

Congreso Latinoamericano de Educación Superior “LATINOAMERICA EDUCA”

Expositor: Ing. Cynthia Lorena Corso.

Temario de Exposición.Lenguaje Python

1. Orígenes.
2. Características.
3. Ventajas.
4. Librerías Utilizadas.
5. Razones para aprender Python como primer lenguaje de programación.

Lenguaje de Programación PYTHON

Introducción

Python es habitualmente comparado a TCL, Perl, Java y Ruby. Actualmente, Python se desarrolla como un proyecto de código abierto administrado por la Python Software Foundation. La última versión estable del lenguaje es actualmente (Marzo 2006) la 2.4.3. Guido van Rossum creó Python(1989), un lenguaje de programación de scripting, la "oposición leal" a Perl.

En origen era un lenguaje para la gestión de Amoeba, basado en ABC y Modula-3, usando para crear interfaces entre módulos.

En febrero de 1991, su creador lo cuelga en USENET. Liberado bajo una licencia de tipo BSD.

Características

Python es un lenguaje de programación interpretado, orientado a objetos, multiplataforma y de sintaxis sencilla.

Otra característica de Python, es la manera de organizar sus instrucciones, es decir permite dividir su programa en módulos reutilizables desde otros programas en Python.

El lenguaje incorpora una gran colección de módulos estándar que puedes utilizar como base de los programas (o como ejemplos para empezar a aprender Python). También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema, ``sockets" y hasta interfaces gráficas con el usuario .

Python permite escribir programas muy compactos y legibles. Con compactos se quiere expresar que con muy pocas líneas de código se puede lograr diversas funcionalidades. Los programas escritos en Python son normalmente mucho más cortos que sus equivalentes en C o C++, por varios motivos:

1. Los tipos de datos de alto nivel permiten expresar operaciones complejas en una sola sentencia.
2. El agrupamiento de sentencias se realiza mediante sangrado (indentación) en lugar de begin/end o llaves.
3. No es necesario declarar los argumentos ni las variables.

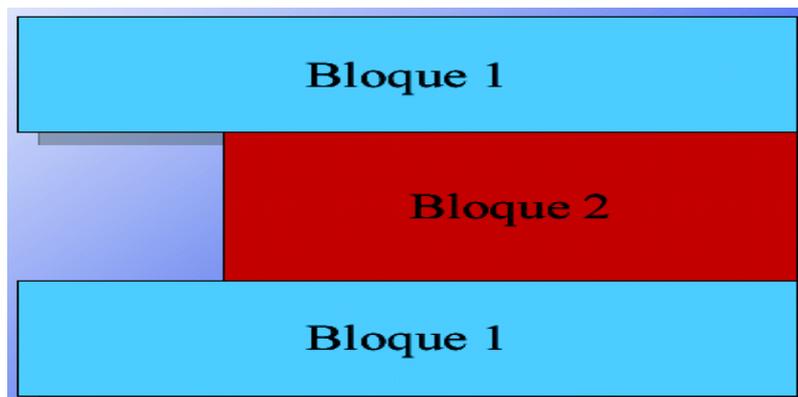
Python es *ampliable* o conocido como *lenguaje Integrador*. Si se conoce un lenguaje como C, es fácil añadir una nueva función o módulo al intérprete, para realizar operaciones críticas a la máxima velocidad.

Una vez integrado, puedes enlazar el intérprete de Python a una aplicación escrita en C y utilizarlo como lenguaje de macros para dicha aplicación.

Otra característica atrayente es que sintáticamente escribir un programa en este lenguaje es muy sencillo y “limpio”.

La razón de esto es que Python usa tabulación o (espaciado) para mostrar estructura de bloques.

- a. Tabula una vez para indicar el comienzo de un bloque.
- b. Des-tabula para indicar el final del bloque.



Código en C/Java	Código en Python
<pre>if (x) { if (y) { f1(); } f2(); }</pre>	<pre>if x: if y: f1() f2()</pre>

En la figura que se muestra anteriormente, se evidencia la facilidad para escribir instrucciones en este lenguaje con respecto a otros.

Utilidades del Lenguaje

Python es ideal:

- a. Como lenguaje "integrador" para combinar varios componentes de un programa.
- b. Para llevar a cabo prototipos del sistema.
- c. Para elaboración de aplicaciones cliente.
- d. Para desarrollos web y de sistemas distribuidos.
- e. Para el desarrollo de tareas científicas, en los que hay que simular y prototipar rápidamente.
- f. Como primer lenguaje para el aprender, esto se fundamenta en muchos aspectos que detallará mas adelante.
- g. Y mucho más....

Como se sabe no hay un único lenguaje que cubra todas los requerimientos, pero es importante mencionar que el mismo no es muy adecuado para realizar tareas relacionadas con:

- a. Programación de bajo nivel (system-programming), como programación de drivers y kernels.
- b. Python es de alto nivel, no hay control directo sobre memoria y otras tareas de bajo nivel.

Librerías de Python

Una de las potencialidades de este lenguaje, como se menciona anteriormente, es que dispone una gran cantidad de librerías, para la realización de diversos propósitos como:

- Acceso a Ficheros, manejo de cadenas.
- Servicios Web.
- Retoque de imágenes.
- Multimedia.
- Interfaces gráficas.
- XML
- Creación de PDFs.(Librería ReportLab)

- Acceso a Base de Datos.
- Estadísticas.
- Muchas más para diversos propósitos.

Librerías para el desarrollo de interfaces gráficas.

En Python existe varias alternativas para la creación de interfaces gráficas. Algunas de las librerías utilizadas para la creación de interfaces gráficas son:

Tkinter: Es la interfaz gráfica de Python usando TK. Es un módulo Python que provee una colección de clases Python y métodos, para acceder a las herramientas TK con Python. Este viene incluido en Python por lo que se puede decir que es casi un standard de él. Se distribuye junto con el propio interprete de Python, es multiplataforma y esta muy bien documentado.

Pmw(MegaWidgets de Python): esta librería incluye una caja de herramientas para los construcción de widgets de alto nivel, tomando como base el módulo de Tkinter. La creación de esta librería surge, como una necesidad de la escasa cantidad de widgets de la librería Tkinter.

GTK es una interfaz orientada a objetos para programadores de aplicaciones (API). Se puede utilizar conjuntamente con Glade. Este programa es bastante sencillo de utilizar, la idea principal es que el mismo tiene una barra de herramientas que contiene los widgets y de cada uno de ellos pueden modificarse sus atributos y conectar las señales que emiten funcionalidad de nuestro programa en Python.

Aprender lenguaje Python como primer lenguaje.

Siempre ha existido un debate entre los docentes de las carreras de informática, con respecto a la elección del primer lenguaje de programación.

Otra pregunta que se ha planteado es: que características o requisitos debe reunir un lenguaje para considerarlo un buen lenguaje para iniciar a los alumnos en la programación.

La elección de aprender Python conjuntamente con otro lenguaje (C,C++), se fundamenta en una serie de ventajas. Uno de ellos es la sintaxis, es decir economía en el uso de símbolos auxiliares. Se puede comparar con otros lenguajes que han adoptado la gran mayoría de las universidades.

C	C++	Java	Python
<pre>#include<stdio.h> int main(void) { printf("Hola, mundo.\n"); return 0; }</pre>	<pre>#include <iostream> int main(void) { std::cout << "Hola, mundo." << std::endl; }</pre>	<pre>public class HolaMundo { public static void main(String [] args) { System.out.println("Hola, mundo."); }}</pre>	<pre>print "Hola Mundo"</pre>

Este lenguaje además incluye una vasta cantidad de librerías que posibilita al alumno explorar distintas funciones que se pueden llevar a cabo con este lenguaje.(Aplicaciones web, serialización de objetos, interfaces gráficas etc)

La actividad de programar que en fases tempranas del aprendizaje, se basa en el método de prueba y error. Con este lenguaje es posible manejar un entorno de programación que facilita la fase de ejecución de una manera más rápida.

Cabe aclarar que los lenguajes de scripts de alto nivel e interpretados como Python, Perl tienen una tendencia creciente en el mundo del software libre. La ventaja de complementar el aprendizaje del lenguaje C,C++ o Java con otro alternativo como Python, es que el alumno a esta instancia ya ha aprendido programar.

Referencias

- *Web de Python*(inglés), disponible en <http://www.python.org>
- *Notas sobre Lenguaje Python*(Moreira Walter) Este artículo describe las generalidades del lenguaje como características, tipos de datos, clases. etc.
<http://www.cmat.edu.uy/~walterm/python/pynotas.pdf>
- *Aprender a Programar Python: una experiencia docente.*(Marzal Andrés, Llorens David, García Isabel)
Este presentación reflexiona sobre la idoneidad de Python, como primer lenguaje de programación y la experiencia docente que significó la implementación del mismo.
http://www.aditel.org/jornadas/03/ponencias/py_c
- *Interfaces múltiples en Python*(Sánchez Marcos Provencio).El artículo es una breve introducción de alternativas para el desarrollo de interfaces gráficas (Tkinter) y un servidor de aplicaciones.
http://es.tldp.org/Presentaciones/200103hispalinux/sanchez/pdf/hl_python.pdf
- *Integración de Python y C*(Marzal Andrés).Esta presentación muestra por medio de un ejemplo práctico, la facilidad de integrar rutinas de C en Python.
http://www.aditel.org/jornadas/03/ponencias/py_c