



JavaScript အားအသုံးပြုခြင်း

Bitwise Operator in Javascript

Bitwise operator ဆိုတာကတော့ binary digit သဘောတရားတွေနဲ့ အလုပ်လုပ်တဲ့ operator တွေပါ။ Javascript မှာဆိုရင် bitwise operator တွေဟာ 32-bit binary digit နဲ့သာလျှင် အလုပ်လုပ်ပါတယ်။ စိတ်ဝင်စားဖို့တော့ ကောင်းပါတယ်။ Assembly language တခုခု လေ့လာဖူးရင်တော့ digit တွေအလုပ်လုပ်ပုံကို သိနေမှာပါ။ ဒါကိုမှ Javascript မှာ ဘယ်လို အလုပ်လုပ်သွားလဲ ဆိုတာကို ကျနော်တို့ အခုလေ့လာကြည့်ကြပါမယ်။ Javascript မှာ အသုံးပြုနိုင်တဲ့ bitwise operator တွေကတော့....

Symbols	Operators	Forms	Definition
&	Bitwise AND operator	x & y	True(1) – True(1) → True(1) ကျန်တာ အားလုံး false(0)
	Bitwise OR operator	x y	True(1) တလုံးပါရင် True(1) ကျန်တာ false(0)
^	Exclusive OR operator	x ^ y	တူရင် False(0) ၊ မတူရင် True(1)
~	Bitwise NOT operator	~x	Not ကတော့ အားလုံးကို ပြောင်းပြန်ယူလိုက်တာပါ။
<<	Bitwise Shift Left operator	x << y	Y က shift left တန်ဖိုးပါ။ ထို shift တန်ဖိုး အရေအတွက်တိုင်း x digit တွေရဲ့ နောက်မှာ သုည တွေတိုးပေးပြီး တွက်ထုတ်လိုက်တာပါ။
>>	Bitwise Shift Right operator	x >> y	Y က shift right တန်ဖိုးပါ။ ထို shift တန်ဖိုး အရေအတွက်တိုင်း x digit တွေရဲ့ ရှေ့မှာ သုည တွေတိုးပေးပြီး တွက်ထုတ်လိုက်တာပါ။
>>>	Bitwise Shift Right with Zero Operator	x >>> y	Shift right နဲ့ စင်တူပါတယ်။ သို့သော် ဒီကောင်ကတော့ နောက်ဆုံးမှာ သုည တလုံးတည်း ကျန်တဲ့ အခြေအနေထိ ဖြတ်ချလိုက်ပြီး ထိုဖြတ်ချလိုက်တဲ့ အရေအတွက်အတိုင်း ရှေ့မှာ သုညတွေ ထပ်ဖြည့်ပေးပါတယ်။

သင်္ကေတလေးတွေနဲ့ သဘောတရားကို နားလည်ပြီဆိုရင်တော့ ကျနော်တို့ လေ့ကျင့်ခန်းလေးတခု လောက်နဲ့ လေ့လာကြည့်မယ်။ ဒီ Binary Digit ပိုင်းလေးတွေကိုတော့ ကျနော် ဒီဆိုက်မှာပဲ [80x86 Assembly Language](#) သင်ခန်းစာတွေမှာ ရှင်းပြခဲ့ဖူးပါတယ်။ လေ့လာကြည့်လိုက်အုံးပေါ့ဗျာ။

```

<html>
<head><title>Bitwise Operator</title></head>
<body>
<script type="text/javascript">
var a = 2;           // 32-bit binary digit 2 → 000000000000000000000000000010
var b = 3;           // 32-bit binary digit 3 → 000000000000000000000000000011
var linebreak = "<br>";
/* a → 2 → 0000 0010
   b → 3 → 0000 0011
   0000 0010 → 2 */

```

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။



```

document.write("(a & b) => ");
document.write(a & b);
document.write(linebreak);

/* a → 2 → 0000 0010
   b → 3 → 0000 0011 |
       0000 0011 → 3 */
document.write("(a | b) => ");
document.write(a | b);
document.write(linebreak);

/* a → 2 → 0000 0010
   b → 3 → 0000 0011 ^
       0000 0001 → 1 */
document.write("(a ^ b) => ");
document.write(a ^ b);
document.write(linebreak);

/* a → 2 → 0000 0010
   b → shift left 3 → 0001 0000 → 16 */
document.write("(a << b) => ");
document.write(a << b);
document.write(linebreak);

/* a → 2 → 0000 0010
   b → shift right 3 → 0000 0000 → 0 */
document.write("(a >> b) => ");
document.write(a >> b);
document.write(linebreak);

/* a → 2 → 0000 0010
   b → shift right zero fill 3 → 0000 0000 --> 0 */
document.write("(a >>> b) => ");
document.write(a >>> b);
document.write(linebreak);
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>

```

Result:

```

(a & b) => 2
(a | b) => 3
(a ^ b) => 1
(a << b) => 16
(a >> b) => 0
(a >>> b) => 0

```

Set the variables to different values and different operators and then try...



အလုပ်လုပ်သွားပုံကို နားလည်မရ် ထင်ပါတရ်။ ကျနော်တို့ သတ်မှတ်ထားတဲ့ a,b တန်ဖိုးတွေက Decimal number တွေပါ။ ဒါကိုမှ Bitwise operator ရဲ့ သဘောသဘာဝက binary number အဖြစ် ပြောင်းလဲ တွက်ထုတ်ပါတရ်။ ရလာတဲ့ binary number ကိုမှ decimal number အဖြစ် ပြန်ပြောင်းပြီး ကျနော်တို့ကို ထုတ်ပြတာပါ။ အပေါ်က ကုဒ်ထဲမှာ ကျနော် comment သဘောမျိုးနဲ့ အလုပ်လုပ်သွားပုံတွေကို ရှင်းပြထားပါတရ်။ ကြည့်လိုက်တာနဲ့ နားလည်သွားမရ် ထင်ပါတရ်။ ကျနော်တို့ နောက်တခု ဆက်လေ့လာကြည့်ကြအုံးစို့ဗျာ။

Using Bitwise Assignment Operator in Javascript

ဒီလေ့ကျင့်ခန်းကတော့ bitwise operator တွေကို assignment operator ပုံစံမျိုးနဲ့ အသုံးပြုနိုင်ပုံကို ရှင်းလင်းသွားမှာဖြစ်ပါတရ်။ အသုံးပြုနိုင်တဲ့ operator ပုံစံတွေကတော့ &=, |=, ^=, >>=, <<=, >>>=,..... စတာတွေပဲဖြစ်ပါတရ်။ အောက်မှာ လေ့ကျင့်ခန်းလေး တခု လေ့လာကြည့်မရ်။

```
<html>
<head><title>Bitwise Assignment Operators</title></head>
<body>
<script type="text/javascript">
var linebreak = "<br>";
var a = 2; // 32-bit binary digit 2 → 00000000000000000000000000000010
/* a → 2 → 0000 0010
   3 → 0000 0011 &
   0000 0010 → 2 */
document.write("Value of a → (a &= 3) => ");
document.write(a &= 3);
document.write(linebreak);
var b = 2; // 32-bit binary digit 2 → 00000000000000000000000000000010
/* b → 2 → 0000 0010
   3 → 0000 0011 |
   0000 0011 → 3 */
document.write("Value of b → (b |= 3) => ");
document.write(b |= 3);
document.write(linebreak);
var c = 2; // 32-bit binary digit 2 → 00000000000000000000000000000010
/* c → 2 → 0000 0010
   3 → 0000 0011 ^
   0000 0001 → 1 */
document.write("Value of c → (c ^= 3) => ");
document.write(c ^= 3);
document.write(linebreak);
var d = 2; // 32-bit binary digit 2 → 00000000000000000000000000000010
/* d → 2 → 0000 0010
   shift left 3 → 0001 0000 → 16 */
document.write("Value of d → (d <<= 3) => ");
document.write(d <<= 3);
document.write(linebreak);
```



```

var e = 2; // 32-bit binary digit 2 → 000000000000000000000000000010
/* e → 2 → 0000 0010
   shit right 3 → 0000 0000 → 0 */
document.write("Value of e → (e >>=3) => ");
document.write(e >>=3);
document.write(linebreak);
var f = 2; // 32-bit binary digit 2 → 000000000000000000000000000010
/* f → 2 → 0000 0010
   shit right zero fill 3 → 0000 0000 --> 0 */
document.write("Value of f → (f >>=3) => ");
document.write(f >>=3);
document.write(linebreak);
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>

```

Result:

```

Value of a ==>(a &=3) => 2
Value of b ==>(b |= 3) => 3
Value of c ==>(c ^=3) => 1
Value of d ==>(d <=3) => 16
Value of e ==>(e >>=3) => 0
Value of f ==>(f >>=3) => 0
Set the variables to different values and different operators and then try...

```

ဒီလောက်ဆိုရင်တော့ Javascript မှာ Bitwise operator တွေ အသုံးပြုပုံကို အနည်းနဲ့အများ နားလည်နိုင်ပြီလို့ ထင်ပါတယ်။ ခက်ခက်ခဲခဲတော့ မရှိပါဘူး။ Assignment operator ပုံစံနဲ့ binary number တွေ အလုပ်လုပ်သွားတာပါ။ နားလည်မရ ဆိုရင်တော့ ကျနော်တို့ ရှေ့ဆက်ကြအုံးစို့....။

Miscellaneous Operator in Javascript

Miscellaneous operator ကတော့ conditional operator တခုပါပဲ။ သူ့ရဲ့အလုပ်လုပ်ပုံ ကတော့ ပထမ conditional တခုကိုစစ်မယ်။ ပြီးရင် ဒုတိယ expression တခုကို ဆက်အလုပ်လုပ်ပါတယ်။ သူ့ရဲ့ syntax ပုံစံလေးကတော့...

If Condition true ? Then value X : otherwise value Y

အသုံးပြုပုံလေးကို နားလည်သွားပြီဆိုရင်တော့ ကျနော်တို့ လေ့ကျင့်ခန်းလေးတခုလောက် လေ့လာကြည့်ကြအုံးစို့ဗျာ။

```

<html>
<head><title>Miscellaneous Operator</title></head>
<body>
<script type="text/javascript">
var a = 10;
var b = 20;
var linebreak = "<br>";
document.write("((a > b) ? a : b) => ");
Max = (a > b) ? a : b;

```

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။



```
document.write("Max num is "+Max);
document.write(linebreak);

document.write("((a < b) ? a : b) => ");
Min = (a < b) ? a : b;
document.write("Min num is "+Min);
document.write(linebreak);
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
```

Result:

```
((a > b) ? a : b) => Max num is 20
((a < b) ? a : b) => Min num is 10

Set the variables to different values and different operators and then try...
```

Result ကိုကြည့်လိုက်ရင် miscellaneous operator အကြောင်းကို နားလည်သွားမှာပါ။ အခု ဒီပုစ္ဆာမှာ အကြီးဆုံးကိန်းနဲ့ အငယ်ဆုံးကိန်းကို ရှာပြထားပါတယ်။ ပထမ Max number ရှာတဲ့နေရာမှာ ကြည့်လိုက်ပါ။ Max=(a>b)? a:b; ဆိုတာကိုတွေ့မှာပါ။ (a>b)? ဆိုတာကို အရင်စစ်ပါတယ်။ ဒီတော့ Condition မှားပြီးပေါ့။ ဒီတော့ အနောက် expression ကို ဆက်အလုပ်လုပ်ပါတယ်။ သူ့ရဲ့သဘောတရားက တခုမဟုတ် တခုလို့ ဆိုလိုပါတယ်။ ဒီတော့ a ကကြီးလားဆိုတဲ့ condition မှားပြီး ဆိုရင်တော့ b ကကြီးပြီးပေါ့။ ဒီတော့ b တန်ဖိုးကို အကြီးဆုံးကိန်းအဖြစ် ထုတ်ပြလိုက်တာပါ။ ဒီလိုပုံစံနဲ့ပဲ အငယ်ဆုံးကိန်းကိုလဲ ထုတ်ပြထားပါတယ်။ နားလည်မရလို့ ထင်ပါတယ်။

ဒီနေ့တော့ ဒီလောက်နဲ့ပဲ နားကြအုံးစို့ဗျာ။ အားလုံးက အခြေခံလေးတွေ ဖြစ်လို့ စတင်လေ့လာသူများ အတွက်သာ ရည်ရွယ်ပါတယ်။ နောက်နေ့ အပိုင်းလေးမှာတော့ Javascript မှာ အသုံးပြုနိုင်မရဲ့ operators လေးတွေထဲကမှ typeof operator, string တွေကို operator တွေနဲ့ ပေါင်းဆပ်အသုံးပြုပုံစံတဲ့ အပိုင်းလေးတွေကို ဆွေးနွေးသွားပါ့မယ်။

ဒါကတော့ Javascript အားအသုံးပြုခြင်း အပိုင်း(10)လေးပါ။ အားလုံးက အခြေခံလေး တွေဖြစ်လို့ စတင်လေ့လာသူများ အတွက်သာ ရည်ရွယ်ပါတယ်။ အခုမှ စဖတ်မိတဲ့ မိတ်ဆွေ၊ ညီအစ်ကို၊ မောင်နှမများအနေနဲ့ ရှေ့က ပိုမိုလေးကို အရင် ဖတ်စေချင်ပါတယ်။ ကျနော် မှားယွင်းတင်ပြမှုများ ရှိရင်လဲ နားလည်ပေးကြပေါ့ဗျာ။ ကျနော် ရည်ရွယ်ချက်ကတော့ အခြေခံလေးတွေကို မြန်မာလို ဖတ်ရှုခြင်းအားဖြင့် လျှင်မြန်စွာ နားလည် သွားနိုင်ပြီး ဒီထပ် မြင့်သော တခြားသော နယ်ပယ်များကို ကူးသွားနိုင်အောင်ပါ။ မိတ်ဆွေအပေါင်း လေ့လာခြင်းဖြင့် ကျေနပ်နိုင်ကြပါစေ။

Tay Zar Lin

Koyinmaung007@gmail.com

Programmingknowledge.blogspot.com