



CONCEPTOS INICIALES

PROGRAMACIÓN III

Abdel G. Martínez L.

CONCEPTO Y DEFINICIÓN

- ⦿ Lenguaje de programación interpretado
- ⦿ Creado por Guido van Rossum
- ⦿ Administrado por PSF
- ⦿ Su nombre proviene de Monty Python
- ⦿ Multiparadigma: orientación a objetos, a eventos, funcional, estructurada.
- ⦿ Dinámicamente y fuertemente tipado
- ⦿ Conteo de referencias



CARACTERÍSTICAS

- ⦿ Sintaxis sencilla y elegante
- ⦿ Ideal para desarrollo de prototipos
- ⦿ Viene con una amplia librería
- ⦿ El modo interactivo es útil para pruebas cortas
- ⦿ Es fácil de extender usando módulos creados en C y C++
- ⦿ Puede ser embebido en aplicaciones
- ⦿ Se ejecuta en múltiples plataformas y sistemas operativos
- ⦿ Es software libre

¿QUIÉN USA PYTHON?

- ◉ Google
- ◉ PBS
- ◉ NASA
- ◉ Yahoo
- ◉ Dropbox
- ◉ Eventbrite
- ◉ Mozilla
- ◉ Quora
- ◉ Pinterest
- ◉ Slideshare
- ◉ SurveyMonkey
- ◉ Red Hat
- ◉ Oracle
- ◉ Linode
- ◉ Cisco
- ◉ Yelp

ZEN DE PYTHON

- ⦿ Bello es mejor que feo
- ⦿ Explícito es mejor que implícito
- ⦿ Simple es mejor que complejo
- ⦿ Complejo es mejor que complicado
- ⦿ Plano es mejor que anidado
- ⦿ Disperso es mejor que denso
- ⦿ La legibilidad cuenta
- ⦿ Los casos especiales no son tan especiales para quebrantar reglas

ZEN DE PYTHON

- ⦿ Aunque lo práctico gana a la pureza
- ⦿ Los errores nunca deberían dejarse pasar silenciosamente
- ⦿ A menos que hayan sido silenciados explícitamente
- ⦿ Frente a la ambigüedad, rechaza la oportunidad de adivinar
- ⦿ Debería haber una, y preferiblemente una, manera obvia de hacerlo
- ⦿ Esa manera no es obvia al inicio a menos que seas holandés
- ⦿ Ahora es mejor que nunca
- ⦿ Aunque nunca es a menudo mejor que ya mismo

ZEN DE PYTHON

- ⦿ Si la implementación es difícil de explicar, es una mala idea
- ⦿ Si la implementación es fácil de explicar, puede que sea buena idea
- ⦿ Los espacios de nombre son una gran idea, hagamos más de eso

VERSIONES

- ⦿ Creado en 1989
- ⦿ **Python 1.0** lanzado en 1994
- ⦿ **Python 2.0** lanzado en 2000
- ⦿ **Python 3.0** lanzado en 2008
- ⦿ Existe una dualidad de versiones:
 - **Python 2.7.10** lanzado en 2015
 - **Python 3.4.3** lanzado en 2015
 - Entonces, *¿qué aprendo?*

DIFERENCIAS ENTRE PYTHON 2.7.X Y 3.X

- ⦿ Para imprimir se usa `print`, que es una función no una sentencia
- ⦿ Se utilizan vistas e iteradores en lugar de vistas
- ⦿ Las reglas de ordenamiento han sido simplificadas
- ⦿ Existe un único tipo de entero: `int`
- ⦿ La división de dos enteros da como resultado un flotante
- ⦿ Todo el texto es Unicode; Unicode codificado es data binaria
- ⦿ ¿Por qué todavía usan Python 2.7?
 - *Por las librerías. Algunas no están portadas a Python 3.*