

Python.

Conceptos esenciales.

Objetos:

En python, todo es un objeto, estos objetos pueden ser de un valor simple como números, cadenas y booleanos o de múltiples valores (contenedores) como tuplas, listas, diccionarios.

Cada objeto posee un nombre, tipo y ID. El tipo y el ID no cambian durante la vida del objeto, el ID de un objeto es un valor único.

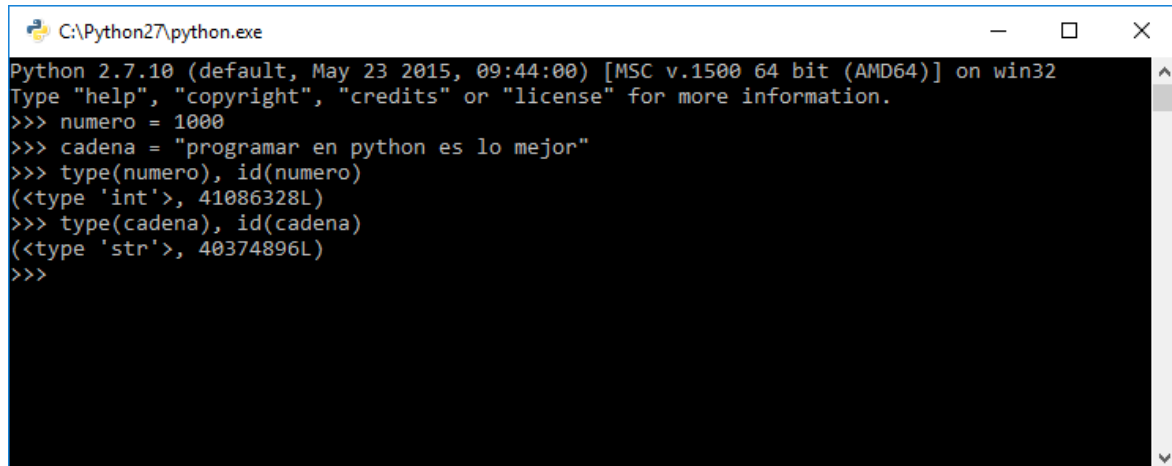
Consideraciones al establecer nombre a los objetos:

- Los nombres de los objetos deben tener solo caracteres alfanuméricos y guion bajo (A-Z, a-z, 0-9, _).
- Los nombres no deben de iniciar con un dígito (0-9).

En python no es necesario definir un tipo de dato para un objeto nuevo, este se establece al asignarle un valor.

Se puede utilizar las funciones `type()` y `id()` para obtener el correspondiente tipo y id del objeto.

Usando la línea de comandos de python :



```
C:\Python27\python.exe
Python 2.7.10 (default, May 23 2015, 09:44:00) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> numero = 1000
>>> cadena = "programar en python es lo mejor"
>>> type(numero), id(numero)
(<type 'int'>, 41086328L)
>>> type(cadena), id(cadena)
(<type 'str'>, 40374896L)
>>>
```

Tipos de objetos:

Números:

En python existen 3 tipos de números:

- **Enteros (Integers).**
Tipo de dato: int , ejemplo: 100, 999, 123322.
- **Numeros con punto flotante (Float).**
Tipo de dato: float, ejemplo: 1.1, 0.5, 100.23.
- **Numeros complejos (Complex).**
Tipo de dato: complex, 10+20j, 4x+8.
Creando objeto de numero complejo: c = complex(10+20j)

Booleanos:

Los números booleanos poseen 2 estados falso (False) y verdadero (True).

Tipo de dato: bool.

Nota: Los valores de los booleanos siempre deben de ir con la primera letra en mayúscula.

Cadenas:

Las cadenas (String) son una secuencia de caracteres, estas pueden ser definidas con comillas simples, dobles y triples(inmutables). Ejemplo: 'hola', "aaaaaa", """hola mundo""".

Tipo de dato: str

Además se les pueden aplicar directamente operadores como: Concatenar (+), repetir cadena (*).

Contenedores:

- **Tuplas:** Una tupla es una lista inmutable. Una tupla no puede modificarse de ningún modo después de su creación.
Tipo de dato: tuple
 - T = (1, 2, 3, 4)
- **Listas:** Una lista en Python es muy parecida a un array en Java.
Tipo de dato: list
 - L = [10, 20, "hola ", "mundo"]
- **Conjuntos:**
Tipo de dato: set
 - S = { 10, 20, 50}
- **Diccionarios:** Los diccionarios en Python son un tipo de estructuras de datos que permite guardar un conjunto no ordenado de pares clave-valor.
Tipo de dato: dict
 - D = { "auto": 4, "moto": 2, "bote": 0 }

Mutabilidad de objetos:

En la programación un objeto inmutable es aquel cuyo estado no puede ser modificado una vez creado, solo podremos acceder a lo que contiene pero no podremos modificar su contenido.

Objetos mutables: Listas, Diccionarios.

Objetos inmutables: Tuplas, números, conjuntos, cadenas.

Esto se comprueba al hacer un cambio en un objeto inmutable este se convierte en un objeto nuevo, para verificarlo se puede usar la función `id()`.

Operadores aritméticos:

Operación	Operador
Suma	+
Resta	-
Multipliación	*
División	/
Potencia	**
División redondeada	//
Modulo	%

Operadores de comparación:

Operación	Operador
Menor que	<
Mayor que	>
Menor o igual que	<=
Mayor o igual que	>=
Igual	==
Diferente a	!=
Equivalente(valor y id)	is
Existe en	in

Operadores de asignación:

- += , -= , *= , /=
- **= , //= , %=

Jerarquía de operadores:

Descripción	Operador	Nivel	Asociatividad
Lógico OR	or	1	Izquierda a derecha
Lógico AND	and	2	Izquierda a derecha
Lógico NOT	not	3	Izquierda a derecha
Comparación	<, <=, >, >=, ==, !=, is, in	4	Izquierda a derecha
Suma, Resta	+, -	5	Izquierda a derecha
Multiplicación, división	*, /, //, %	6	Izquierda a derecha
Exponente	**	7	Derecha a izquierda
Paréntesis	()	8	Izquierda a derecha

Palabras reservadas en la versión 2.x:

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

Indentación:

Una de las características de python es la Indentación estricta que maneja, este es un concepto clave para iniciar a aprender sobre el lenguaje.

La Indentación es escribir bloques de texto más a la derecha que los anteriores en función a su relación, la Indentación se puede realizar añadiendo espacios o tabulaciones. En python una mala Indentación puede provocar que errores en partes del código o que este no se ejecute, algunos programadores usan esta técnica para tener un código más legible y ordenado, en python es estricto el uso de la Indentación.

Estructuras condicionales:

Bucles condicionales:

- If, else, elif
- While
- For

Control:

- Continue
- Break
- Pass

Función rango:

- Range(inicio)
- Range(inicio, final)
- Range(inicio, final, pasos)