

Tipos de datos Python

String, Las cadenas (o strings) son un tipo de datos compuestos por secuencias de características que representan texto. Estas cadenas son de tipo "str" y se delimitan mediante el uso de comillas simples o dobles.

En el caso que queramos usar comillas (o un apóstrofo) dentro de una cadena tenemos distintas opciones. La más simple es encerrar nuestra cadena mediante un tipo de comillas (simples o dobles) y usar el otro tipo dentro de la cadena.

Metodos String

Capitalize() Convierte el primer caracter en mayuscula

Casefold() Convierte la string en minuscula

endswith() Devuelve true si la string termina con un valor en especifico

find() Busca en la string por un valor en especifico y devuelve la posición donde lo encontró

replace() Devuelve una string donde un valor en especifico es remplazado

split() Parte la string con un separador en especifico y devuelve una lista

isnumeric() Devuelve true si todos los caracteres de la string son numericos

isalnum() Devuelve true si todos los caracteres son alphanumericos

swapcase() Convierte mayusculas en minusculas y viceversa

Int Los tipos enteros o int. en Python permiten almacenar un valor numérico no decimal ya sea positivo o negativo de cualquier valor. Una gran ventaja de Python es que no hay que preocuparse por el valor máximo de un int ya que por debajo se encarga de asignar más o menos memoria al número, y podemos representar prácticamente cualquier número.

Metodos int

bit_length() Retorna el número de bits necesarios para representar un número entero, excluyendo el bit de signo y los ceros a la izquierda.

bit_count() Retorna el número de unos en la representación binaria del valor absoluto del entero. Esto se le conoce como recuento de la población.

to_bytes() Retorna un array de bytes que representan un número entero.

from_bytes() Retorna el número entero representado por el vector de bytes.

as_integer_ratio() Retorna una pareja de números enteros cuya proporción es igual a la del número entero original, y con un denominador positivo.

float. El tipo numérico float permite representar un número positivo o negativo con decimales, es decir, números reales. En Python las cosas son algo distintas, y los float son en realidad double.

Métodos float

as_integer_ratio() Retorna una pareja de números enteros cuya proporción es exactamente igual que la del valor en punto flotante original, con un denominador positivo. Si se llama con valores infinitos lanza una excepción de tipo `OverflowError` y si se llama con NaN eleva una excepción de tipo `ValueError`.

is_integer() Retorna true si el valor en coma flotante se puede representar sin pérdida con un número entero, y false si no se puede.

hex() Retorna la representación de un valor en coma flotante en forma de cadena de texto en hexadecimal. Para números finitos, la representación siempre empieza con el prefijo `0x`, y con una `p` justo antes del exponente.

fromhex(s) Método de clase que retorna el valor en coma flotante representado por la cadena de caracteres en hexadecimal en `s`. La cadena `s` puede tener espacios en blanco al principio o al final.

Bool. Una Variable booleana es una Variable que sólo puede tomar dos posibles valores: True (verdadero) o False (falso)

En Python cualquier Variable (en general, cualquier objeto) puede considerarse como una variable booleana. En general los elementos nulos o vacíos se consideran False y el resto se consideran True

Métodos bool

Bool() Devuelve si un elemento se considera true o false