

Prioridad de operadores

Prioridad de operadores

Los operadores se muestran en orden decreciente de prioridad de arriba a abajo. Los operadores del mismo grupo tienen la misma prioridad (precedencia) y se ejecutan de izquierda a derecha (asociatividad)

<i>Operador</i>	<i>Tipo</i>	<i>Asociatividad</i>
()	Paréntesis	Dcha-Izda
()	Llamada a función	Dcha-Izda
[]	Subíndice	Dcha-Izda
.	Acceso a miembros de un objeto	Dcha-Izda
++	Prefijo incremento	Dcha-Izda
--	Prefijo decremento	Dcha-Izda
+	Más unitario	Dcha-Izda
-	Menos unitario	Dcha-Izda
!	Negación lógica unitaria	Dcha-Izda
~	Complemento bit a bit unitario	Dcha-Izda
(tipo)	Modelado unitario	Dcha-Izda
new	Creación de objetos	Dcha-Izda
*	Producto	Izda-Dcha
/	División	Izda-Dcha
%	Resto entero	Izda-Dcha
+	Suma	Izda-Dcha
-	Resta	Izda-Dcha
<<	Desplazamiento bit a bit a la izquierda	Dcha-Izda
>>	Desplazamiento bit a bit a la derecha con extensión de signo	Dcha-Izda
>>>	Desplazamiento bit a bit a la derecha rellenando con ceros	Dcha-Izda

<i>Operador</i>	<i>Tipo</i>	<i>Asociatividad</i>
<	Menor que	Izda-Dcha
<=	Menor o igual que	Izda-Dcha
>	Mayor que	Izda-Dcha
>=	Mayor o igual que	Izda-Dcha
instanceof	Verificación tipo de objeto	Izda-Dcha
==	Igualdad	Izda-Dcha
!=	Desigualdad	Izda-Dcha
&	AND bit a bit	Izda-Dcha
^	OR exclusive bit a bit	Izda-Dcha
	OR inclusive bit a bit	Izda-Dcha
&&	AND lógico	Izda-Dcha
	OR lógico	Izda-Dcha
?:	Condicional ternario	Dcha-Izda
=	Asignación	Dcha-Izda
+=	Asignación de suma	Dcha-Izda
-=	Asignación de resta	Dcha-Izda
*=	Asignación de producto	Dcha-Izda
/=	Asignación de división	Dcha-Izda
%=	Asignación de módulo	Dcha-Izda
&=	Asignación AND bit a bit	Dcha-Izda
^=	Asignación OR exclusive bit a bit	Dcha-Izda
=	Asignación or inclusive bit a bit	Dcha-Izda
<<=	Asignación de desplazamiento a izquierda bit a bit	Dcha-Izda
>>=	Desplazamiento derecho bit a bit con asignación de extensión de signo	Dcha-Izda
>>>=	Desplazamiento derecho bit a bit con asignación de extensión a cero	Dcha-Izda

