



Ciclo 1

Semana 2

Algoritmos y lenguaje Python

Lectura 4 – Expresiones



| Expresiones

En Python se tienen diferentes operadores: aritméticos, lógicos y de comparación o relacionales, y tienen diferente precedencia en su ejecución cuando se combinan unos con otros. Ver figura 1.

Operación	Operador	Aridad	Asociatividad	Precedencia
Exponenciación	**	Binario	Por la derecha	1
Identidad	+	Unario	—	2
Cambio de signo	-	Unario	—	2
Multiplicación	*	Binario	Por la izquierda	3
División	/	Binario	Por la izquierda	3
Módulo (o resto)	%	Binario	Por la izquierda	3
Suma	+	Binario	Por la izquierda	4
Resta	-	Binario	Por la izquierda	4
Igual que	==	Binario	—	5
Distinto de	!=	Binario	—	5
Menor que	<	Binario	—	5
Menor o igual que	<=	Binario	—	5
Mayor que	>	Binario	—	5
Mayor o igual que	>=	Binario	—	5
Negación	not	Unario	—	6
Conjunción	and	Binario	Por la izquierda	7
Disyunción	or	Binario	Por la izquierda	8

Figura 1: Tomado de Marzal y Gracia.



Semana 2

Algoritmos y lenguaje Python

Vale la pena recordar las tablas de verdad de los operadores lógicos and y or:

and			or		
operandos		resultado	operandos		resultado
izquierdo	derecho		izquierdo	derecho	
True	True	True	True	True	True
True	False	False	True	False	True
False	True	False	False	True	True
False	False	False	False	False	False

Figura 2: Tomado de Marzal y Gracia.

En las expresiones también se puede tener funciones predefinidas como, por ejemplo:

- **abs(-3)**: calcula el valor absoluto del argumento dado.
- **float(4)**: si recibe un número entero como argumento, lo retorna convertido en un número de punto flotante.
- **int(-2.9)**: si recibe un número de punto flotante como argumento, lo retorna convertido en un número entero eliminando la parte fraccionaria.
- **str(2.1)**: si recibe un número como argumento, devuelve una representación de éste como cadena de caracteres.
- **round(-2.9)**: redondea al entero más cercano.