

## VARIABLES Y SU REFERENCIACIÓN

### Expresiones matemáticas

Una expresión matemática en un lenguaje de programación requiere de operadores, variables y funciones para indicar la operación de sus componentes y generar un resultado numérico que se almacenará en la memoria a través de la variable referenciada.

Los operadores aritméticos como suma ( $a+b$ ), resta ( $a-b$ ), multiplicación ( $a*b$ ), división real ( $a/b$ ) y elevar a una potencia ( $a**b$ ) están disponibles en todos los lenguajes, y *Python* no es la excepción; sin embargo, existen otros operadores de uso no cotidiano que están disponibles en el lenguaje: división entera ( $a//b$ ), resto de la división entera o módulo ( $a\%b$ ); para modificar la jerarquía de las operaciones (también llamado precedencia, prelación o prioridad) implícita en el lenguaje, sólo se usan los paréntesis () como signos de agrupación.

La precedencia de las operaciones numéricas se muestra en la tabla no. 1.

Tabla 1. Orden de precedencia de las operaciones en Python.

Precedencia	Operación	Resultado
1	-x	Cambio de signo
1	+x	Operador identidad (no hace nada)
2	x+y	Suma x e y
2	x-y	Resta de y a x
3	x*y	Multiplca x por y
3	x/y	División real
3	x//y	División entera
3	x%y	Resto de la división entera de x/y
4	x**y=pow(x,y)	Elevación de x a la y
5	(expresión)	Paréntesis: fuerza el orden de ejecución

Se recomienda poner solo los paréntesis necesarios en las expresiones, pero en operaciones complejas, los paréntesis aumentan legibilidad.

Tanto el conjunto de los números enteros como el de los reales, (float), en matemáticas pueden ir desde  $-\infty$  a  $+\infty$ . Además, mientras que para cualquier entero existe un anterior y un siguiente, entre dos reales cualesquiera existen infinitos números reales. La implementación completa del tipo float en Python depende del sistema operativo sobre el que trabaja el Shell. Generalmente está representado en 8 bytes, con un rango de valores que va desde  $1.7 \times 10^{-308}$  a  $1.7 \times 10^{+308}$ , para un sistema operativo Windows de 64 bits. En caso de presentarse un desbordamiento, el lenguaje mostrará una salida *inf* (*infinite*). La representación de un flotante en formato de notación científica sustituye el «x10<sup>x</sup>» por «e<sup>x</sup>». Algunos ejemplos se muestran en la figura 4 ejecutados en el Shell.

```

>>> 23.7+51*9.81
524.01
>>> 1+45*3
136
>>> 1/(3+4*7)
0.03225806451612903
>>> (8+30)/2-2/(8+30)
18.94736842105263
>>> 1*(1/(3+4*(5-1)))
0.05263157894736842
>>> 3.1415927/2
1.57079635
>>> math.pi/2
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    math.pi/2
NameError: name 'math' is not defined
>>> import math
>>> math.pi/2
1.5707963267948966
>>> round(math.pi/2,3)
1.571
>>> int(math.pi)
3
>>> pi=math.pi
>>> pi/4
0.7853981633974483
>>> round(pi+0.5)
4
>>> round(pi,4)
3.1416
>>> round(pi/32)
0
>>> round(pi/2,32)
1.5707963267948966
>>> 2**120
1329227995784915872903807060280344576
>>> 2.5**45
8.07793566946316e+17
>>> 1.5e-9
1.5e-09
>>> 0.000000000000000023
2.3e-16

```

*Figura 4. Ejemplos de operaciones en el Shell.*

En una de las entradas (la séptima) puede observarse que al pedirse el resultado de  $\pi/2$ , donde  $\pi=3.141592657$ , y se invoca mediante la biblioteca (*library*) *math*, se presenta un error al no haberse invocado primero usando la instrucción *import*, misma que más adelante la estudiaremos.