

Operadores em C e C++ e *Espaço Digital*

Origem: Wikipédia, a enciclopédia livre.

[Saltar para a navegação](#)[Saltar para a pesquisa](#)

C++ também contém os operadores para conversão de tipos de dados `const_cast`, `static_cast`, `dynamic_cast` e `reinterpret_cast`, que não estão listados na tabela.



Índice

- 1Precedência de operadores
- 2Tabela
 - 2.1Operadores aritméticos
 - 2.2Operadores comparativos
 - 2.3Operadores lógicos sobre bits
 - 2.4Outros operadores
- 3Sinônimos em C++

Precedência de operadores

A seguir é listada a ordem de precedência e associatividade dos operadores. Elementos na mesma linha são calculados com a mesma precedência, de acordo com a direção dada.

A sintaxe das expressões é especificada por uma gramática livre de contexto, a tabela a seguir é inferida pela gramática.

Operadores	Descrição	<u>Associatividade</u>
::	Resolução de escopo	esquerda para direita
++ -- () [] . ->	Incremento e decremento pós-fixo Parênteses (chamada de função) Elemento de arranjo Seleção de elemento por identificador Seleção de elemento por ponteiro	esquerda para direita
++ -- + - ! ~ (tipo) * & sizeof new [] delete []	Incremento e decremento prefixo Adição e subtração unária Não lógico e complemento Conversão de tipo de dado Desreferência Referência (endereço de elemento) tamanho de elemento Alocação dinâmica de memória Desalocação dinâmica de memória	direita para esquerda
.* ->*	Ponteiro para membro	esquerda para direita

* / %	Multiplicação, divisão, e módulo (resto)	
+ -	Adição e subtração	
<< >>	Deslocamento de bits à esquerda e à direita	
< <=	“menor que” e “menor ou igual que”	
> >=	“maior que” e “maior ou igual que”	
== !=	“Igual a” e “diferente de”	
&	E para bits	
^	Ou exclusivo para bits	
	Ou para bits	
&&	E lógico	
	Ou lógico	
c?t:f	<u>Condição ternária</u>	direita para esquerda
=	Atribuição	
+= -=	Atribuição por adição ou subtração	
*= /= %=	Atribuição por multiplicação, divisão ou módulo (resto)	
<<= >>=	Atribuição por deslocamento de bits	
&= ^= =	Atribuição por operações lógicas	
throw	Lançamento de <u>exceção</u>	---
,	Vírgula	esquerda para direita

Tabela

Operadores aritméticos			
Operador	Sintaxe	É sobrecarregável?	Presente em <u>C</u> ?
Adição unária	+a	✓	✓
Adição	a + b	✓	✓
Incremento pré-fixado	++a	✓	✓
Incremento pós-fixado	a++	✓	✓
Atribuição por adição	a += b	✓	✓
Subtração unária	-a	✓	✓
Subtração	a - b	✓	✓
Decremento pré-fixado	--a	✓	✓
Decremento pós-fixado	a--	✓	✓
Atribuição por subtração	a -= b	✓	✓
Multiplicação	a * b	✓	✓
Atribuição por multiplicação	a *= b	✓	✓
Divisão	a / b	✓	✓
Atribuição por divisão	a /= b	✓	✓
Módulo (resto)	a % b	✓	✓

Atribuição por módulo (resto)	a % b	✓	✓
Operadores comparativos			
Operador	Sintaxe	É sobrecarregável?	Presente em <u>C</u> ?
Menor que	a < b	✓	✓
Menor ou igual que	a <= b	✓	✓
Maior que	a > b	✓	✓
Maior ou igual que	a >= b	✓	✓
Diferente de	a != b	✓	✓
Igual a	a == b	✓	✓
Não lógico	!a	✓	✓
E lógico	a && b	✓	✓
Ou lógico	a b	✓	✓
Deslocamento à esquerda	a << b	✓	✓
Atribuição de deslocamento à esquerda	a <<= b	✓	✓
Deslocamento à direita	a >> b	✓	✓
Atribuição de deslocamento à direita	a >>= b	✓	✓
Operadores lógicos sobre bits			
Operador	Sintaxe	É sobrecarregável?	Presente em <u>C</u> ?
Complemento	~a	✓	✓
E	a & b	✓	✓
Atribuição por e	a &= b	✓	✓
Ou	a b	✓	✓
Atribuição por ou	a = b	✓	✓
Ou exclusivo	a ^ b	✓	✓
Atribuição por ou exclusivo	a ^= b	✓	✓
Outros operadores			
Operador	Sintaxe	É sobrecarregável?	Presente em <u>C</u> ?
Atribuição	a = b	✓	✓
Chamada de função	a ()	✓	✓
Elemento de arranjo	a []	✓	✓
De referência	*a	✓	✓
Referência	&a	✓	✓
Membro de ponteiro	a->b	✓	✓
Membro de identificador	a . b	✗	✓
De referência de membro de	a . *b	✗	✗

identificador			
De-referência de membro de ponteiro	<code>a->*b</code>	✓	✗
Conversão de tipo de dados	<code>(tipo) a</code>	✓	✓
Vírgula	<code>a , b</code>	✓	✓
Condição ternária	<code>a ? b : c</code>	✗	✓
Resolução de escopo	<code>a :: b</code>	✗	✗
Tamanho de	<code>sizeof a</code>	✗	✓
Identificador de tipo	<code>typeid (<u>tipo de dado</u>)</code>	✗	✗

Sinônimos em C++

C++ define palavras-chave que atuam como apelidos para alguns operadores: `and` (`and`), `bitand` (`and`), `and_eq` (`and=`), `or` (`||`), `bit or` (`|`), `or_eq` (`|=`), `xor` (`^`), `xor_eq` (`^=`), `not` (`!`), `not_eq` (`!=`), `compl` (`~`). Eles são processados pelo analisador sintático da mesma forma que seus equivalentes.

Já C fornece na biblioteca padrão o cabeçalho `iso646.h`, que define esses símbolos através de macros.

Categorias:

- C (linguagem de programação)
- C++