INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO

Maestría en Ciencia de Datos

Minería de Datos

Tarea 01: Tipos de datos de Python

Profesor:

Dr. Ángel Fernando Kuri Morales

Alumna:

Gabriela Flores Bracamontes

Clave única:

160124

México, D.F. 24 de agosto de 2015.

1. Tipos de datos enteros en Python

1.1. Enteros

En Python se pueden representar mediante el tipo int (de integer, entero) o el tipo long (largo). La única diferencia es que el tipo long permite almacenar números más grandes. Es aconsejable no utilizar el tipo long a menos que sea necesario, para no malgastar memoria.

El tipo int de Python se implementa a bajo nivel mediante un tipo long de C. Y dado que Python utiliza C por debajo, como C, y a diferencia de Java, el rango de los valores que puede representar depende de la plataforma.

En la mayor parte de las máquinas el long de C se almacena utilizando 32 bits, es decir, mediante el uso de una variable de tipo int de Python podemos almacenar números de -231 a 231 – 1, o lo que es lo mismo, de -2.147.483.648 a 2.147.483.647.

En plataformas de 64 bits, el rango es de -9.223.372.036.854.775.808 hasta 9.223.372.036.854.775.807.

El tipo long de Python permite almacenar números de cualquier precisión, limitado por la memoria disponible en la máquina.

Al asignar un número a una variable esta pasará a tener tipo int, a menos que el número sea tan grande como para requerir el uso del tipo long.

```
# type(entero) daría int

entero = 23
```

También podemos indicar a Python que un número se almacene usando long añadiendo una L al final:

```
# type(entero) daría long

entero = 23L
```

1.2. Reales

Los números reales son los que tienen decimales. En Python se expresan mediante el tipo float. En otros lenguajes de programación, como C, tenemos también el tipo double, similar a float pero de mayor precisión (double = doble precisión). Python, sin embargo, implementa su tipo float a bajo nivel mediante una variable de tipo double de C, es decir, utilizando 64 bits, luego en Python siempre se utiliza doble precisión, y en concreto se sigue el estándar IEEE 754: 1 bit para el signo, 11 para el exponente, y 52 para la mantisa. Esto significa que los valores que podemos representar van desde ±2,2250738585072020 x 10-308 hasta ±1,7976931348623157×10308.

Los valores corresponden a varias constantes definidas en la **libreria float.h** para C.

sys.float_info(

max=1.7976931348623157e+308, max_exp=1024, max_10_exp=308,

min=2.2250738585072014e-308, min_exp=-1021, min_10_exp=-307,

dig=15, mant_dig=53, epsilon=2.220446049250313e-16, radix=2, rounds=1)

La mayor parte de los lenguajes de programación siguen el mismo esquema para la representación interna. Pero como muchos sabréis esta tiene sus limitaciones, impuestas por el hardware. Por eso desde Python 2.4 contamos también con un nuevo tipo *Decimal*, para el caso de que se necesite representar fracciones de forma más precisa.

Para representar un número real en Python se escribe primero la parte entera, seguido de un punto y por último la parte decimal.

real = 0.2703

También se puede utilizar notación científica, y añadir una e (de exponente) para indicar un exponente en base 10. Por ejemplo:

real = 0.1e-3

sería equivalente a 0.1 x 10-3 = 0.1 x 0.001 = 0.0001

1.3. Complejos

En el caso de que necesitéis utilizar números complejos, o simplemente tengáis curiosidad, os diré que este tipo, llamado complex en Python, también se almacena usando coma flotante, debido a que estos números son una extensión de los números reales.

En concreto se almacena en una estructura de C, compuesta por dos variables de tipo double, sirviendo una de ellas para almacenar la parte real y la otra para la parte imaginaria.

Las cadenas no son más que texto encerrado entre comillas simples (‘cadena’) o dobles (“cadena”). Dentro de las comillas se pueden añadir caracteres especiales escapándolos con ‘\’, como ‘\n’, el carácter de nueva línea, o ‘\t’, el de tabulación.

1.4. Cadenas

Una cadena puede estar precedida por el carácter ‘u’ o el carácter ‘r’, los cuales indican, respectivamente, que se trata de una cadena que utiliza codificación Unicode y una cadena raw (del inglés, cruda). Las cadenas raw se distinguen de las normales en que los caracteres escapados mediante la barra invertida (\) no se sustituyen por sus contrapartidas. Esto es especialmente útil, por ejemplo, para las expresiones regulares.