

Viabilidad de Python en la enseñanza de la programación

Feasibility of Python in teaching programming

Autores: Lic. Rafael Martínez Estévez; MSc. Manuel Pereira Rosa; MSc. Raimundo González Fernández

Centro de procedencia: Universidad de Ciencias Pedagógicas "Rafael María de Mendive"

Email: rmartinez@ucp.pr.rimed.cu; manuel@ucp.pr.rimed.cu;
raimundo@ucp.pr.rimed.cu

Resumen:

Dada la diversidad de objetivos de los cursos de programación del sistema educacional cubano y la capacitación de los profesores que los imparten, no es sencillo decidir el lenguaje que debe utilizarse en cada caso. La intención de este trabajo es traer a debate en nuestro contexto una tendencia en aumento en la última década: Python como primer lenguaje de programación. El objetivo de este trabajo es recopilar algunas experiencias internacionales del uso de Python en cursos introductorios de programación, analizando además sus ventajas y desventajas.

Palabras clave: programación, python, enseñanza

Abstract:

Given the diversity of the objectives of the programming courses in the Cuban educational system and the training of teachers, it is not easy to decide the language to be used in each case. The intention of this article is to bring to debate to our context a trend that has been growing in the last decade: Python as a first programming language. The aim of this study is to compile some international experiences in the use of Python in introductory programming courses, also analyzing their advantages and disadvantages.

Keywords: programming, Python, teaching

Papel de la informática en la vida moderna

Es innegable, y no es necesario extenderse en ello, el papel cada vez más importante de la informática en la vida moderna. Cuba, a pesar de ser un país bloqueado por la mayor potencia económica de nuestro tiempo, no está ausente a este proceso de movimiento hacia la sociedad de la información que, desde hace varias décadas viene llevándose a cabo en el mundo. La creación de la Universidad de Ciencias Informáticas y la apertura de la carrera de Ingeniería Informática en varias universidades hace más de diez años, es una muestra fehaciente de nuestra voluntad por enfrentar los retos que impone, para el desarrollo tecnológico del país, pertenecer al llamado Tercer Mundo.

No solo para los especialistas sino para el ciudadano en general, cada vez reconocemos más la importancia de adquirir conocimientos básicos de informática y computación.

Enmarcando esto dentro del proceso conocido como informatización de la sociedad. Por otro lado, los fundamentos y las técnicas de programación están en el centro mismo de lo que es la computación. Es por ello que la enseñanza y aprendizaje de la programación, son procesos vitales para la necesidad de formar ciudadanos con conocimientos profundos de los principios de la computación.

No obstante el hecho que se utilice Visual Basic para introducir los primeros conceptos de programación, como parte del programa de 12 Grado de la asignatura de Informática, no puede

dejar de parecerse, al menos, discutible. De igual manera que se continúe utilizando C++, independientemente del tipo de perfil de la carrera, para introducir la programación en los primeros años en nuestras universidades. Nuestro objetivo no es desacreditar ningún lenguaje ni programa que lo utilice, las discusiones de este tipo parecen ser interminables; si no traer al debate una tendencia que se ha venido incrementando en el mundo en la última década.

En la década de los 80 e inicios de los 90 del siglo pasado el problema de qué lenguaje utilizar en la enseñanza de la programación tuvo una respuesta segura, diseñada para la pregunta: Pascal. Sin embargo, con el advenimiento de la programación orientada a objetos, Pascal dejó de ser una opción viable dada la necesidad de la industria de profesionales con conocimientos modernos en materia de programación. En esta situación, la elección del primer lenguaje de programación se convirtió en un problema de respuesta mucho menos clara. Se comienzan a utilizar entonces lenguajes comerciales como Scheme, Haskell, ML, GBeta, Smalltalk, Delphi, C++ y sobre todo Java [1]. Es en este contexto donde surge Python.

Un lenguaje de programación usado en la educación debe ser fácil de aprender, organizado en su diseño, de uso general y poderoso computacionalmente. Debe tener además una sintaxis simple, proveer un fácil manejo de la entrada y salida de datos, entre otras características. El hecho de que Python sea un lenguaje de propósito general, pero que tuvo desde sus inicios, entre sus objetivos llegar a todo tipo de audiencia [2], lo hace ser un lenguaje sencillo, que sin embargo no llega a ser un lenguaje "de juguete"; características que hacen que sea un candidato a tomar muy en serio para la educación [3].

Con el objetivo de determinar el más adecuado para la introducción a la programación, un interesante estudio [4] utiliza 17 indicadores sugeridos por creadores de lenguajes considerados útiles para la enseñanza por la comunidad académica: Seymour Papert (creador de LOGO), Niklaus Wirth (creador de Pascal), Guido van Rossum (creador de Python) y Bertrand Meyer (creador de Eiffel). En la comparación de 12 lenguajes, Eiffel y Python obtuvieron 15 puntos, Java 14, C++ 11 y Visual Basic 9.

En el sistema educacional cubano existen diversos cursos dedicados a la enseñanza de la programación. Dada la diversidad de los objetivos de dichos cursos y de las formaciones de los profesores que los imparten, no es sencillo decidir el lenguaje que debe utilizarse en cada caso. El presente trabajo propone el uso del lenguaje Python en los primeros cursos de programación.

Nuestro objetivo es compilar algunas experiencias internacionales en el uso de Python como primer lenguaje de programación, en cursos de educación media superior y universitaria, que permitan aportar argumentos a esta discusión en nuestro país. Analizando además algunas ventajas y desventajas que pudiera aportar Python, en nuestro contexto, como lenguaje introductorio a la programación.

El uso de Python en la enseñanza

Con la entrada en escena, a finales de los 80', del paradigma de la programación orientada a objetos, Pascal dejó de ser el lenguaje preferido de los educadores, quienes se volcaron a utilizar los nuevos lenguajes surgidos al calor de la nueva corriente, como C++, ADA, Java, Eiffel, Smalltalk, entre otros. Sin embargo, luego de algún tiempo, comenzó a ser evidente para algunos que estos lenguajes, aunque muy útiles en el ámbito industrial o de investigación, no necesariamente poseían las características necesarias para la docencia.

Python es un lenguaje orientado a objeto de muy alto nivel diseñado por Guido van Rossum, su primera versión data de 1991. Este lenguaje concentra la experiencia de muchos años de desarrollo de la Ciencia de la Computación; incorporando ideas de los paradigmas de programación imperativo, orientado a objetos y funcional, incluyendo excepciones, módulos y clases. Python sacrifica algún rendimiento con el objetivo de maximizar la velocidad de desarrollo. Es un lenguaje maduro y bien establecido; muy reconocido por su claridad, sintaxis legible y por la ganancia de productividad que significa para los programadores que lo prefieren sobre otros lenguajes. [5]

Al contrario de otros lenguajes con valor educativo, Python no es considerado de ninguna manera un lenguaje de juguete. Ha sido usado exitosamente en miles de aplicaciones del mundo real, incluyendo aplicaciones largas y sistemas críticos de grandes compañías como Google, IBM, Hawlett Packard, Disney, Nokia, entre otros. También ha sido muy popular en el mundo

académico y de la investigación, donde ha sido usado en el diseño de misiones espaciales, el modelado biomolecular, el control de simulaciones físicas de gran escala, el análisis y la visualización de datos climáticos, por solo poner algunos ejemplos [6].

En 1999, un estudio pionero de John M. Zelle, proponía el uso de Python como primer lenguaje de programación. Argumentando que los lenguajes script de alto nivel, como Python, Perl, Tcl, Rexx, o Visual Basic, poseen mejores condiciones como primer lenguaje. En particular Python dadas su sintaxis y demás características, parece ser ideal [1]. Varios años después el profesor Zelle publicó un libro de texto de introducción a la computación haciendo uso de Python, elogiado entre otros por Guido van Rossum, creador de Python [8].

También en este período Jeffrey Elkner comienza a aplicar Python en el programa de Ciencia de la Computación que impartía en el Yorktown High School de Arlington, Virginia [9]. De dicha experiencia surgió el que quizás pueda considerarse el primer libro de texto dedicado a la enseñanza de la programación usando Python [10].

Por otro lado el propio Guido van Rossum proponía el lenguaje para su proyecto "Computer Programming for Everybody" [2], que pretende desarrollar herramientas y documentación que brinden la posibilidad de aprender a programar, modificar y personalizar sus aplicaciones a todo tipo de persona. Pues a diferencia de lenguajes diseñados específicamente para principiantes, Python es también la elección de muchos programadores profesionales y tiene una activa comunidad en crecimiento que ha desarrollado y tiene la capacidad de desarrollar materiales de enseñanza y herramientas de aprendizaje [2]. Esto era cierto en 1999 lo es mucho más ahora.

Mientras tanto, un artículo de Frank Stajano de la Universidad de Cambridge señalaba que dado su juicioso balance entre simplicidad, concisión y versatilidad Python constituía una excelente elección para introducir los conceptos fundamentales del arte de la programación [11]. Por otro lado, una tesis doctoral de la Universidad de Michigan proponía el uso de la programación en Python como un medio de promover la alfabetización informática [12].

Otro estudio posterior analizaba que el hecho de que existan muchos menos obstáculos cognitivos que un estudiante debe vencer antes de comenzar a escribir sus primeros programas, sugerían a Python como un candidato a analizar para cursos introductorios a la programación. Reportando sus resultados, se concluía que los resultados eran alentadores, aunque deberían ser posteriormente evaluados [13].

Luego de estos primeros pasos, en algunas instituciones educativas se comenzó a utilizar Python. No tardaron en aparecer los primeros resultados:

Ceder y Yergler [14] demuestran la utilidad del desarrollo de juegos en la motivación para la introducción a la programación, haciendo uso de la Python y la librería para desarrollo de juegos PyGame [15].

En [16] se reporta que los estudiantes fueron capaces de completar el doble de ejercicios de programación que en otros cursos (que no usaban Python), adicionando a esto que una mayor cantidad de estudiantes fueron capaces de completar satisfactoriamente los proyectos de curso.

En el Centre College de Danville, KY [17] se comprobaba que los estudiantes disfrutaban usar Python pues les permitía enfocarse rápidamente en problemas interesantes, como aplicaciones web. Por otro lado permitía a los profesores introducir conceptos que serían profundizados en cursos posteriores, como la programación orientada a eventos.

Grandell et al [3]. proponían también el uso de Python en la educación media superior para introducir la programación. Entre otras cosas porque la sintaxis simple y extensible de Python reduce notablemente las complicaciones de notación que otros lenguajes. El ambiente interactivo ofrece respuesta inmediata aún con una limitada experiencia, la cual permite enfrascar rápidamente a los estudiantes en la mecánica de prueba y error, que está en el centro de la enseñanza-aprendizaje de la programación.

Una tesis de maestría [5] investigaba entre los estudiantes y profesores de algunos preuniversitarios en Grecia. Después de haber usado Python varios cursos se reconocían algunas ventajas: dado que la sintaxis de Python es más limpia reduce la cantidad de código, lo cual reduce también la posibilidad de errores; los estudiantes se mostraban entusiastas sobre su uso ya que permitía enfrascarse en proyectos interesantes casi al comienzo del curso.

En la página wiki.python.org se mantiene una lista extensiva de las instituciones educativas que usan Python[18].

Finalmente un estudio reciente [19] señala que, mientras Java permanece como el lenguaje más usado para la introducción a la programación su popularidad va en decrecimiento; el uso de C++ se ha mantenido estable, desgraciadamente según nuestro criterio; la popularidad de Python en la enseñanza va en aumento, argumentando como causas de este crecimiento algunos de los factores que ya hemos visto: simplicidad de la sintaxis, rapidez de desarrollo, interactividad, etc.

Ventajas

La limpieza de sintaxis de Python es una de las principales ventajas. Dicha limpieza permite desarrollar programas con mayor rapidez y con menos líneas de código, con la consiguiente reducción de los errores, algo de mucha mayor importancia en programadores principiantes. Considérese en los siguientes códigos del archiconocido "Hola Mundo", los conceptos implicados C++ no podrán ser abordados hasta muy adelantado el curso.

El tipado dinámico puede ser considerado como una ventaja en la medida que también reduce la posibilidad de errores al reducir la cantidad de notación envuelta en un programa, permitiendo al estudiante enfocarse directamente en el problema que está resolviendo. No obstante, muchos especialistas consideran esto una desventaja, ya que el estudiante no tiene una visión clara del algoritmo que programa.

Al ser un código interpretado línea a línea, la respuesta inmediata del intérprete habilita una rápida e interactiva demostración de los conceptos de programación envueltos en un problema en particular. Añadiendo a esto el hecho de que los mensajes de error llegan de una manera más "natural" (en el orden en que aparecen en el código), muy diferente a los de los lenguajes compilados, a veces crípticos, cuestión en la que merecen mención especial los muchas veces "místicos" mensajes de error de C++.

La indentación, usada para delimitar la estructura del programa permitiendo establecer bloques de código, es una mayores ventajas de Python pues obliga a los estudiantes a adquirir hábitos correctos al escribir programas, algo que les será muy útil incluso cuando utilicen otros lenguajes donde la indentación no es obligatoria.

Posee excelentes herramientas para cadenas de caracteres y expresiones regulares. Es posible comenzar a estudiar el trabajo con cadenas de caracteres desde bien temprano, aumentando la amplitud de los problemas.

Las listas de Python son similares a los arreglos de todos los lenguajes, pero con la particularidad de que son totalmente heterogéneos. Los ciclos proveen un mecanismo integrado para la iteración sobre las listas. El diccionario integrado de Python permite introducir conceptos y usos relacionados con este tipo de datos de una manera sencilla.

Un módulo de Python es sencillamente un fichero, con listas y clases implementadas. Una de las principales ventajas de este tipo de desarrollo es que los estudiantes tienen acceso a los códigos fuentes de desarrolladores experimentados, lo que les permite observar de primera mano las mejores prácticas y facilitaría a los estudiantes más aventajados la posibilidad de avanzar mucho. Por otro lado, se ha demostrado que los estudiantes que comienzan con Python rápidamente modularizan sus programas en varios ficheros.

Tabla 1 El archiconocido Hola Mundo.

C++

```
#include <iostream>
int main(int argc, char **argv)
{
    std::cout << "Hola, Mundo" << std::endl;
return 0;
```

```
}  
}
```

Visual Basic

```
Private Sub Command1_Click()  
    MsgBox "Hola, Mundo"  
End Sub
```

Java

```
class Hi  
{  
    public static void main (String args[])  
    {  
        System.out.println("Hola, Mundo");  
    }  
}
```

Python

```
print ("Hola, Mundo")
```

El programa Hola Mundo, escrito en 4 lenguajes en la tabla, imprime el texto "Hola, Mundo". Nótese que en este programa tan sencillo, en C++ están implicados los conceptos de importación de librerías, función, función principal, argumentos, tipos de datos. En Java a esto se le adiciona el de clase, encapsulamiento y método estático. En Visual Basic, se entra directamente en la programación visual y conducida por eventos, el Hola Mundo en este caso es una ventana con botón que al evento click levanta un cuadro de mensaje con el texto "Hola Mundo". En Python sencillamente una función, print (imprimir en inglés), que imprime en la consola de Python el texto que recibe. Lógicamente una comparación profunda entre estos lenguajes, que no es objetivo de este trabajo, no solo utilizaría ejemplos tan sencillos. Pero lo que sí nos quiere decir este ejemplo es que este tipo de programas, propios de un curso introductorio a la programación, es mucho más sencillo de escribir y de explicar en Python.

El mecanismo integrado de documentación de Python es también una ventaja, pues enseña la importancia de la documentación al mismo tiempo que crea buenos hábitos en este sentido.

Python está disponible gratis, para Windows, MacOS, GNU/Linux y otros sistemas operativos. Más allá de la interminable polémica sobre cuál es el mejor sistema operativo, el software libre representa una alternativa muy seria si se considera solamente el factor económico. Es por ello que en nuestro país se ha hablado en varias ocasiones sobre una estrategia para la migración hacia el software libre en todos los ámbitos de la sociedad [20].

Al ser uno de los lenguajes más usado por la comunidad del software libre, esta ha desarrollado una considerable cantidad de herramientas, tutoriales, libros, cursos, ejercicios y documentación accesible libremente en Internet.

Ciertamente en este contexto Python puede ser un paso de avance. Por ser un lenguaje multiplataforma puede usarse, prácticamente de la misma manera en cualquier sistema operativo. No es necesario que la migración haya sido completada para enseñar Python, al contrario, puede ser útil en el proceso de migración. Claramente en este sentido constituye una fuerte ventaja sobre el actual Visual Basic, un lenguaje diseñado exclusivamente para Windows.

Desventajas

Como cualquier otro lenguaje industrial, existe mucho más acerca de Python que lo que se puede enseñar en un curso introductorio de programación. Se debe escoger cuidadosamente un

subconjunto del lenguaje consistente y efectivo para la enseñanza. Esto, que puede ser considerado como una desventaja de Python, es igualmente cierto en cualquier otro lenguaje utilizado por la industria.

En un ambiente diferente al cubano, que los estudiantes tuviesen la posibilidad de descargar las últimas versiones de Python pudiera considerarse como una dificultad, ya que los profesores necesitarían estar muy atentos a las últimas modificaciones del lenguaje[17]. Sin embargo en nuestro contexto, dado el limitado acceso que tenemos a Internet, esto no significa un problema. No obstante, que un lenguaje en constante crecimiento como Python fuerce a los docentes a mantenerse al tanto de los últimos avances en materia de programación, como desventaja no puede considerarse menos que dudosa.

Aunque el tipado dinámico de Python se consideró como una ventaja, también puede verse como una desventaja pues los estudiantes no tienen contacto directo con los conceptos asociados con los tipos, lo que puede derivar en una posible falta de comprensión cabal del programa y es una posible fuente de errores ya que los principiantes pueden tratar de realizar operaciones sobre dos tipos distintos. Por otro lado, que Python sea fuertemente tipado, es decir que no se puede tratar a una variable como un tipo distinto al dato que contiene sin hacer una conversión explícita, reduce la posibilidad de errores. En todo caso, la discusión sobre el tipado dinámico o estático en los lenguajes de programación es amplia, y múltiples los criterios desde la perspectiva de la educación.

Asimismo la condición de código interpretado, aunque posibilita la respuesta inmediata del sistema y la generación de mejores mensajes de error, deja al estudiante sin el concepto de la fase de compilación. En nuestro criterio, esta desventaja pudiera considerarse tal si se trata de estudiantes de informática, en cuyo caso el concepto de compilación sería obligatoriamente tratado en cursos posteriores. Sin embargo en el caso de los estudiantes preuniversitarios, el concepto de fase de compilación no es relevante para los objetivos del programa.

Considerando las ventajas y desventajas de Python analizadas anteriormente; su aceptación en la comunidad científica como uno de los lenguajes más adecuados para la enseñanza; el impacto que tendría para el proceso de migración al software libre; y el hecho de ser un lenguaje que ha demostrado evolucionar a la par de las últimas teorías en materia de lenguajes y técnicas de programación es que se propone su uso en la enseñanza de la programación en Cuba.

Bibliografía

- [1] J.M. Zelle: "Python as a first language", , 1999.
- [2] Guido van Rossum: "Computer Programming for Everybody", , 1999.
- [3] L. Grandell, M. Peltomäki, R.J. Back, T. Salakoski: "Why complicate things?: introducing programming in high school using Python", , pp. 71—80, 2006.
- [4] L. Mannila, M. De Raadt: "An objective comparison of languages for teaching introductory programming", , pp. 32—37, 2006.
- [5] F. Georgatos: "How applicable is Python as first computer language for teaching programming in a pre-university educational environment, from a teacher's point of view?", arXiv preprint arXiv:0809.1437, 2008.
- [6] N. Efford, T. Jenkins: "AND NOW FOR SOMETHING COMPLETELY DIFFERENT: LEARNING PROGRAMMING WITH PYTHON", , pp. 81—94, 2008.
- [7] : Autodesk Maya. 2012. URL www.autodesk.es/maya.
- [8] J.M. Zelle: Python Programming: An Introduction to Computer Science. Franklin, Beedle & Associates, Inc., 2004.
- [9] J. Elkner: "Using Python in a high school computer science program", , pp. 2000—01, 2000.
- [10] J. Elkner, A. Downey, C. Meyers: How to Think Like a Computer Scientist: Learning with Python.. 2001. URL [http://www.ibiblio.org/obp/thinkCSpy/..](http://www.ibiblio.org/obp/thinkCSpy/)
- [11] F. Stajano: "Python in education: Raising a generation of native speakers", , pp. 2000—01,

2000.

[12] J.A. Miller: Promoting Computer Literacy Through Programming Python. 2004.

[13] T. Jenkins: "The First Language—AC ASE FOR Python?", , 2003.

[14] V. Ceder, N. Yergler: "Teaching Programming with Python and PyGame", Apresentado na PyCon, 2003.

[15] W. McGugan: Beginning Game Development with Python and Pygame. Will McGugan, 2007.

[16] B.N. Miller, D.L. Ranum: "Teaching an introductory computer science sequence with Python", , 2005.

[17] J.D. Oldham: "What happens after Python in CS1?", Journal of computing sciences in colleges, pp. 7—13, 2005.

[18] wiki.python.org: "Schools Using Python", , 2011. URL <http://wiki.python.org/moin/SchoolsUsingPython>.

[19] R.M. Siegfried, D. Greco, N. Miceli, J. Siegfried: "Whatever Happened to Richard Reid s List of First Programming Languages", Information Systems Education Journal, pp. 24, 2012.

[20] Octavio Lavastida, Amaury E. del Valle: "Y del Linux qué (I y II)", Juventud Rebelde. Edicion digital., 2011. URL <http://www.juventudrebelde.cu/suplementos/informatica/2011-10-05/y-del-linux-que-i>.