

C operátorok összefoglaló táblázata

Operátor	Példa	Jelentés/leírás
()	(e)	e, a zárójellezett kifejezés
	fv(a,b)	az fv függvény értéke, függvényhívás
[]	t[3]	hivatkozás a 3 indexű tömbelemre
.	s.tag	hivatkozás a struktúra adattagjára
->	ps->tag	hivatkozás a mutatóval kijelölt struktúra adattagjára, a (*ps).tag kifejezéssel egyenértékű
sizeof	sizeof(t)	a t típus mérete byte-ban
	sizeof e	az e kifejezés mérete byte-ban
(típus)	(t)e	a t típusúvá konvertált e kifejezés
?:	a?e1:e2	az e1 kifejezés, ha a≠0, e2 ha a=0
,	e1, e2	e2, szekvenciális kiértékelés (e1 értékelődik ki először)
Pointer operátorok		
* (unáris)	*a	hivatkozás a p által mutatott objektumra, dereferencia
& (unáris)	&a	az a változó címe, címképző operátor
Aritmetikai operátorok		
- (unáris)	-a	az a negáltja, ellentétes előjelűre változtatás
+	a + b	a és b összege
- (bináris)	a - b	a és b különbsége
* (bináris)	a * b	a és b szorzata
/	a / b	a és b hányadosa
%	a % b	az a/b művelet maradéka
Összehasonlító és logikai operátorok		
<	a < b	1, ha a<b, 0 különben
>	a > b	1, ha a>b, 0 különben
<=	a <= b	1, ha a≤b, 0 különben
>=	a >= b	1, ha a≥b, 0 különben
==	a == b	1, ha a egyenlő b-vel, 0 különben
!=	a != b	1, ha a nem egyenlő b-vel, 0 különben
!	!a	a logikai tagadása
&&	a && b	a és b közötti logikai és kapcsolat
	a b	a és b közötti logikai vagy kapcsolat
Léptető operátorok		
++ (prefix)	++a	az a értéke a növelés után
++ (postfix)	a++	az a értéke a növelés előtt
-- (prefix)	--a	az a értéke a csökkentés után
-- (postfix)	a--	az a értéke a csökkentés előtt
Értékadó operátorok		
=	a = b	a, a felveszi b értékét, „legyen egyenlő”
+=	a += b	a + b, az eredmény a-ba kerül, a = a + b-vel egyenértékű
-=	a -= b	a - b, az eredmény a-ba kerül, a = a - b-vel egyenértékű
*=	a *= b	a * b, az eredmény a-ba kerül, a = a * b-vel egyenértékű
/=	a /= b	a / b, az eredmény a-ba kerül, a = a / b-vel egyenértékű

<code>%=</code>	<code>a %= b</code>	<code>a % b</code> , az eredmény a-ba kerül, <code>a = a % b</code> -vel egyenértékű
<code><<=</code>	<code>a <<= b</code>	<code>a << b</code> , az eredmény a-ba kerül, <code>a = a << b</code> -vel egyenértékű
<code>>>=</code>	<code>a >>= b</code>	<code>a >> b</code> , az eredmény a-ba kerül, <code>a = a >> b</code> -vel egyenértékű
<code>&=</code>	<code>a &= b</code>	<code>a & b</code> , az eredmény a-ba kerül, <code>a = a & b</code> -vel egyenértékű
<code> =</code>	<code>a = b</code>	<code>a b</code> , az eredmény a-ba kerül, <code>a = a b</code> -vel egyenértékű
<code>^=</code>	<code>a ^= b</code>	<code>a ^ b</code> , az eredmény a-ba kerül, <code>a = a ^ b</code> -vel egyenértékű
Bitműveletek		
<code>~</code>	<code>~a</code>	az a bitenkénti negáltja
<code><<</code>	<code>a << b</code>	az a bitjeinek balra tolása b-vel
<code>>></code>	<code>a >> b</code>	az a bitjeinek jobbra tolása b-vel
<code>& (binary)</code>	<code>a & b</code>	a és b bitjei közötti és kapcsolat
<code> </code>	<code>a b</code>	a és b bitjei közötti vagy kapcsolat
<code>^</code>	<code>a ^ b</code>	a és b bitjei közötti kizáró vagy kapcsolat

C operátorok precedenciája és asszociativitása

	Operátor	Asszociativitás
1.	<code>() [] . -></code>	balról jobbra
2.	<code>! ~ - ++ - & * (típus) sizeof</code>	jobbról balra
3.	<code>* / %</code>	balról jobbra
4.	<code>+ -</code>	balról jobbra
5.	<code><< >></code>	balról jobbra
6.	<code>< <= > >=</code>	balról jobbra
7.	<code>== !=</code>	balról jobbra
8.	<code>&</code>	balról jobbra
9.	<code>^</code>	balról jobbra
10.	<code> </code>	balról jobbra
11.	<code>&&</code>	balról jobbra
12.	<code> </code>	balról jobbra
13.	<code>? :</code>	jobbról balra
14.	<code>= += -= *= /= %= <<= >>= &= = ^=</code>	jobbról balra
15.	<code>,</code>	balról jobbra