



JavaScript အားအသုံးပြုခြင်း

Use FunctionName.arguments to reference the arguments

မိမိ function ထဲ pass လုပ်လာတဲ့ elements အားလုံးကို ခြုံငုံသုံးသတ်ချင်တဲ့ အခါမျိုးမှာ function.arguments လုပ်ငန်းစဉ်ကို အသုံးပြုနိုင်ပါတယ်။

```
<html>
<head><title>Use FunctionName.arguments to reference the arguments</title>
<script type="text/javascript">
function addNums( )
{
    var Answer = 0;
    for (var i = 0; i < addNums.arguments.length; i++)
    {
        var Num = Number(addNums.arguments[i]);
        Answer += Num;
    }
    return Answer;
}
</script>
</head>
<body>
<form Name="Sum">
    <INPUT Type=Text Name="num1">
    <INPUT Type=Text Name="num2">
    <INPUT Type=Text Name="num3">
    <INPUT Type=Text Name="num4">
    <INPUT Type=Button Value="Add Them" onClick='document.write("Sum of my series= "
+addNums(Sum.num1.value, Sum.num2.value, Sum.num3.value, Sum.num4.value));'>
</form>
</body>
</html>
```

Result:

1	2	3	9	Add Them
---	---	---	---	----------

Sum of my series= 15

FunctionName.Arguments အသုံးပြုပုံက Array အသုံးပြုပုံနဲ့ ဆင်တူပါတယ်။ အလုပ်လုပ်သွားပုံ ကတော့ ကျနော်တို့ <form> ထဲမှ text box 4 ခုမှာ ရိုက်ထည့်မရဲ့ တန်ဖိုးတွေကို addNums function ဆီကို ပို့ပါတယ်။ ဒါကိုမှ FunctionName.Argument.length method ကို အသုံးပြုပြီး TextBox 1 to 4 ဆုံးသည်အထိ for loop ကို အသုံးပြုကာ looping ပတ်ပါတယ်။ ဒီမှာက Function Argument ရဲ့ array တန်ဖိုး [i] ကို Num လို့ သတ်မှတ်ပြီး loop တစ်ပတ်တိုင်း ထိုဂင်လာမဲ့ i တန်ဖိုးတိုင်းကို Answer နဲ့

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။



ပေါင်းပေးသွားကာ looping ပြီးဆုံးသည်အထိ အလုပ်လုပ်သွားပါလိမ့်မည်။ နောက်ဆုံးအနေနဲ့တော့ ပေါင်းခြင်း ရလဒ်ဖြစ်တဲ့ Answer တန်ဖိုးကို return ပြန်ပေးလိုက်တာပါပဲ။ နားလည်မရလို့ ထင်ပါတယ်။

Pass an array to a function

ကျနော်တို့ data တွေအများကြီးပါတဲ့ array တစ်ခုကို function ထဲကို pass လုပ်ချင်တဲ့ အခါမျိုးမှာ ဒီနည်းလမ်းကို အသုံးပြုပါတယ်။

```
<html>
<head><title>Pass an array to a function</title>
<script type="text/javascript">
function makeArray( )
{
    var myArray = new Array( );
    myArray[0] =11;
    myArray[1] =8;
    myArray[2] =25;
    myArray[3] =24;
    myArray[4] =102;
    return myArray;
}
function ShowEvenNum(AA)
{
    for (var i = 0; i < AA.length;++i)
    {
        if(AA[i]%2==0)
        {
            even = AA[i] ;
            document.write(even + ",");
        }
    }
    document.write(" are even numbers.");
}
</script>
</head>
<body>
<h4>Pass an array to a function</h4>
<script type="text/javascript">
    var x = makeArray( );
    ShowEvenNum(x);
</script></body>
</html>
```

X → AA

Result:

Pass an array to a function

8, 24, 102, are even numbers.



ဒီပုစ္ဆမှာတော့ ကျနော်တို့ function နှစ်ခုကို အသုံးပြုထားပါတယ်။ MakeArray Function ထဲက data တွေကို ShowEvenNum Function ထဲကို pass လုပ်ပြီး even number တွေကို ဆွဲထုတ်ပြသွားတာပါ။

ရှင်းရမရင်ဆိုတော့ MakeArray function ကရိုးရှင်းပါတယ်။ New Array တခုကြေငြာပြီး ထို array ထဲကို data တွေ ထည့်ထားယုံပါပဲ။ ကျနော်တို့က ဒီ array ထဲမှာ even number ဘယ်နှစ်လုံးပါတယ်ဆိုတာ သိချင်တဲ့အတွက်ကြောင့် ထို MakeArray Function ကြီးတခုလုံးကို x နဲ့ ပေးညှိလိုက်ပြီး ShowEvenNum function ထဲကို pass လုပ်လိုက်တာပါ။

ShowEvenNum Function အလုပ်လုပ်သွားပုံ ကလဲ ရိုးရှင်းပါတယ်။ For loop ကိုအသုံးပြုပြီး Array length ဆုံးသည်အထိ loop လျှောက်ပါတယ်။ ဒီ Looping အတွင်းမှာမှ if statement ကိုအသုံးပြုပြီး array ထဲက element တွေဖြစ်တဲ့ i တန်ဖိုးတွေကို 2 နဲ့စားပြီး အကြွင်း သုညရတဲ့ elements တွေကြီး ထုတ်ပြပေးပါဆိုပြီး စစ်ထုတ်ထားလိုက်တာပါ။ နားလည်မရလို့ ထင်ပါတယ်။

Function Return

Return attribute ကို ကျနော်တို့ အပေါ်လေ့ကျင့်ခန်းတွေမှာလဲ အသုံးပြုခဲ့ဖူးကြပါတယ်။ ဘယ်လို အလုပ်လုပ်တယ်ဆိုတာ နားလည်နေကြပြီ ထင်ပါတယ်။ Return ရဲ့ အလုပ်ကတော့ သူနဲ့ ဖမ်းထားတဲ့ variable value ကို သူနဲ့သတ်ဆိုင်တဲ့ function, object,... စတာတွေဆီကို ပို့ဆောင်ပေးပါတယ်။ လေ့လာကြည့်ကြအုံးစို့ဗျာ။

```
<html>
<head><title>Return values in function</title></head>
<body>
<script type="text/JavaScript">
function MultiplyByFive(Num)
{
    return Num*5;
}
document.write(" Multiplication of 3 and 5= ",MultiplyByFive(3));
</script>
</body>
</html>
```

Result:

Multiplication of 3 and 5=15

အရမ်းရိုးရှင်းတဲ့ ပုစ္ဆာလေးပါ။ 3 ဆိုတဲ့ value ကို MultiplyByFive Function ထဲကို Num အဖြစ် pass လုပ်ထားပြီး ထို Num ကိုမှ 5 နဲ့ မြှောက်ပြီး ရလာတဲ့ ရလဒ်ကို return အဖြစ် MultiplyByFive Function ထဲကို ပြန်ထည့်ထားတာပါ။ ဒါကြောင့် document.write() ထဲမှာ MultiplyByFive function ကိုညွှန်းပေးထားယုံနဲ့ နောက်ဆုံးရလဒ်ကို ဖော်ပြပေးနိုင်တာပါ။ နားလည်ကြမရီ ထင်ပါတယ်။

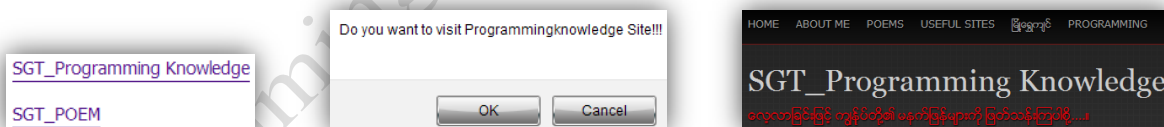


Return Boolean value from function

ပုံသေအနေနဲ့ True / False သာ စစ်ချင်တာဆိုရင်တော့ boolean return ပုံစံကို အသုံးပြုရပါတယ်။ True ဆိုရင် သတ်မှတ်ထားတဲ့ အလုပ်ကိုလုပ်ပြီး False ဖြစ်ရင်တော့ ဘာအလုပ်မှ မလုပ်ပဲ program က ထွက်သွားမှာဖြစ်ပါတယ်။

```
<html>
<head><title>Return Boolean value from function</title>
<script type="text/JavaScript">
function CheckSite(SiteURL)
{
    if(confirm("Do you want to visit " +SiteURL+ " Site!!!"))
    {
        return true;
    }
    else
    {
        return false;
    }
}
</script>
</head>
<body>
<p><a href="http://programmingknowledge.blogspot.com" onClick="return CheckSite('SGT program')">
SGT_Programming Knowledge</a></p>
<p><a href="http://shwegyinthaipoem.blogspot.com" onClick="return CheckSite('SGT Poem')">
SGT_POEM</a></p>
</body>
</html>
```

Result:



ပုံတွေမှာမြင်တဲ့အတိုင်းပါပဲ။ code ကို run လိုက်ရင်ဖြင့် ပုံ(၁) ပုံစံပြပါလိမ့်မယ်။ ကျနော်တို့က နှစ်သက်ရာ site link တစ်ခုခုကို နှိပ်လိုက်ပြီ ဆိုရင်တော့ ပုံ(၂) လို ကျနော်တို့ကို ထို site ကိုတက်ရ သွားလည်မှာလား မလည်ဘူးလားဆိုတာ ရွေးချယ်ခွင့်ပေးထားပါတယ်။ ok button ကိုနှိပ်လိုက်ရင်တော့ ပုံ(၃) အတိုင်း သူရည်ညွှန်းတဲ့ URL ကိုပြပေးပြီး cancel button နှိပ်ရင်တော့ program ကထွက်သွားပါလိမ့်မယ်။

Code ကိုရှင်းရမယ်ဆိုရင်တော့ ကျနော်တို့က CheckSite ဆိုတဲ့ function တခုတည်ဆောက်ပြီး အတွင်းမှာ if statement နဲ့ confirm box ကိုတွဲဖတ်အသုံးပြုကာ user အတွက် True, False ဆိုတဲ့ တန်ဖိုးနှစ်ခုကို return ပြန်ပေးထားပါတယ်။ Return true က confirm box ရဲ့ OK button အတွက်၊ Return false ကတော့ Cancel button အတွက်ပေါ့ဗျာ။ ဒါကိုမှ အောက်က <body> ထဲက onClick မှာ



CheckSite Function ကို user ရွေးချယ်ပေးတဲ့ value အတိုင်း return ပြန်ပေးပါဆိုပြီး onClick="return CheckSite('...')"

Create the Dynamic function as Object

Dynamic function ပုံစံကိုတော့ function တွေကို variable တွေအဖြစ် အသေးသုံးချင်တဲ့ အခါမျိုးမှာ အသုံးပြုနိုင်ပါတယ်။ သူ့ရဲ့ syntax ပုံစံလေးကတော့...

```
var variableName = new Function(argument1,argument2,...argumentN);
```

သဘောတရားကိုသိပြီဆိုရင်တော့ ကျနော်တို့ လေ့ကျင့်ခန်းလေးတခုလောက် လေ့လာကြည့်ကြရအောင်ဗျာ။

```
<html>
<head><title>Dynamic function as object</title></head>
<body>
<script language='JavaScript'>
  var multiplyByFive = new Function("y","return y*5");
  document.write("Multiplication of 3 and 5= ",multiplyByFive(3));
</script>
</body>
</html>
```

Result:

Multiplication of 3 and 5= 15

Call a function in hyperlink

Function တွေကို hyperlink ကခေါ်သုံးနိုင်တာကိုတော့ for loop သင်ခန်းစာမှာ ကျနော်တခါသုံးပြခဲ့ဖူးပါတယ်။ အခု လေ့ကျင့်ခန်း တပုဒ်လောက် ဆက်လေ့လာကြည့်ကြအုံးစို့ဗျာ။

```
<html>
<head><title>Square by Function</title>
<script type="text/javascript">
function Rectangle( )
{
  var breadth=prompt("Enter the breadth of rectangle");
  var height=prompt("Enter the height of rectangle");
  AreaRectangle=breadth*height;
  PerimeterRectangle=(2*breadth)+(2*height);
  alert("Area of Rectangle= " +AreaRectangle+ " and Perimeter of Rectangle= " +PerimeterRectangle);
}
</script>
</head>
<body>
<h5>Rectangle Formula:</h5>
  Area of Rectangle    =>breadth * height<br>
  Perimeter of Rectangle=>(2*breadth)+(2*height)
<p><a href="javascript:Rectangle( )">Calculated Area and Perimeter of Rectangle</a></p>
```



```
</body>
</html>
```

Result:

အလုပ်လုပ်သွားပုံက အရမ်းရိုးရှင်းပါတယ်။ Prompt box ကိုအသုံးပြုပြီး ကျနော်တို့ Rectangle ရဲ့ breadth နဲ့ height ကိုရိုက်ထည့်ခိုင်းပြီး Formula တွေအတိုင်း တွက်ပေးလိုက်ယုံပါပဲ။ Formula ရဲ့ လုပ်ဆောင်ချက်အားလုံးကို function ထဲမှာ တွက်ထုတ်ထားပြီး ၎င်း function ကို href tag နဲ့ ဆွဲသုံးနိုင်တဲ့ `....` ကိုအသုံးပြုထားတာပါ။ နားလည်နိုင်ကြမယ် ထင်ပါတယ်။ မိတ်ဆွေအားလုံး လေ့လာခြင်းဖြင့် ကျေနပ်နိုင်ကြပါစေ။

ဒီနေ့တော့ ဒီလောက်နဲ့ပဲ နားကြအုံးစို့ဗျာ။ အားလုံးက အခြေခံလေးတွေ ဖြစ်လို့ စတင်လေ့လာသူများ အတွက်သာ ရည်ရွယ်ပါတယ်။ နောက်နေ့ အပိုင်းလေးမှာတော့ Javascript Function သင်ခန်းစာလေးတွေထဲကပဲ နားလည်လွယ်တာလေးတွေနဲ့ လိုအပ်တဲ့ အပိုင်းလေးတွေကို ဆွေးနွေးသွားပါမယ်။

ဒါကတော့ Javascript အားအသုံးပြုခြင်း အပိုင်း(17)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။ လေ့လာပြီးသား သူတွေအတွက်လဲ ပြန်လည် အမှတ်ရစေနိုင်မရဲလို့ မျှော်လင့်ပါတယ်။ မိတ်ဆွေအပေါင်း လေ့လာခြင်းဖြင့် ကျေနပ်နိုင်ကြပါစေ။

Tay Zar Lin**Koyinmaung007@gmail.com****Programmingknowledge.totalh.com**