



ÓBUDAI EGYETEM
NEUMANN JÁNOS INFORMATIKAI KAR

Operátorok

Aritmetikai, logikai, bitszintű, összehasonlító operátorok

Bevezetés

- Operátorok:
 - aritmetikai,
 - logikai,
 - bitszintű,
 - összehasonlító,
 - értékadó.

Aritmetikai operátorok

- Kivonás: -
- Szorzás: *
- Osztás: /
- Egész osztás: /
- Modulo osztás/maradék képzés: %

Logikai operátorok

- ÉS: &&
- VAGY: ||
- NEGÁCIÓ: !

Logikai operátorok – ÉS: &&

a	b	a && b
0	0	0
0	1	0
1	0	0
1	1	1

Logikai operátorok – VAGY: ||

a	b	a b
0	0	0
0	1	1
1	0	1
1	1	1

Logikai operátorok – NEGÁCIÓ: !

a	!a
0	1
1	0

Összehasonlító operátorok

- Kisebb mint: <
- Nagyobb mint: >
- Kisebb vagy egyenlő mint: <=
- Nagyobb vagy egyenlő mint: >=
- Egyenlő: == (Figyelem! Az értékadó operátor „=”, nem összekeverendő!)
- Nem egyenlő: !=

Bitszintű operátorok

- A következő BITSZINTŰ műveletek érhetőek el:
 - shiftelés (biteltolás jobbra vagy balra): \gg , \ll
 - ÉS kapcsolat: $\&$
 - VAGY kapcsolat: $|$
 - 1-es komplement/negáció: \sim
 - XOR (kizáró vagy): \wedge

Operátorok - példa

```
1 void setup() {  
2     // put your setup code here, to run once:  
3     Serial.begin(9600);  
4  
5     unsigned char data = 0b00000001;  
6     Serial.print(data);  
7     Serial.print('\t');  
8     Serial.println(data, BIN);  
9  
10    //balra shiftelés  
11    data = 0b00000001;  
12    Serial.print(data);  
13    Serial.print('\t');  
14    Serial.println((data<<2), BIN);
```

Operátorok - példa

```
16 //jobbba shiftelés
17 data = 0b0001000;
18 Serial.print(data);
19 Serial.print('\t');
20 Serial.println(data>>2, BIN);
21
22 //Bit törlés
23 data = 0b11111111;
24 unsigned char mask = (1 << 3);
25 Serial.println(mask, BIN);
26 mask = ~(1 << 4);
27 Serial.println(mask, BIN);
28 Serial.println(data, BIN);
29 data &= ~(1 << 4);
30 Serial.println(data, BIN);
```

Operátorok - példa

```
16 //jobbira shiftelés
17 data = 0b0001000;
18 Serial.print(data);
19 Serial.print('\t');
20 Serial.println(data>>2, BIN);
21
22 //Bit törlés
23 data = 0b11111111;
24 unsigned char mask = (1 << 3);
25 Serial.println(mask, BIN);
26 mask = ~(1 << 4);
27 Serial.println(mask, BIN);
28 Serial.println(data, BIN);
29 data &= ~(1 << 4);
30 Serial.println(data, BIN);
```

```
32 //Bit beállítás
33 data = 0;
34 mask = (1 << 4);
35 Serial.println(mask, BIN);
36 Serial.println(data, BIN);
37 data |= (1 << 4);
38 Serial.println(data, BIN);
```

Operátorok - példa

```
40 //bit billegtetés
41 data = 0b01000000;
42 Serial.println(data, BIN);
43 unsigned char x_or = (1<<7);
44 Serial.println(x_or, BIN);
45 x_or ^= x_or;
46 Serial.println(x_or, BIN);
47 data ^= x_or;
48 Serial.println(data, BIN);
49 data ^= x_or;
50 Serial.println(data, BIN);
51 data ^= x_or;
52 Serial.println(data, BIN);
53 }
54
55 void loop() {}
```



ÓBUDAI EGYETEM

NEUMANN JÁNOS INFORMATIKAI KAR

Köszönöm a figyelmet!