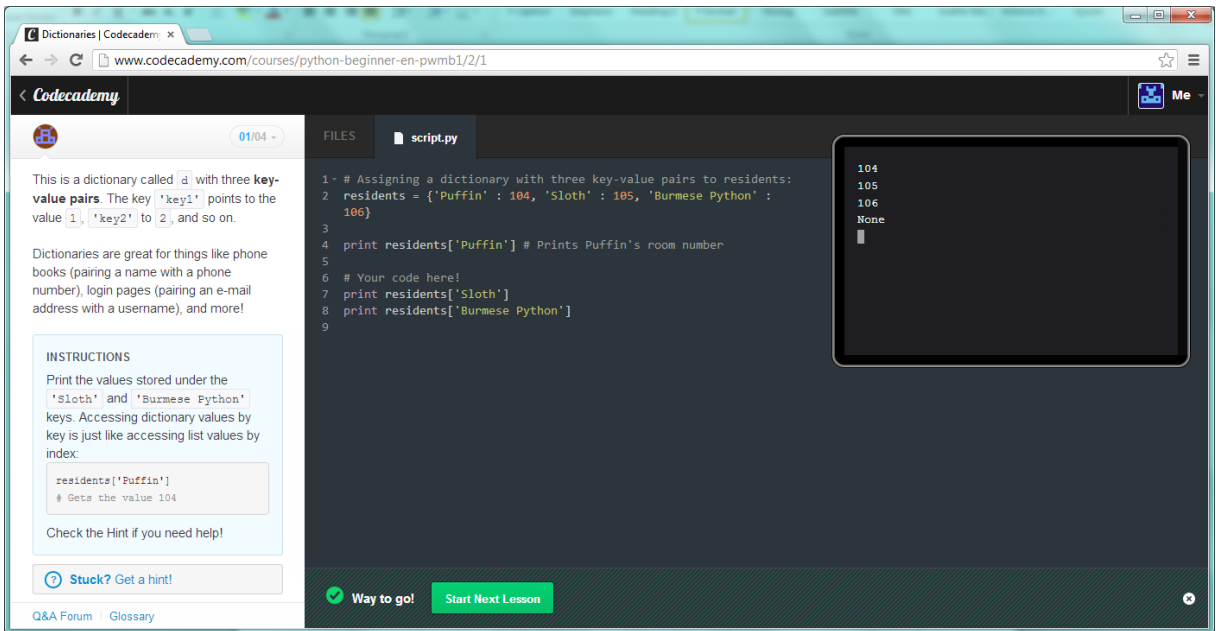


Figure 2. Screenshot from a Python lesson on *Codecademy*

to Think Like a Computer Scientist: Interactive Edition³⁶. In addition to the introductory videos for specific topics, all Python code examples can be executed directly inside the web page, can be modified and saved, including examples with the graphics. For some examples there is also step by step animation of the program execution with intermediate results displayed. This is a mix of a good textbook for introduction to computer science and cutting-edge interactive technology. Additionally, the content is fully open - both the text of the book and the whole technology platform is open source - if you want you can make your own version of a lesson or completely new lessons on the same platform. This interactive edition you can use via your personal computer or tablet, and ultimately, via a mobile phone, although using a keyboard to edit program code is more practical.

Interactive online contents can be used both for self-study and classroom lectures, where

students go through content under the teacher's supervision, but it also can be used for teacher's training and class preparation.

14. Python helps you to focus on learning to program, rather than learning a programming language

If we try to sum up all the conclusions in a single statement, the ultimate conclusion could be: Python doesn't bother you. When you learn to program with Python, you deal less with the programming language, and deal more with algorithms and concepts you aim to understand.

In recent decades, many programming languages stood out as the most common choice for an introduction to programming during various periods, and among them we can single out Pascal, Basic, and Logo. We can say that Python encompasses the key advantages of all of these programming languages, regarding learning to program:

- Python enables you to elegantly express an algorithm in program code itself, what

36 <http://interactivepython.org/>

- the Pascal language is proud of too;
- simple syntax and taken advantages of an interpreting language were the reasons why Basic once became popular for learning to program;
 - learning to program with Logo is based on examples and problems of geometric shapes drawing using «turtle» and simple syntactic constructs, which is equally feasible in Python.

In addition, Python raises the level of abstraction by introducing the basic collection types into built-in types, as well as by many other details, so that in the introduction to programming lessons you can go a step closer to the interesting problems solving.

Understanding of the basic concepts of programming is useful even for those who will never be programmers, and this kind of general education knowledge comes in the scope of what is called computational thinking³⁷, and can be defined as:

„Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.“³⁸

General education approach to learning programming leads us to the principle that with less programming we should solve more interesting problems and Python with its understandable syntax, logicity, rich standard library and third-party tools perfectly fits into this approach.

37 Jeannette M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (March 2006), 33-35. <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf>

38 Jan Cuny, Larry Snyder and Jeannette M. Wing, “Demystifying Computational Thinking for Non-Computer Scientists,” work in progress, 2010

Literature

Крстев, Цветана. 2012. Алгоритми и програмирање. http://poincare.matf.bg.ac.rs/~cvetana/Nastava/Materijal/Algoritmi_1213.pdf (приступљено 18. IX 2013)

Крстев, Цветана. 2012. Програмирање – део 3. http://poincare.matf.bg.ac.rs/~cvetana/Nastava/Materijal/NizoviListe_1213.pdf (приступљено 18. IX 2013)

Miller, B., and Ranum D. *How to Think Like a Computer Scientist: Interactive Edition*. <http://interactivepython.org/runestone/static/thinkcspy/toc.html> (приступљено 18. IX 2013)

Pérez, F., and Granger, B. E. 2007. IPython: A System for Interactive Scientific Computing, *Computing in Science and Engineering*, vol. 9, no. 3, pp. 21-29. <http://ipython.org> (приступљено 18. IX 2013)

van Rossum, G., and Drake, F. Jr. 2011. *An Introduction to Python - The Python Tutorial*. Bristol: Network Theory Ltd.

van Rossum, G., and Drake, F. Jr. 2011. *The Python Language Reference Manual*. Bristol: Network Theory Ltd.

Wikipedia. Closure (computer science). [http://en.wikipedia.org/wiki/Closure_\(computer_science\)](http://en.wikipedia.org/wiki/Closure_(computer_science)) (приступљено 18. IX 2013)

Wikipedia. Measuring programming language popularity. http://en.wikipedia.org/wiki/Measuring_programming_language_popularity (приступљено 18. IX 2013)

Wing, Jeannette M. 2006. Computational thinking. *Commun. ACM* 49 (3) : 33-35. <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf> (приступљено 18. IX 2013)

Web sites

Blender project – Free and Open 3D creation software. <http://www.blender.org/> (приступљено 18. IX 2013)

Why Python is a Good Programming Language for Learning to Program

Codecademy Python tutorials. <http://www.codecademy.com/tracks/python> (приступљено 18. IX 2013)

CodeSculptor. A browser-based Python interpreter. <http://www.codesculptor.org/> (приступљено 18. IX 2013)

Coursera online courses. <https://www.coursera.org/> (приступљено 18. IX 2013)

Scratch – Imagine, Program, Share. <http://scratch.mit.edu/> (приступљено 18. IX 2013)

Django. A high-level Python Web framework. <https://www.djangoproject.com/> (приступљено 18. IX 2013)

IPython Interactive Computing. <http://ipython.org/> (приступљено 18. IX 2013)

IPython Qt Console. <http://ipython.org/ipython-doc/dev/interactive/qtconsole.html> (приступљено 18. IX 2013)

JetBrains PyCharm. Python IDE & Django IDE for Web developers. <http://www.jetbrains.com/pycharm/> (приступљено 18. IX 2013)

MIT OpenCourseWare – Introduction to Computer Science and Programming. <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-00-introduction-to-computer-science-and-programming-fall-2008/> (приступљено 18. IX 2013)

Qt Project. A cross-platform application and UI framework. <http://qt-project.org/> (приступљено 18. IX 2013)

PyDev. A Python IDE for Eclipse. <http://pydev.org/> (приступљено 18. IX 2013)

PyScripter. An open-source Python IDE. <https://code.google.com/p/pyscripter/> (приступљено 18. IX 2013)

Python(x,y). Scientific-oriented Python Distribution based on Qt and Spyder. <https://code.google.com/p/pythonxy/> (приступљено 18. IX 2013)

Python Programming Language – Official Website.

<http://www.python.org/> (приступљено 18. IX 2013)

Python Tools for Visual Studio. <http://pytools.codeplex.com/> (приступљено 18. IX 2013)

The Python Wiki – Schools using Python. <http://wiki.python.org/moin/SchoolsUsingPython> (приступљено 18. IX 2013)

Rice University. <http://www.rice.edu/> (приступљено 18. IX 2013)

Spyder. Scientific PYthon Development EnviRonment. <https://code.google.com/p/spyderlib/> (приступљено 18. IX 2013)

VPython. 3D Programming for Ordinary Mortals. <http://www.vpython.org/> (приступљено 18. IX 2013)

WinPython. Portable Scientific Python 2/3 32/64bit Distribution for Windows. <https://code.google.com/p/winpython/> (приступљено 18. IX 2013)

wxWidgets. Cross-Platform GUI Library. <http://www.wxwidgets.org/> (приступљено 18. IX 2013)

Received: 16th May 2013.

Accepted: 24th June 2013.