

## Operadores

Operadores são elementos de programação utilizados para combinar valores, variáveis e expressões em uma expressão ou instrução mais complexa. A tabela abaixo mostra os operadores que o IDL tem suporte, sua precedência, e mostra alguns exemplos de uso destes operadores.

Prioridade	Operadores	Descrição	Exemplos
Primeiro	( )	Parênteses para agrupar.	IDL> print, 3*2+1, 3*(2+1) 7 9
	[ ]	Colchetes para concatenar matrizes ou vetores.	IDL> a = indgen (3) IDL> print, [a, -17] 0 1 2 -17
Segundo	( )	Parênteses em uma chamada de função.	IDL> print, indgen (3)*2 0 2 4
	[ ]	Colchetes para sobrescrever matrizes ou vetores.	IDL> a = indgen (5) IDL> print, a[3]*2 6
	.	Referência a estruturas.	IDL> s = {val :5} IDL> print, s.val*2 10
Terceiro	*	Referência a ponteiros.	IDL> p = ptr_new(1) IDL> print, *p 1
	++	Incrementar.	IDL> a = 5 IDL> print, a++, a, ++a 5 7 7
	--	Decrementar.	IDL> print, a--, a, --a 7 5 5
	^	Exponenciação.	IDL> print, a^2, a^(-2) 25.0000 0.0400000
Quarto	# e ##	Multiplicação de matrizes.	(veja logo a frente "Multiplicação de Matrizes")
	*	Multiplicação.	IDL> print, 5*2 10
	/	Divisão.	IDL> print, 5/2, 5/2. 2 2.50000
	mod	Módulo.	IDL> print, 5mod2 1
Quinto	+	Adição / Concatenação de strings.	IDL> print, 5+2 7 IDL> print, 'Dave'+ ' Stern' Dave Stern
	-	Subtração / Negação unária.	IDL> print, 5-2, -5 3 -5
	<	Mínimo.	IDL> print, 5 < 2 2
	>	Máximo.	IDL> print, 5 > 2 5
	not	Complemento Bit a Bit.	DL> print, not 5 -6
Sexto	eq	Igual á.	IDL> print, 5eq2, 2.0eq2 0 1
	ne	Diferente de.	IDL> print, 5ne2, 2.0ne2 1 0
	le	Igual ou menor que:	IDL> print, 5 le 2, 2.0 le 2 0 1
	lt	Menor que:	IDL> print, 5 lt 2, 2.0 lt 2 0 0

	ge	Maior ou igual que:	IDL> print, 5 ge 2, 2.0ge2 1 1
	gt	Maior que:	IDL> print, 5 gt 2, 2.0gt2 1 0
Sétimo	and	E Bit a Bit.	IDL> print, 5 and 6 4
	or	OU Bit a Bit.	IDL> print, 5 or 6 7
	xor	XOR Bit a Bit.	IDL> print, 5 xor 6 3
Oitavo	&&	E Lógico.	IDL> print, 5 && 6 1
		OU Lógico.	IDL> print, 5    6 1
	~	Negação Lógica.	IDL> print, ~5 0

Quando operadores do mesmo nível de precedência são usados sem serem agrupados em uma expressão, eles são avaliados da esquerda para direita. Por exemplo:

```
IDL> print, 5 mod 2 * 7
7
```

## Operação de Matrizes ou Vetores

Os operadores do IDL podem ser usados com matrizes ou vetores da mesma forma que são usados na escalar.

Por exemplo:

```
IDL> print, indgen(5) mod 2
0 1 0 1 0
```

Aqui, o operador modulo (MOD) é aplicado em cada elemento retornado da função *INDGEN*.

```
IDL> print, indgen(5) > 2
2 2 2 3 4
IDL> print, indgen(5) gt 2
0 0 0 1 1
```

Neste caso, os operadores são aplicados em cada elemento do vetor.

## Operadores Compostos

O IDL começou a ter suporte a estes operadores a partir da versão 6.0 . Os operadores válidos estão listados na tabela abaixo.

*=	<=	eq=	mod=
##=	^=	ge=	#=
xor=	gt=	-=	or=
le=	+=	and=	lt=
/=	>=	ne=	

Estes operadores compostos podem ser combinados com outros operadores. Uma instrução como essa

$a \text{ op } = \text{expressão}$

é equivalente á

$a = \text{temporário} (a) \text{ op } (\text{expressão})$

onde *op* é um operador do IDL podendo ser combinado para formar um dos operadores compostos da lista acima, e *expressão* é qualquer expressão do IDL, o resultado será o mesmo que da primeira instrução.

Por exemplo:

```
IDL> a = 3
IDL> a += 5 * !pi
IDL> print, a
      18.7080
```

O valor final é o valor inicial mais 5 vezes  $\pi$ .