

TABELA 1.1. Operatory stosowane w wyrażeniach logicznych w języku Python

SYMBOL	ZNACZENIE	PRZYKŁAD	WARTOŚĆ
==	Równość (porównywanie elementów wskazywanych przez zmienne lub złożone wyrażenia logiczne po obu stronach tego operatora, np. $a+3==x$). Nie pomył operatora == z przypisaniem (pojedynczy znak =, np. $x=5$, nie jest wyrażeniem logicznym!).	$x=5$	(przypisanie)
		$x==5$	True
!=	Nierówność (np. $5!=0$ zwróci True).	$x!=5$	False
		$x-5!=1$	True
or	Lub. Zwraca True, gdy co najmniej jeden człon zdania logicznego także zwraca wartość True.	$x=5, y=6$	(przypisanie)
		$x==5$ or $x!=6$	True
		$x==6$ or $x==y$	False
		$x<1$ or $x>y$	False
and	I. Zwraca True, gdy wszystkie porównywane człony zdania logicznego także zwracają wartość True.	$x>1$ or $y>x$	True
		$2<1$ and $5<6$	False
		$2>1$ and $5<6$	True
not	Zaprzeczenie. Zmienia wartość wyrażenia logicznego na przeciwną.	not $(5>0)$	False
>	Większy.	$2>1$	True
>=	Większy lub równy.	$2>=1$	True
<	Mniejszy.	$2<1$	False
<=	Mniejszy lub równy.	$1<=1$	True
is	To słowo kluczowe Pythona pozwala na sprawdzenie, czy zmienne wskazują na ten sam obiekt.		
in	Pozwala na sprawdzenie, czy obiekt znajduje się w pewnej kolekcji.		



Poprawne instrukcje logiczne to takie, które zwracają wartości True lub False. Python nie pozwoli na niejawne konwersje, np. wyrażenie $b>(a=1)$ zwróci błąd, gdyż użyto pojedynczego znaku = (przypisanie) zamiast == (sprawdzanie równości). Przykładem poprawnego wyrażenia logicznego jest: $(x>b)$ and $(b<=100)$; może to być warunek sterujący przebiegiem pętli for lub while lub wyrażenie zawarte w konstrukcji if.