

Operátory

Operátory se používají k provádění operací na hodnotách (nebo proměnných). Existuje několik druhů a tady si je všechny ukážeme.

Arytmetické

operátor	jméno	popisek
<code>+</code>	součet	sečte 2 čísla
<code>-</code>	rozdíl	odečte 2 čísla
<code>*</code>	součin	vynásobí 2 čísla
<code>/</code>	podíl	vydělí 2 čísla
<code>%</code>	modulus	vrátí zbytek po celočíselném dělení
<code>++</code>	zvětšení	zvýší hodnotu proměnné o 1
<code>--</code>	zmenšení	sníží hodnotu proměnné o 1

Porovnávací

Porovnávací operátory se používají (překvapivě) k porovnávání hodnot, a vrací hodnotu `boolean`

operátor	jméno	popisek
<code>==</code>	rovnost	zda hodnoty se rovnají
<code>!=</code>	nerovnost	zda se hodnoty nerovnají
<code>></code>	větší než	zda je první hodnota větší než druhá
<code><</code>	menší než	zda je první hodnota menší než druhá

Logické

Logické operátory pracují striktně s typem `boolean` a vrací hodnotu `boolean`

operátor	jméno	popisek
<code>&&</code>	logické and	vrací <code>true</code> pokud jsou obě hodnoty <code>true</code>
<code> </code>	logické or	vrací <code>true</code> pokud je aspoň + hodnota <code>true</code>
<code>!</code>	logické větší než	převrací boolean hodnotu (z <code>false</code> na <code>true</code> a naopak)

Bitwise

Bitwise operátory pracují s binární reprezentací čísel, porovnávají každý bit zvlášť, a produkují 3. binární číslo jako výsledek. Bitwise logické operátory fungují tak, že porovnávají hodnoty bitů “nad sebou” na dané pozici.

operátor	jméno	popisek
<code>&</code>	bitwise and	dává <code>1</code> pouze pokud jsou oba bity <code>1</code>
<code> </code>	bitwise or	dává <code>1</code> pokud je alespoň 1 bit <code>1</code>
<code>^</code>	xor	dává <code>1</code> pokud je právě jeden z bitů <code>1</code>
<code>~</code>	bitwise not	převrací hodnoty všech bitů v čísle
<code><<</code>	left shift	posune bity v čísle o 1 pozici doprava
<code>>></code>	right shift	posune bity v čísle o 1 pozici doleva

Příklad



```
10111000 & 01110001
```

se napíše jako:

```
10111001  
01110001
```

a pouze tam, kde jsou nad sebou **1** se do výsledku propíše **1** :

```
00110001
```

Tato operace:

```
10111000 >> 3
```

posune binární číslo o 3 místa doprava, dostáváme tedy:

```
00010111
```

bity na pravo se smažou, a vlevo se přidají nové **0**

Nastavovací

Nastavovací operátory se používají k nastavování hodnot proměnným, a většina z nich je zjednodušením ostatních operací na dané proměnné.

operátor	příklad	zjednoduší
=	x = 5	nic
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
&=	x &= 5	x = x & 5
=	x = 5	x = x 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5