



JavaScript အားအသုံးပြုခြင်း

Typeof Operator in Javascript

Typeof operator ကတော့ unary operator အမျိုးအစားဖြစ်ပြီး Javascript မှာ မိမိအသုံးပြုတဲ့ data တွေဟာ ဘာအမျိုးအစားဖြစ်တယ်ဆိုတာကို သိချင်တဲ့အခါ typeof operator ကို အသုံးပြုနိုင်ပါတယ်။ သူ့ရဲ့ syntax ပုံစံလေးကတော့...

```
typeof(variable);
```

အသုံးပြုနိုင်တဲ့ typeof operator အမျိုးအစားလေးတွေကိုလဲ လေ့လာကြည့်လိုက်ပါအုံးဗျာ။

Type	String Returned by typeof
Number	"number"
String	"string"
Boolean	"boolean"
Object	"object"
Function	"function"
Undefined	"undefined"
Null	"object"

Syntax လေးနဲ့ အမျိုးအစားလေးတွေကို သိပြီဆိုရင်တော့ ကျနော်တို့ လေ့ကျင့်ခန်းလေး တခုလောက် လေ့လာကြည့်ကြရအောင်ဗျာ။

```
<html>
<head><title>Typeof Operator</title></head>
<body>
<script type="text/javascript">
var a = 10;
var b = "Tayza";
var c = true;
var d;
document.write("Variable a is " +typeof(a));
document.write("<br>Variable b is " +typeof(b));
document.write("<br>Variable c is " +typeof(c));
document.write("<br>Variable d is " +typeof(d));
</script>
<p>Set the variables to different values and different operators and then try...</p>
</body>
</html>
```

Result:

```
Variable a is number
Variable b is string
Variable c is boolean
Variable d is undefined

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

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။

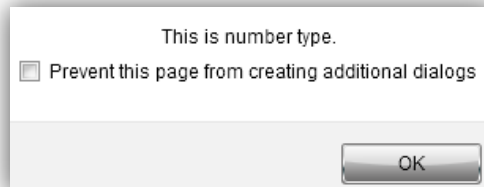
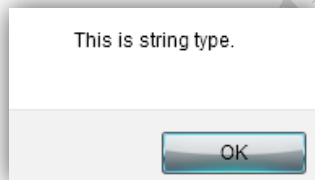


Result ကိုကြည့်လိုက်တာနဲ့တင် typeof() operator ရဲ့ အလုပ်လုပ်နိုင်စွမ်းကို နားလည်သွားပြီ ထင်ပါတယ်။

နောက်တစ်ခုအနေနဲ့ ကျနော်တို့ typeof operator ကို statement တွေနဲ့ တွဲဖတ်အသုံးပြုပြီး function သဘောတရားမျိုးနဲ့ အသုံးပြုနိုင်ပါတယ်။ လေ့လာကြည့်လိုက်ပါအုံးဗျာ။

```
<html>
<head><title>Typeof Operator with function</title></head>
<body>
<script type="text/javascript">
function TypeOfFunction(TOperator)
{
    if (typeof TOperator == "number" )
    {
        window.alert("This is number type.");
    }
    if ( typeof TOperator == "string")
    {
        window.alert("This is string type.");
    }
}
TypeOfFunction("abcd");
TypeOfFunction(1234);
</script>
</body>
</html>
```

Result:



Function သဘောတရားကို နားလည်မရဆီရင်တော့ အရမ်းကို လွန်ကူတဲ့ ပုစ္ဆာလေးပါ။ TypeOfFunction("abcd") နဲ့ TypeOfFunction(1234) စတာတွေကို မြင်လိုက်တာနဲ့ function ဖြစ်တဲ့ TypeOfFunction ထဲကိုရောက်သွားပါပြီ အဲ့ဒီမှာ abcd , 1234 ဖြစ်တဲ့ parameter တွေကို TOperator အဖြစ်သတ်မှတ်လိုက်ပါတယ်။ အဲ့ဒီအခါမှ if statement အတွင်းမှာ typeof operator ကိုအသုံးပြုကာ TOperator ဟာ String ဖြစ်ရင် string message ကိုထုတ်ပေးမယ်။ Number ဖြစ်ရင်တော့ number message ကို ထုတ်ပေးတာပါ။ ဒီလောက်ဆိုရင်ဖြစ် typeof() operator အကြောင်းကို နားလည်လောက်ပြီ ထင်ပါတယ်။ ကျနော်တို့ ရှေ့ဆက်ကြအုံးစို့ဗျာ။



“+” Operator used in string And Addition string and number

ဒါကျတော့ ကျနော်တို့ javascript မှာ operator သင်ခန်းစာတွေရဲ့ နောက်ဆုံး သင်ခန်းစာလေးပေါ့ဗျာ။ အရင်က လေ့ကျင့်ခန်းတွေမှာလဲ ပြောပြခဲ့ဖူးပါတယ်။ ဒါပေမဲ့ သီခြားနားလည်သွားအောင်ပါ။ ပထမဆုံးအနေနဲ့ “+” operator ကိုအသုံးပြုပြီး string တွေကို ပေါင်းဆက်ဖော်ပြနိုင်ကြောင်း အရင်ဆုံး လေ့လာကြည့်ကြမယ်။

```
<html>
<head><title>Using string and number </title></head>
<body>
<script type="text/javascript">
var a="If you add a string and another string, ";
var b="The + operator can also be used to add string variables or text values together.";
var c=a+b;
document.write(c);
</script>
</body>
</html>
```

Result:

If you add a string and another string, The + operator can also be used to add string variables or text values together.

နားလည်မရှိထင်ပါတယ်။ အခု ကျနော်တို့ တဆက်တည်းပဲ “,” operator အသုံးပြုနိုင်လေးပါ တခါတည်း ပြောပြချင်ပါတယ်။ အရင်က လေ့ကျင့်ခန်းတွေမှာ string တွေနဲ့ variable result တွေကြားမှာ “+” operator အသုံးပြုပြီး အဖြေကို ထုတ်ပြခဲ့တာတွေကြီးပါပဲ။ အခု “,” operator အသုံးပြုပုံလေးကို လေ့လာကြည့်လိုက်ကြပါအံ့ဗျာ။

```
<html>
<head><title>Using Comma Operator </title></head>
<body>
<script type="text/javascript">
var a=2;
var b=3;
var c=a+b;
document.write("A= ",a,"<br>B= ",b,"<br>C= ",c);
</script>
</body>
</html>
```

Result:

```
A= 2
B= 3
C= 5
```



အရင်လေ့ကျင့်ခန်းတွေက “+” operator နေရာတွေမှာ “,” operator အစားထိုး အသုံးပြုနိုင်တယ်ဆိုတာ ဒီလေ့ကျင့်ခန်းလေးကြည့်လိုက်ရင် နားလည်သွားမှာပါ။ နှစ်သက်သလို အသုံးပြုနိုင်တာပေါ့ဗျာ။ အခု javascript ရဲ့ operator လေ့ကျင့်ခန်းလေးတွေရဲ့ နောက်ဆုံး လေ့ကျင့်ခန်းလေးဖြစ်တဲ့ number type တွေနဲ့ string type တွေ ပေါင်းဆက် အသုံးပြုနိုင်ပုံကို ပြောပြပါမယ်။ လေ့လာကြည့်လိုက်ပါအုံးဗျာ။

```
<html>
<head><title>Addition string and number</title></head>
<body>
<script type="text/javascript">
var x;
x=15+10;
document.write(x);
document.write("<br />");
x="15"+"10";
document.write(x);
document.write("<br />");
x=15+"10";
document.write(x);
document.write("<br />");
x="15"+10;
document.write(x);
document.write("<br />");
</script>
<p>The rule is: If you add a number and a string, the result will be a string.</p>
</body>
</html>
```

Result:

```
25
1510
1510
1510

The rule is: If you add a number and a string, the result will be a string.
```

အရင်လေ့ကျင့်ခန်းတစ်ခုမှာလဲ ကျနော်ပြောပြခဲ့ဖူးပါတယ်။ String type နဲ့ number type မှာ ပေါင်းဆက် အသုံးပြုနိုင်ပေမဲ့ math သဘောတရားအတိုင်း အလုပ်မလုပ်ဘူးဆိုတာ ဒီလေ့ကျင့်ခန်းလေးကို ကြည့်လိုက်ရင် နားလည်သွားမှာပါ။ ဒီလောက်ဆိုရင်တော့ ကျနော်တို့ Javascript မှာ Operator တွေကို နှစ်သက်သလို အသုံးပြုနိုင်ပြီလို့ ထင်ပါတယ်။ နားမလည်နိုင်သေးရင်တော့ ထပ်ခါတလဲလဲ လေ့လာပေါ့ဗျာ။ ကျနော်ကတော့ အခြေခံလေးတွေကြီး ပြောပြခဲ့လို့ လွန်တန်ဆန်ထင်ရင်တော့ Javascript နဲ့ ပတ်သတ်တဲ့ တခါ tutorial site တွေ အများကြီးပါပဲ။ လေ့လာနိုင်ကြပါစေ။



ဒီနေ့တော့ ဒီလောက်နဲ့ပဲ နားကြအုံးစို့ဗျာ။ အားလုံးက အခြေခံလေးတွေ ဖြစ်လို့ စတင်လေ့လာသူများ အတွက်သာ ရည်ရွယ်ပါတယ်။ နောက်နေ့ အပိုင်းလေးမှာတော့ Javascript မှာ အသုံးပြုနိုင်မရဲ့ statements လေးတွေအကြောင်း ဆွေးနွေးသွားပါမယ်။

ဒါကတော့ Javascript အားအသုံးပြုခြင်း အပိုင်း(11)လေးပါ။ အားလုံးက အခြေခံလေး တွေဖြစ်လို့ စတင်လေ့လာသူများ အတွက်သာ ရည်ရွယ်ပါတယ်။ အခုမှ စဖတ်မိတဲ့ မိတ်ဆွေ၊ ညီအစ်ကို၊ မောင်နှမများအနေနဲ့ ရှေ့က ပိုစ်လေးကို အရင် ဖတ်စေချင်ပါတယ်။ ကျနော် မှားယွင်းတင်ပြမှုများ ရှိရင်လဲ နားလည်ပေးကြပေါ့ဗျာ။ ကျနော် ရည်ရွယ်ချက်ကတော့ အခြေခံလေးတွေကို မြန်မာလို ဖတ်ရှုခြင်းအားဖြင့် လျင်မြန်စွာ နားလည် သွားနိုင်ပြီး ဒီထပ် မြင့်သော တခြားသော နယ်ပယ်များကို ကူးသွားနိုင်အောင်ပါ။ မိတ်ဆွေအပေါင်း လေ့လာခြင်းဖြင့် ကျေနပ်နိုင်ကြပါစေ။

Tay Zar Lin

Koyinmaung007@gmail.com

Programmingknowledge.blogspot.com