

Introducción a:



Introducción a Python

- Python es un lenguaje de programación muy potente creado por Guido Van Rossum.
- El nombre de Python viene de la afición del creador al grupo de humoristas Monty Python. La versión 1.0 de Python fue publicada en enero de 1994. La última versión publicada durante la estancia de Rossum en el CWI fue la 1.2. Continuaría el desarrollo de Python en la CNRI en Reston, Virginia.

Características

- Es un lenguaje interpretado
- Sintaxis muy sencilla que permite su fácil lectura

- **Legibilidad**

La sintaxis de Python es muy elegante e intuitiva y permite una muy buena lectura durante la programación.

- **Expresividad**

Los programas en Python son muy compactos, pues se pueden programar cosas muy complejas en pocas líneas.

- **Sencillo**

No requiere de llaves u otros elementos para indicar que una instrucción forma parte de un bucle o una condición, sino que se utilizan los espacios que también utilizan la mayoría de programadores para aumentar la legibilidad de su código.

- Gran variedad de recursos y librerías

- Multiparadigma

Programación orientada a objetos, programación funcional...

- Multiplataforma

Puede correr en diferentes sistemas operativos.

Filosofía

```
Python 3.4.3 (default, Sep 12 2015, 15:13:07)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>> 
```

GRACIAS POR SU ATENCIÓN

Sandra Lorena Hoyos
Cead- Palmira