Informática para Científicos

Introducción al Lenguaje Python

Prof Pablo García Tahoces

Dpto Electrónica y Computación

ETSE. USC.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Introducción a Python

Lenguaje de programación interpretado, de alto nivel, procedimental (aunque también permite crear objetos).

Conceptualmente fue creado a finales de los 80's por Guido van Rossum en el marco del proyecto *Amoeba* del CWI (*Centrum Wiskunde & Informatica*, Holanda), heredando características del lenguaje ABC.

El nombre proviene de la afición del autor por el grupo humorístico inglés *Monty Python.*

Introducción a Python. Versiones.

Python 2 vs Python 3.

Python 2.x → 2.7 → Mediados 2010 → El Legado.

Python 3.x → 3.0 →2008 → El Futuro → Todas las mejoras en las librerías estándar solo están disponibles en Python 3.x.

3.3 → 2012
3.4 → 2014
3.7 → 2019
…
3.11 → 2022

¿Qué versión debo usar? → Analizar dependencias de aplicaciones con Python 2.x.

Introducción a Python. Versiones.

Python 3. Principales Cambios:

- Mejor soporte para *unicode* (todas las cadenas unicode por defecto).
- La sentencia *print*, ha sido sustituida por la función *print()*.
- Se han simplificado las reglas para los comparadores (< > → !=).
- Sólo hay un tipo de entero (long desaparece).
- Expresiones del tipo 1/2 retornan flotante por defecto.
- *range()* no retorna una lista (se comporta como *xrange*() de la versión 2).
- Otros …

Introducción a Python. Sintáxis Básica.

Palabras reservadas:

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Introducción a Python. Sintáxis Básica.

Identificador Python.- nombre utilizado por el lenguaje para identificar variables, funciones, clases, módulos u otros objetos. Seguirán las siguientes premisas:

- Comienza por letra [A-Z] o [a-z] o _, seguido de cero o más letras subrayados y dígitos [0-9].
- No se permiten otros símbolos tales como: @, $, %.
- Sensible a mayúscula/minúscula.
- No se permite el uso de palabras reservadas.

Introducción a Python. Sintáxis Básica.

Sintaxis Básica:

- # → Símbolo de comentario.
- ; → Símbolo de fin de instrucción.
- \ → Símbolo de continuación de línea de código.
- ‘ … ‘ o “ … “ → Símbolo de cadena de caracteres.
- “”” … “”” → Símbolo de cadena de caracteres incluyendo retorno de carro y/o
  → Símbolo de comentario multilínea.

Introducción a Python. Tipos de Datos.

Tipos de Datos Python:

- Números.- int (-3), float (0.5) , complex (3+4j).
  - float(), int(), round(), …
- Booleanos.- True, False
- Cadenas de caracteres.- *str= 'Hola Mundo'* → *str[2]*
  - *ord('H')* → devuelve el ASCII de *H.*
  - *chr(87)* → devuelve el carácter de ASCII *87*

Introducción a Python. Tipos de Datos.

Tipos de Datos Python:

- Listas.- *lst= [‘hola’, 34, 2.34, ‘no’] → lst[1]*
  - Son mutables, heterogéneas y ordenadas.
  - Listas de listas → matrices.- m=[[1,2],[3,4]] → m[0][1]

- Tuplas.- *tp= (‘hola’, 34, 2.34, ‘no’) → tp[2]*
  - Como listas pero no mutables

- Diccionarios.- *dict= {‘nombre’: ‘Pablo’, ‘edad’: 40, ‘sexo’: ‘h’} → dict[‘edad’]*
  - Permiten establecer relaciones (mapeos) entre claves y valores
  - Son mutables y no ordenadas

- Conjuntos (set).- s= set([1, ‘hola’, 3.0])
  - Mutables, no ordenados, no contienen duplicados
  - s.pop() → Devuelve un elemento de un conjunto y lo elimina del mismo.

- Conjuntos (frozenset).- fs= frozenset([1, ‘hola’, 3.0])
  - Como listas pero no mutables

Introducción a Python. Operadores de Asignación.

Operador	Description	Ejemplo
=	Operador asignación	c = a + b → c= 30
+=	Adición sobre valor actual	c += a → c = c + a
-=	Subtracción sobre valor actual	c -= a →c = c - a
*=	Multiplicación sobre valor actual	c *= a → c = c * a
/=	División sobre valor actual	c /= a → c = c / a
%=	División Modulo sobre valor actual	c %= a → c = c % a
**=	Exponenciación sobre valor actual	c **= a → c = c ** a
//=	División Floor sobre valor actual	c //= a → c = c // a

a= 10, b= 20

Introducción a Python. Operadores Aritméticos.

Operador	Description	Ejemplo
+	Adición	a + b → 30
-	Sustraccion	a – b → -10
*	Multiplicacion	a * b → 200
/	Division	b / a → 2
%	División Módulo → Devuelve el resto.	b % a → 0
**	Exponenciación	a**b → 10^{20}
//	Division Floor → Devuelve cociente sin decimales	9//2 →4 9.0//2.0 → 4.0a

a= 10, b= 20

Otros → @ (Producto escalar/matricial para arrays)

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

Introducción a Python. Operadores Relacionales.

Operador	Description	Ejemplo
==	Comprueba igualdad entre operandos	a == b → False
!=	Comprueba si dos operandos son distintos	a != b → True
<>	Comprueba si dos operandos son distintos	a <> b → True (sólo v2)
>	Comprueba si un operando es mayor	a > b → False
<	Comprueba si un operando es menor	a < b → True
>=	Comprueba si un operando es mayor o igual	a >= b → False
<=	Comprueba si un operando es menor o igual	a <= b → True

a= 10, b= 20

Nota.- en Python son válidas expresiones del tipo → 3 < x < 5

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

Introducción a Python. Operadores Lógicos.

Operador	Description	Ejemplo
and	AND lógico	True and True → True
or	OR lógico	True or False → True
not	Invierte valor variable booleana	Not (True) → False
&	AND Binario	(a & b) → 0 (0000 0000)
|	OR Binario	(a | b) → 30 (0001 1110)
^	XOR Binario	(a ^ b) → 30 (0001 1110)

a= 10 → 00001010
b= 20 → 00010100

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

Introducción a Python. Operadores Binarios.

Operador	Description	Ejemplo
~	Invierte bits	(~a) → -5 (1111 1101)
<<	Desplazamiento a la izquierda binario	a << 2 → 16 (0001 0000)
>>	Desplazamiento a la derecha binario	a >> 2 → 1 (0000 0001)

a= 4 → 00000100

Introducción a Python. Operadores sobre objetos.

Operador	Description	Ejemplo
in	Búsqueda de variable en una secuencia	x in y → 1 si x es un miembro de la sequencia y
not in	Búsqueda negativa de variable en una secuencia	x not in y → 1 si x no es un miembro de la sequencia y
is	Evalúa si las variables de ambos lados apuntan al mismo objeto	x is y → 1 si id(x) == id(y)
is not	Evalúa si las variables de ambos lados no apuntan al mismo objeto	x is not y → 1 si id(x) != id(y)

Informática para Científicos

Introducción al Lenguaje Python

Prof Pablo García Tahoces

Dpto Electrónica y Computación

ETSE. USC.

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA