**SISTEMAS NUMERICOS**

Los sistemas numéricos son: binario, octal y hexadecimal, además del decimal.

Para escribir cadenas “sobre estas bases” se usan los prefijos 0b, 0o, 0x:

binario = 0b1000000000000

octal = 0o12

hexa = 0xffffff

Tambien pueden declararse los números sobre cierta base, indicando la misma como segundo parámetro, instanciando la clase **int()**

b = int('101', 2)

(podría incluirse cualquier base, no solo 2,10,8,16)

**TIPO FLOAT**

Cuando se trabaja con datos de tipo float, puedo indicar con cuantos decimales “trabajo”

a = 3.1  
print(f'{a:.5f}')

Estas líneas dan como result: 3.10000

Además puedo trabajar con notación científica:

#notacion cientifica  
anio = 9.64e11  
print(f'{anio}')  
pennis = 1e-5  
print(f'{pennis:.5f}')

**MANEJANDO VALORES INFINITOS**

Para crear valores infinitos existen varias formas. Una es mediante el constructor de la clase float:

infinito\_positivo = float('inf')  
infinito\_negativo = float('-inf')

para comprobar que un numero sea infinito, puede incorporarse el modulo **math** y ejecutar de la siguiente manera:

print(math.isinf(infinito\_positivo))  
print(math.isinf(infinito\_negativo))

Tambien podría “crear” estos valores con el modulo math:

infinito\_positivo = math.inf  
infinito\_negativo = -math.inf

La clase Decimal también nos permite realizar esto:

from decimal import Decimal  
infinito\_positivo = Decimal('Infinity')  
infinito\_negativo = Decimal('-Infinity')

**NOT A NUMBER**

Not a Number es un tipo númerico no definido. Puede “obtenerse” de la siguiente manera:

a = float('NaN')  
b = Decimal('nan')

//las declaraciones no son case sensitive

**BOOL**

El constructor de esta clase puede “convertir” varios tipos de datos, comportándose de distinta forma dependiendo del resultado.

Cuando se convierte un tipo int, devuelve false cuando es =0 y True en los demás casos.

Cuando se convierte un tipo str, devuelve false con la cadena vacia y True en los demás casos.

Con las listas, tuplas y diccionarios, devuelve false cuando están vacíos y True en los demás casos.