



Ciencias Holguín

E-ISSN: 1027-2127

revista@ciget.holguin.inf.cu

Centro de Información y Gestión Tecnológica  
de Santiago de Cuba  
Cuba

Challenger-Pérez, Ivet; Díaz-Ricardo, Yanet; Becerra-García, Roberto Antonio

El lenguaje de programación Python

Ciencias Holguín, vol. XX, núm. 2, abril-junio, 2014, pp. 1-13

Centro de Información y Gestión Tecnológica de Santiago de Cuba

Holguín, Cuba

Disponible en: <http://www.redalyc.org/articulo.oa?id=181531232001>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

***El lenguaje de programación Python/The programming language Python***

Ivet Challenger-Pérez. [ichallengerP@facinf.uho.edu.cu](mailto:ichallengerP@facinf.uho.edu.cu)

Yanet Díaz-Ricardo. [ydiazr@facinf.uho.edu.cu](mailto:ydiazr@facinf.uho.edu.cu)

Roberto Antonio Becerra-García. [iderator@facinf.uho.edu.cu](mailto:iderator@facinf.uho.edu.cu)

**Institución de los autores**

Universidad de Holguín "Oscar Lucero Moya"

**PAÍS:** Cuba

**RESUMEN**

El software libre se ha convertido en uno de los movimientos tecnológicos de mayor auge en el siglo XXI. Para su desarrollo ha sido necesario contar con un grupo de herramientas que hagan óptima su utilización y sean fáciles de aprender. Python es un lenguaje de programación que cumple con lo planteado y se viene perfilando como una opción recomendada para el desarrollo de software libre. En este artículo se realizó un análisis de sus características fundamentales, así como de los principales usos que tiene en la actualidad.

**PALABRAS CLAVES:** SOFTWARE LIBRE; LENGUAJES DE PROGRAMACIÓN; PYTHON

**ABSTRACT**

Free Software has become one of the most important technological movements in the 21st century. Its development has required a set of tools that make its use more optimal and easy to learn. Python is a programming language that fills this requirement and is fast becoming a recommended choice for free software development. This paper makes an analysis of its main features, as well as the main cases in which it is employed today.

**KEY WORDS:** FREE SOFTWARE; PROGRAMMING LANGUAGES; PYTHON

**INTRODUCCIÓN**

Es el año 2013, y en la Industria Informática ha venido tomando influencia un fenómeno que se conoce como **software libre**. Este es un movimiento que proclama el acceso al código fuente de un programa, lo que admite ser libre de

uso, ejecución, distribución y modificación. Es decir, el nuevo software creado bajo este concepto podría emplearse para cualquier fin, ejecutarse en cualquier ambiente, distribuirse a discreción del propio usuario y modificarse de ser necesario.

Durante los años 90 ocurre una serie de eventos que marcan ciertos pautas para el futuro desarrollo del software libre, como es el lanzamiento de la primera versión del núcleo Linux por Linus Torvalds en 1991, y en ese mismo año Guido van Rossum libera la primera versión del lenguaje de programación Python.

Los lenguajes de programación son la herramienta básica de construcción de programas, como lo son el machete y el azadón para un campesino, el pico y la pala para un constructor. Python ha ido ganando adeptos en comunidades como la de software libre, científica y educacional, por su sencillez y posibilidad de concentrarse en los problemas actuales.

Este artículo hace referencia a las principales características y los diferentes usos de este lenguaje de programación, por lo que se ha tratado de simplificar la parte técnica. No obstante, para su completa comprensión se necesita un nivel básico de conocimientos acerca de programación.

## **MATERIALES Y MÉTODOS**

Para la elaboración de este trabajo se emplearon diferentes métodos de investigación, fundamentalmente teóricos, entre los que se destacan el histórico lógico y el análisis y la síntesis.

## **RESULTADOS DEL TRABAJO**

Python cuenta con facilidades para la programación orientada a objetos, imperativa y funcional, por lo que se considera un lenguaje multi-paradigmas. Fue basado en el lenguaje ABC y se dice que fue influenciado por otros como C, Algol 60, Modula-3 e Icon según su propio autor<sup>1</sup>.

---

<sup>1</sup> Kuchling, Andrew. Interview with Guido van Rossum. **Linux Journal** 1998 Nov # 55 [seriada en línea] <http://www.linuxjournal.com/article/2959> [consultado: 3 de octubre de 2012].

Es un lenguaje de alto nivel ya que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible.

## Historia



Python fue creado por Guido van Rossum, un programador holandés a finales de los 80 y principio de los 90 cuando se encontraba trabajando en el sistema operativo Amoeba. Primariamente se concibe para manejar excepciones y tener interfaces con Amoeba como sucesor del lenguaje ABC.

El 16 de octubre del 2000 se lanza Python 2.0 que contenía nuevas características como completa recolección de basura y completo soporte a Unicode. Pero el mayor avance lo constituye que este comenzó a ser verdaderamente desarrollado por la comunidad, bajo la dirección de Guido.

El Python 3.0 es una versión mayor e incompatible con las anteriores en muchos aspectos, que llega después de un largo período de pruebas el 3 de diciembre del 2008. Muchas de las características introducidas en la versión 3 han sido compatibilizadas en la versión 2.6 para hacer de forma más sencilla la transición entre estas.

A Guido van Rossum le fue otorgado el Free Software Award (Premio del Software Libre) en el 2001, por sus trabajos en la creación y desarrollo del lenguaje Python. En el 2005 fue contratado por Google, donde trabaja en la actualidad, aunque sigue liderando los esfuerzos en el desarrollo del Python.

## Cultura de Python

Los desarrolladores de Python han creado algo que se podía denominar como una subcultura dentro de la comunidad de desarrolladores. Es decir, estos tienen sus propias formas o metodologías de escribir código. Según el sitio web oficial de [Python](#) existe una lista de principios de diseño que se deberían seguir cuando se escribe código en Python:

- Hermoso es mejor que feo
- Explícito es mejor que implícito
- Simple es mejor que complejo
- Plano es mejor que anidado

- Disperso es mejor que denso
- El código legible cuenta
- Casos especiales no son lo suficientemente especiales para romper las reglas
- Casi siempre lo práctico vence a lo formal
- Los errores no deben pasar nunca desapercibidos, a menos que se especifique este comportamiento
- Ante una ambigüedad, descarte la tentación a adivinar
- Debe haber una, y preferentemente una sola, manera obvia de lograr algo, aunque esta generalmente no está clara a primera vista a menos que seas un genio
- Ahora es mejor que nunca, aunque en muchas ocasiones nunca es mejor que ahora mismo
- Si la implementación es difícil de explicar, entonces es una mala idea
- Si la implementación es fácil de explicar, entonces pudiera ser una buena idea
- Los espacios de nombre son una buena idea, hagamos más de eso

Esta lista<sup>2</sup>, que pudiera tomar libros en explicar en toda su totalidad, simplemente expone que mientras más sencilla y clara se mantengan e implementen las ideas, mejores serán estas.

## Eventos

Python, al convertirse en un fenómeno internacional, cuenta con un grupo de eventos que se realizan para difundir y discutir ideas respecto a este. Al igual que la mayoría de las tecnologías importantes, cuenta con una conferencia oficial, el PyCon (Python Conference) que se realiza una vez al año en varias ciudades del mundo.

El PyCon es una gran fuente de conocimientos para aquellos interesados en adentrarse en diversos temas y tecnologías de Python, además de ser una buena opción para exponer productos y tecnologías relacionadas con este.

---

<sup>2</sup> Peters, Tim. The Zen of Python. 2004 [documento en línea]

<http://www.python.org/dev/peps/pep-0020/> [consultado: 3 de octubre de 2012].

Existen también otras conferencias con tópicos específicos como la de SIPI, donde se debate básicamente el uso del Python por la comunidad científica, entre otras.

### Sintaxis

La sintaxis de Python es muy sencilla, tanto que en algunas ocasiones parece pseudocódigo. Es muy interesante observar las diferencias que existen entre el programa Hola Mundo de Python y el de otro lenguaje de alto nivel como C++:

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello World" <<
endl;
    return 0;
}
```

**Fig. 1 Programa Hola Mundo en C++**

Y ahora se ve cómo lograr esta misma tarea en Python:

```
print "Hello World"
```

**Fig. 2 Programa Hola Mundo en Python**

No es necesario explicar mucho sus diferencias porque resaltan a simple vista, en Python es más simple que en C++. Y esto es de vital importancia para ramas como la educación, donde al utilizar lenguajes como C++, generalmente se escuchan frases como “No te preocupes por eso, más adelante en el curso lo veremos”<sup>3</sup> que pueden ser muy dañinas en el proceso de enseñar y motivar a personas a que aprendan programación.

### Intérprete al vuelo

Python cuenta con un intérprete o consola que permite probar ciertas capacidades del lenguaje sin tener que crear un módulo de este. Incluso, en ocasiones se utiliza como una poderosa calculadora, con capacidades muy

---

<sup>3</sup> Downey, Allen B. How to Think Like a Computer Scientist. Learning with Python, Green Tea Press, 2008.

similares a las que ofrece el lenguaje Matlab para la realización de ciertas tareas.

### **Librería Estándar**

Una de las fortalezas de Python, y quizás la mayor, es la librería estándar con que cuenta. Con decenas de módulos cubre la mayoría de las necesidades básicas de un programador y mucho más. En esta se le da cobertura de forma muy intuitiva a tópicos como:

- Cadenas
- Estructura de datos
- Funciones numéricas y matemáticas
- Compresión de datos
- Formatos de archivo
- Criptografía
- Servicios de los Sistemas Operativos
- Comunicación entre Procesos
- Manejo de datos de Internet
- Servicios multimedia
- Manejo de excepciones

Existen más de 40 tópicos en la referencia de la librería de Python<sup>4</sup> por lo que se puede afirmar que es una de las más completas con que se cuenta en la actualidad, comparable con la de Java y .NET.

### **Rendimiento**

Uno de los grandes mitos alrededor de Python es acerca de su pobre rendimiento. Esto no es del todo correcto, ya que aunque es un lenguaje interpretado y estos tienden a ser más lentos que los lenguajes compilados, Python, a diferencia de otros lenguajes interpretados, ha implementado toda su librería estándar en el lenguaje C, lo que hace que sus funciones primitivas sean bastante eficientes. Además, puede compilarse su código a bytecodes, similar al que usan Java y .NET, lo que optimiza aún más el proceso de interpretación.

### **Documentación**

---

<sup>4</sup> Guido van Rossum y F.L. Drake. The Python Library Reference. Release 2.6.4. Ene. 2010.

Otra de las características principales de Python es la inclusión de un sistema de documentación. La documentación fue tomada en cuenta a la hora del diseño del lenguaje. Por ejemplo, todas las funciones y clases tienen una variable implícita que se llama *doc* comúnmente llamada docstring o cadena de documentación<sup>5</sup>. En estas cadenas se describe el uso de esa función o clases y los parámetros que intervienen en detalle. La sintaxis de estos docstring es muy similar a la de JavaDoc.

Estas cadenas de documentación pueden llamarse inclusive en tiempo de ejecución, por lo que se pueden consultar en el propio intérprete usando la función `help`, modelo muy similar al empleado en Matlab. La librería estándar de Python fue documentada utilizando este método.

### **Extensibilidad**

De todas las características que posee Python, una de las más importantes es su capacidad de reutilizar código escrito en los lenguajes C y C++. Existen mecanismos que hacen muy sencilla la tarea de envolver funciones y clases hechas en estos lenguajes, entre los que se encuentran Boost.Python, Sip y Shiboken.

La importancia de esta integración es relevante, ya que las bases de código en lenguajes como C y C++ son las más grandes disponibles por el software libre hoy en día, y permiten no tener que duplicar código ya existente.

### **Licencia**

Python es liberado bajo una licencia propia llamada Python Licence que ha sido certificada por el movimiento Open Source, y es compatible con la GPL (GNU Public Licence) de la Free Software Foundation (Fundación del Software Libre). La principal diferencia de esta con respecto a la GPL es que no existe una restricción copyleft en esta, concepto que la acerca más al movimiento OpenSource. Lo que quiere decir que se pueden producir programas que se podrán distribuir libremente sin la necesidad de entregar su código fuente. Esto significa que puede usarse Python tanto para hacer software libre como software privativo.

---

<sup>5</sup> González Duque, Raúl. Python para todos. [documento en línea]

<http://mundogeek.net/tutorial-python/> [consultado: 2 de octubre de 2012].



## Herramientas

A un lenguaje de programación le es vital contar con herramientas que hagan más productivo el trabajo con este. Algunas de las más importantes de Python son:

### ▪ Implementaciones

Al igual que Java, que cuenta con distintas máquinas virtuales como IceTea, Sun JDK, OpenJDK, etc.; Python posee varias implementaciones. La más importante, y que es la que se mantiene oficialmente, es CPython, elaborada completamente en lenguaje C y se considera como la más estable de todas.

Después de esta existen otras importantes, como IronPython escrita en C# y diseñada para la plataforma .NET, Jython implementada en Java y compila para el bytecode de esta plataforma, PyPy realizada con el propio Python y destinada para correr en diversas plataformas, y tinypy una implementación minimalista que fue diseñada para caber en 64 Kb de memoria, muy útil para sistemas embebidos.

### ▪ Entornos de Desarrollo Integrado

Dado el éxito tenido por Python en plataformas Unix, se han elaborado para este una gran cantidad de entornos de desarrollo integrado, considerados entre los de mejor calidad:

1- IDLE (Integrated DeveLopment Environment), un entorno de desarrollo bastante simple desarrollado en el propio Python, por lo cual cuenta con un intérprete al vuelo y fue desarrollado usando Tkinter. Este es un entorno ideal para pequeñas aplicaciones, debido a su sencillez, además de que se distribuye junto con los paquetes de Python para los sistemas operativos más comunes: Windows, Linux, MacOS. Cuenta con resaltado de código y depurador.

2- Eclipse con el plugin PyDev. El Eclipse es una plataforma de desarrollo de aplicaciones llamada RCP, en la cual se montan varios plugins que extienden su funcionalidad. En el caso de PyDev, cuenta con un sistema de proyectos que ayuda a organizar las aplicaciones, además de tener completado inteligente de código, depurador, resaltado de código, gestión de documentación, entre otras muchas características que lo hacen ideal para medianos y grandes proyectos.

## ▪ **Servidores de Aplicaciones**

La web se ha convertido hoy en el centro de la vida digital del planeta. Un lenguaje que no cuente con el debido soporte técnico de los distintos servidores de aplicaciones web puede considerarse un lenguaje obsoleto.

Python puede ser usado en conjunto con el servidor web Apache vía `mod_python`, gracias a la API Web Server Gateway Interface, desarrollada para crear un estándar que sirviera para estos fines.

Python cuenta con otros servidores de aplicaciones y frameworks que hacen posible el desarrollo de aplicaciones web, como Zope, Pylons, web2py, TurboGears y Django. De esta lista resalta el Django, que es un framework muy productivo que cuenta con características de tan alto nivel, como un ORM (Object to Relational Mapper), y un sistema de plantilla muy útil a la hora de elaborar formularios.

## **Casos de Éxito**

Una medida principal para saber cuán exitoso es un lenguaje es saber quién lo usa, y qué uso le da a este. A continuación se muestran los principales usos que se le da a Python hoy en día.

### • **Empresas que emplean Python comercialmente**

Google, uno de los gigantes de la informática y el Internet, ha empleado el Python para construir una gran parte de sus algoritmos de búsqueda para la web, no por gusto esta compañía contrató a Guido van Rossum para trabajar para ellos.

Otras empresas como Lucasfilm lo usan como plataforma de guiones (scripts) para la elaboración de los motores de representado en 3D para sus películas.

### • **Productos libres en Python**

Python es un lenguaje que está de moda para escribir productos de software libre, la mayoría de las grandes distribuciones lo usan como estándar para escribir sus aplicaciones, entre las que se destaca Ubuntu.

El Gimp, un paquete de retoque fotográfico similar al Adobe Photoshop, usa como lenguaje de guiones para la realización de filtros a la medida Python.

El Blender, un software para modelado 3D, similar al 3D Studio o al Maya, también está escrito en Python.

### • **Python para la Educación**

Una de las aplicaciones para la que fue concebido Python en sus principios fue como un lenguaje fácil de aprender. Su creador ha expresado que llegará el día donde la programación se convierta en una asignatura tan importante como las matemáticas y las físicas para los currículos de la enseñanza media<sup>6</sup>.

Universidades como la prestigiosa MIT (Massachusetts Institute of Technology) lo han seleccionado para impartir cursos como Introducción a las Ciencias de la Computación y a la Programación e Introducción a los Algoritmos. En la Universidad de Chicago se utiliza para impartir Introducción a las Ciencias de la Computación 2, y en la Universidad de Jaume de España se usa en Introducción a la Programación.

Se puede ver cómo en algunos centros de estudios superiores importantes ya se viene adoptando el Python como lenguaje, tanto para cursos introductorios como para cursos un poco más avanzado. Esto indica cuánto ha avanzado el lenguaje en este campo y cuáles serán sus proyecciones de adopción en el futuro.

- **Python para la Ciencia**

Hoy en día, los científicos de cualquier rama usan la programación como una herramienta más para la solución de sus problemas. Ellos necesitan que sus herramientas sean simples y eficientes, de modo que se puedan concentrar en los problemas de sus respectivos campos.

Actualmente se puede observar una gran tendencia al uso de Python en grandes centros de investigación como el CERN (Organización Europea para la Investigación Nuclear) y por parte de científicos en ramas como la Bioinformática, Neurofisiología, Física, Matemáticas, etc. Esto es debido a la disponibilidad de librerías de visualización, procesamiento de señales, estadísticas, álgebra, etc.; de fácil utilización y que cuentan con muy buena documentación.

Los paquetes NumPy (Python Numérico) y SciPy (Python Científico) son pilares para el trabajo científico hoy en día, estos emulan en muchas ocasiones las funciones disponibles por el Matlab (otro de los lenguajes de corte científico existentes) lo que hace más fácil la transición a partir de este.

---

<sup>6</sup> Guido van Rossum, Computer Programming for Everybody. A Scouting Expedition for the Programmers of Tomorrow. 1999.

El Toolkit Modular para Procesamiento de Datos (MDP), por ejemplo, ofrece funciones para temas más específicos y avanzados como el Análisis de Componentes Principales y Análisis de Componentes Independientes, muy útiles para el procesamiento de señales digitales.

Otra de las razones de su uso es la existencia de librerías que soportan archivos de intercambio de datos científicos como el netCDF, además de que pueden leer de ficheros de datos generados con Matlab. Esto hace mucho más fácil y estandarizado el intercambio de datos entre la comunidad científica.

La madurez de proyectos de código abierto, como NumPy y SciPy ha dado acceso a un gran número de funciones científicas que compiten en tamaño y velocidad con la bien conocida alternativa comercial Matlab, solidificando el papel de Python en la comunidad científica de Python hoy.

## CONCLUSIONES

La programación es considerada por muchos un arte, y como todo arte necesita un lenguaje que permita expresar las ideas de la forma que se quiere. Python es el lienzo que permite reflejar, de forma simple y elegante, las ideas en forma algorítmica. Sus aplicaciones, tanto en la comunidad docente como en la científica, le permitirán aumentar su popularidad y adopción a nivel internacional.

Sirva esta información especialmente a la comunidad cubana, donde hay muchas personas, informáticos o no de titulación, interesadas en dar sus primeros pasos en el mundo de la programación en un ambiente de software libre, pues Python constituye, sin lugar a dudas, una de las mejores variantes.

## BIBLIOGRAFÍA

- [1]. Downey, A.B. (2008). *How to Think Like a Computer Scientist*. Learning with Python, Green Tea Press.
- [2]. Finlay, J. (2006). PyGTK 2.0 Tutorial, Version 2.5. [documento en línea]. <http://www.pygtk.org/pygtk2tutorial/index.html> [consultado: 3 de octubre de 2012].
- [3]. González Duque, R. Python para todos [documento en línea]. <http://mundogeek.net/tutorial-python/> [consultado: 2 de octubre de 2012].

- [4]. Rossum, G. (1999). Computer Programming for Everybody. A Scouting Expedition for the Programmers of Tomorrow.
- [5]. Rossum, G . y Drake, F.L. (2010). The Python Library Reference. Release 2.6.4.
- [6]. Kuchling, A. (1998). Interview with Guido van Rossum [versión electrónica]. *Linux Journal* 55 disponible en: <http://www.linuxjournal.com/article/2959> [consultado: 3 /10/ 2012].
- [7]. O'Reilly Python DevCenter [documento en línea] <http://www.onlamp.com/python> [consultado: 2 /10/ 2012].
- [8]. Peters, T. (2004). The Zen of Python. [documento en línea] <http://www.python.org/dev/peps/pep-0020/> [consultado: 3 /10/ 2012].
- [9]. Sitio Oficial de Python ([www.python.org](http://www.python.org)) [consultado: 3 /10/ 2012].

## **Síntesis Curricular de los Autores**

**Ing. Ivet Challenger-Pérez** Ingeniero Informático. Profesor. Dpto. de Informática.  
*Investigaciones realizadas:* Sistema para la gestión de la información de los medios audiovisuales.

*Ha impartido posgrado como:* Ingeniería de Software (UML), Introducción a la Plataforma de Aprendizaje a Distancia (Moodle) para administradores.

*Cursos de posgrado recibidos:* Seguridad Informática, Oficina para la Seguridad de las Redes Informáticas, Software Libre, Metodología de la Investigación Científica, Diplomado Docencia Universitaria

*Participación en eventos:* Tercer Taller Provincial de Informatización de la Sociedad, Holguín, V Conferencia Científica Internacional Universidad de Holguín "Oscar Lucero Moya",

**Ing. Yanet Díaz-Ricardo** Ingeniero Informático. Profesor. Dpto. de Informática.  
*Investigaciones realizadas:* Sistema para la Gestión de Contratos Económicos en la Zona Oriente Norte, ECASA S.A.

*Cursos de posgrado impartidos:* Curso Introductorio a la Plataforma de Aprendizaje a Distancia (Moodle) para administradores, Ingeniería de Software (UML), Instalación, configuración y administración del CMS Joomla.

*Cursos de posgrado recibidos:* Seguridad Informática, Oficina para la Seguridad de las Redes Informáticas, Dirección y Administración, Metodología de la Investigación Científica, Diplomado Docencia Universitaria

*Participación en eventos:* IV Conferencia Científica Internacional, V Conferencia Científica Internacional, XVI Forum Municipal de Ciencia y Técnica, Relevante, XVI Forum Provincial de Ciencia y Técnica, Destacado.

## **Ing. Roberto Antonio Becerra-García**

Profesor instructor en la Facultad de Matemática e Informática e imparte Arquitectura de Computadoras a estudiantes de Ingeniería Informática. Graduado en Ingeniería Informática, cursa la maestría de "Matemática e Informática Aplicada a la Administración" y el doctorado curricular "Modelos Matemáticos y Aplicaciones". Su área de investigación se encuentra alrededor del procesamiento de datos biomédicos, en específico de señales electrooculográficas. Como resultado de sus investigaciones ha realizado una aplicación de procesamiento de registros electrooculográficos siendo autor de varios artículos científicos en este campo.

## **Institución de los autores**

Universidad de Holguín "Oscar Lucero Moya"

**Fecha de Recepción:** 14/01/2013

**Fecha de Aprobación:** 12/12/2013

**Fecha de Publicación:** 16/04/2014