



JavaScript အားအသုံးပြုခြင်း

While loop in Javascript

While loop ကိုတော့ Javascript မှာ condition တစ်ခုတည်းကိုပဲ ထပ်ခါတလဲလဲ စစ်ဆေးပြီး Looping ကို အသက်ဝင်စေချင်တဲ့အခါမှာ အသုံးပြုနိုင်ပါတယ်။ သူ့ရဲ့ syntax ပုံစံလေးကတော့....

```
while(expression)
{
    statement;
}
```

အလုပ်လုပ်ပုံကတော့ program မှာ while() ကိုတွေ့ပြီးဆိုတာနဲ့ သူ့ရဲ့ condition ကို စစ်ဆေးပါတယ်။ ထို condition မှန်နေသ၍ သူ့ရဲ့ statement ကို ထပ်ခါတလဲလဲ အလုပ်လုပ်ပေးပြီး condition မှားသွားပြီဆိုရင်တော့ looping ကထွက်သွားပြီး program ရဲ့ ကျန်ရှိနေသေးတဲ့ statement တွေကို အလုပ်ဆက်လုပ်စေပါတယ်။ သဘောတရားကို နားလည်ရင် ကျနော်တို့ လေ့ကျင့်ခန်းလေးတွေနဲ့ လေ့လာကြည့် ကြမယ်။

```
<html>
<head><title>While loop</title></head>
<body>
<script type="text/javascript">
var count = 1;
document.write("Starting Loop<br>");
while (count <= 10)
{
    document.write("Current Count : " + count + "<br />");
    count++;
}
document.write("Loop stopped!");
</script>
<p>Set the variable to different value and then try...</p>
</body>
</html>
```

Result:

```
Starting Loop
Current Count : 1
Current Count : 2
Current Count : 3
Current Count : 4
Current Count : 5
Current Count : 6
Current Count : 7
Current Count : 8
Current Count : 9
Current Count : 10
Loop stopped!
Set the variable to different value and then try...
```

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။



ခက်ခက်ခဲခဲ မရှိလို့ နားလည်မရလို့တော့ ထင်ပါတယ်။ ကျနော်တို့က count=1 လို့ အသေကြေကြာထားပြီးတော့ while(count<=10) ဆိုပြီး condition စစ်ထားပါတယ်။ ဒီ condition မှန်နေသရွေ့ count ကို ရိုက်ထုတ်ပြသွားပြီး count++; ဆိုပြီး loop တစ်ခတ်တိုင်း count ကို တစ်တိုးပြီး အလုပ်လုပ်ခိုင်းထားပါတယ်။ While condition မှားသွားပြီ ဆိုရင်တော့ while loop ကထွက်သွားပြီး အပြင်ဘက် statement ဖြစ်တဲ့ Loop stopped ဆိုတာကို ထုတ်ပြပေးသွားတာပါ။ ဒီပုစ္ဆာကတော့ count ဆိုတဲ့ စမှတ်ကို အသေထားပြီး အလုပ်လုပ်စေတာပါ။ ဒါကိုမှ ကျနော်တို့ နှစ်သတ်ရာ Number ကို ရိုက်ထည့်ပြီး while loop ကို အလုပ်လုပ်ခိုင်း နိုင်ပါသေးတယ်။ Javascript မှာ user input ကို အသုံးပြုမရဆိုရင်တော့ prompt box ကို အသုံးပြုပါမယ်။ လေ့လာကြည့်လိုက်ပါအုံးဗျာ။

```
<html>
<head><title>While loop</title></head>
<body>
<script type="text/javascript">
var num=prompt("Enter a number >0: ");
while (num>0)
{
    document.write("Current Count : " + num + "<br />");
    num--;
}
document.write("Loop Out! Invalid number.!!!!");
</script>
<p>Set the variable to different value and then try...</p>
</body>
</html>
```

Result:

Enter a number >0:

OK Cancel

Loop Out! Invalid number.!!!!

Set the variable to different value and then try...

For num<0

Enter a number >0:

OK Cancel

Current Count : 6
 Current Count : 5
 Current Count : 4
 Current Count : 3
 Current Count : 2
 Current Count : 1
 Loop Out! Invalid number.!!!!

Set the variable to different value and then try...

For num>0

ဒီပုစ္ဆာမှာတော့ ကျနော်တို့က စမှတ်ကို user စိတ်ကြိုက် ရိုက်ထည့်ခိုင်းထားပါတယ်။ User က စမှတ်ရိုက်ထည့်လိုက်ပြီဆိုတာနဲ့ while(num>0) ဆိုပြီး while condition ကိုစစ်ပါပြီး။ Result ပုံတွေမှာ ပြထားတဲ့

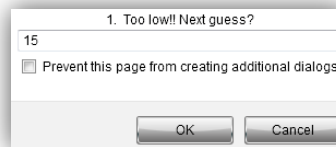
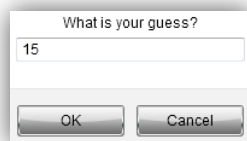


အတိုင်းပါပဲ။ ကျနော်တို့က သုညအောက် ငယ်တဲ့ number နဲ့ စစ်မိရင်တော့ number Invalid ဆိုပြီးပဲ loop ကို အလုပ်မလုပ်ပဲ ခုန်ထွက်သွားမှာပါ။ ကျနော်တို့က သုညနဲ့ ကြီးတဲ့ ကိန်းနဲ့ စစ်မိရင်တော့ while condition ကမှန်သွားပါပြီး ဒီတော့ အလုပ်မလုပ်ပြီပေါ့။ ဒီပုစ္ဆာမှာတော့ decrement operator ကို အသုံးပြုထားတဲ့အတွက်ကြောင့် loop တစ်ခါပတ်တိုင်း num ကို တစ်လျှော့လျှော့သွားပြီး နောက်ဆုံး while condition မှားတဲ့အခြေအနေထိ အလုပ်လုပ်ပေးသွားမှာပါ။ နားလည်မရင်ထင်ပါတရ်ဗျာ။

ဒီတစ်ခါတော့ ကျနော်တို့ while loop ကို အသုံးပြုပြီး keyword ထားတဲ့သဘောကို ဂိမ်းဆန်ဆန် လေးနဲ့ သုံးနိုင်မဲ့ လေ့ကျင့်ခန်းလေးတစ်ခုလောက် လေ့လာကြည့်ကြအုံးစို့ဗျာ။

```
<html>
<head><title>While Loop</title>
<script type="text/javascript">
var key =53;
var turns = 0;
var msg ="";
var guess =prompt("What is your guess?","Enter your number...");
while (guess != key)
{
    turns++;
    if (guess > key)
    {
        guess =prompt (turns + ". Too high!! Next guess?", guess);
    }
    if (guess < key)
    {
        guess = prompt (turns + ". Too low!! Next guess?", guess);
    }
}
if (guess == key)
{
    msg = "YOU WIN!!! \n";
    msg += "It took you ";
    msg += turns + " turns! ";
    alert (msg);
}
</script>
</head>
</html>
```

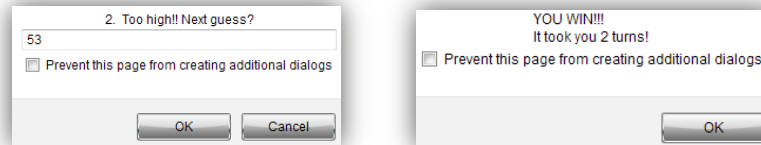
Result:



For guess<key



For guess>key



For guess==key

ပုစ္ဆာတိုင်းကတော့ ခက်ခက်ခဲခဲမရှိပါဘူး။ ဒီပုစ္ဆာမှာဆိုရင်လဲ ကျနော်တို့က keyword ဖြစ်တဲ့ key ကို 53 အဖြစ် အသေသတ်မှတ် ကြေငြာထားပါတယ်။ သူနဲ့ တိုက်ဆိုင်စစ်ဆေးမဲ့ user input key ကိုတော့ prompt box ကို အသုံးပြုပြီး keyboard ကိုရိုက်ထည့် ခိုင်းထားပါတယ်။ User က number တခုခုကို ရိုက်ထည့်လိုက်ပြီ ဆိုရင်တော့ while(guess !=key) ဆိုပြီး while condition နဲ့ စစ်စစ်ပါပြီး။ Condition မှန်ရင်တော့ သူ့ရဲ့ statement နှစ်ခုဖြစ်တဲ့ if(guess>key) နဲ့ if(guess<key) နှစ်ခုနဲ့ ဆက်လက် စစ်ဆေးပါတယ်။ မှန်တဲ့ statement ကိုထုတ်ပါတယ်။ ဒီလို ထုတ်ပြတဲ့နေရာမှာ prompt box နဲ့ လက်ဆက်တွဲခေါ်ထားတဲ့အတွက် user ကို number တခု ထပ်ရိုက်ထည့်ခိုင်းပါလိမ့်မယ်။ ဒီလိုနဲ့ guess==key မဖြစ်သေးတဲ့ အခြေအနေထိ ကြီးရင်ကြီးတယ်၊ ငယ်လျှင် ငယ်တစ်ခုဆိုပြီး ထပ်ခါတလဲ while loop က အလုပ်လုပ်နေမှာပါ။ guess==key ဖြစ်သွားတဲ့ အခြေအနေ ရောက်သွားရင်တော့ while loop ကခုန်ထွက်သွားပြီး သူ့ရဲ့အပြင်ဘက် if(guess==key) statement ကို ထုတ်ဖော်ပြသပါလိမ့်မယ်။

ဒီမှာ တခုပြောစရာရှိတာက turns နဲ့ msg ပါ။ turns ကိုတော့ စစ်ချင်းမှာ သုညလို့ သတ်မှတ်ပေးထားပြီး loop တခါပတ်တိုင်း တစ်တိုးသွားကာ user input အရေအတွက် ဘယ်လောက် ရှိနေပြီလဲဆိုတာကို မှတ်တမ်းတင်တဲ့ သဘောပါပဲ။ ဒီလို count သဘောတရားကို အသုံးပြုပြီး user အတွက် အခွင့်အရေးကို ကန့်သတ်နိုင်ပါတယ်။ ဥပမာ user ကို သုံးခါပဲ ကြိုးစားခွင့်ပေးမယ် စသည် ဖြစ်ပေါ့။ msg ကတော့ တမျိုးတဖုံ အသုံးပြုကြည့်လိုက်တာပါ။ ဒါကို မသုံးချင်ဘူးဆိုရင်လဲ alert() နေရာမှာ ဖော်ပြမဲ့ စာသားကိုတခါတည်း ရိုက်ထည့် ဖော်ပြထားနိုင်ပါတယ်။ ဒီပုစ္ဆာကတော့ keyword ကို အသေထားပြီး user ကို ထို keyword နဲ့ ကိုက်ညီအောင် ရိုက်ထည့်ကြိုးစားခိုင်းတာပါ။ ဒါကိုမှ random သဘောတရားကို အသုံးပြုပြီး keyword ကို random ခေါက်ထားခိုင်းပါသေးတယ်။ ဆက်လေ့လာ ကြည့်လိုက်ပါအုံးဗျာ။

```
<html>
<head><title>While Loop</title></head>
<body>
<script type="text/javascript">
var turns = 0;
var key= Math.floor(Math.random( ) * 100) + 1;
var guess =prompt("What is your guess?","Enter your number");
```



```

while (guess != key)
{
    turns++;
    if (guess > key)
    {
        guess = prompt (turns + ". Too high!! Next guess?", guess);
    }
    if (guess < key)
    {
        guess = prompt (turns + ". Too low!! Next guess?", guess);
    }
}
alert("YOU WIN!!!... If took you "+turns+ " turns!!");
</script>
</body>
</html>

```

အပေါ်က ပုစ္ဆာနဲ့ သဘောတရား အားလုံး အတူတူပါပဲ။ ဒီမှာ တစ်ခုထူးခြားသွားတာက ကျနော်တို့ ရှာဖွေရမဲ့ keyword က ပုံသေမရှိပါဘူး။ Math.random() ကိုအသုံးပြုပြီး random ခေါက်ထားပါတယ်။ ဒီတော့ program ကို run လိုက်တဲ့အခါတိုင်း key တန်ဖိုးတွေက မတူညီကြဘူးပေါ့။ Run ကြည့်လိုက်ရင် နားလည်သွားမှာပါ။

အခုလို keyword တွေကို number တွေကြီး အသုံးပြုနိုင်သလို string type သဘောမျိုးနဲ့လဲ အသုံးပြုနိုင်ပါသေးတယ်။ ဆက်လေ့လာကြည့်လိုက်ပါအုံးဗျာ။

```

<html>
<head><title>While Loop</title></head>
<body>
<script type="text/javascript">
var key="Tayza";
var guess = "";
while (guess != key)
{
    guess = prompt ("Enter your keyword:");
    if (guess == key)
    {
        alert (" Welcom " +key);
    }
    else
    {
        alert ("That's not it... Try again");
    }
}
</script>
</body>
</html>

```

**Result:**

Enter your keyword:

OK Cancel

That's not it... Try again

☐ Prevent this page from creating additional dialogs

OK

For uncorrect

Enter your keyword:

☐ Prevent this page from creating additional dialogs

OK Cancel

Welcom Tayza

☐ Prevent this page from creating additional dialogs

OK

For correct

အလုပ်လုပ်သွား ပုံတွေကိုတော့ ရှင်းပြစရာမလိုတော့ပဲ နားလည်နိုင်ပြီ ထင်ပါတယ်။ ဒီမှာ တခုပြောစရာ ရှိတာက ကျနော်တို့ user input ဖြစ်တဲ့ prompt box ကို while(guess != key) ဆိုတဲ့ condition ထဲမှာ ထည့်သွားထားတဲ့အတွက် ကျနော်တို့ ရိုက်ထည့်လိုက်တဲ့ keyword မှားတိုင်း(သို့) condition မှန်နေတဲ့ အခါတိုင်းမှာ ထပ်ခါတလဲလဲ ဆက်အလုပ်လုပ် ခိုင်းနေပါလိမ့်မယ်။ ဒီတော့ key မမှန်မချင်း ကြိုးစားခိုင်းတဲ့ သဘောပဲပေါ့။ ဒီလောက်ဆိုရင်ဖြစ် while loop အကြောင်းကို နားလည်နိုင်ပြီလို့ ထင်ပါတယ်။ သုံးတတ်ပြီလို့လဲ ယူဆပါတယ်။ ကျနော်တို့ ရှေ့ဆက်ကြပါအုံးစို့။

Do...While statement in Javascript

Do...while နဲ့ while က သဘောတရားချင်း အတူတူပါပဲ။ အလုပ်လုပ်ပုံချင်း အနည်းငယ်သာ ကွာခြားပါတယ်။ While ကတော့ program စလျှင်စချင်း condition ကိုစစ်ပါတယ်။ မှန်နေရင် ဆက်အလုပ်မရှိ။ မှားရင် looping ကထွက်မရဲပေါ့။ Do...while ကတော့ အဲ့လိုမဟုတ်ပါဘူး။ Looping ထဲကို အတားအစီးမရှိ do ကိုကျော်ပြီး looping body ထဲက statement တွေကို ပထမ execute လုပ်ပါတယ်။ ပြီးမှ while(condition) ကိုစစ်ပါတယ်။ မှန်ရင် ဆက်အလုပ်လုပ်မရှိ။ မှားရင် looping ကထွက်မရှိ။ သူ့ရဲ့ syntax ပုံစံလေးကတော့....

```
do{
    Statement;
}while(expression);
```

သဘောတရားကို နားလည်ပြီးဆိုရင်တော့ ကျနော်တို့ လေ့ကျင့်ခန်းလေးတွေ လေ့လာကြည့် ကြအုံးစို့။

```
<html>
<head><title>While Loop</title></head>
<body>
<script type="text/javascript">
var Sum=0;
var x=1;
var num=prompt("Enter last number:");
do{
    Sum+=x;
```

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။



```

++x;
}while(x<=num);
alert("Addition of your series 1 to "+num+" = "+Sum);
</script>
</body>
</html>

```

Result:

Enter last number:

OK Cancel

Addition of your series 1 to 100= 5050

☐ Prevent this page from creating additional dialogs

OK

အလုပ်လုပ်သွားပုံက ရိုးရှင်းပါတယ်။ ပထမဆုံးအနေနဲ့ ကျနော်တို့ကို prompt box အတွင်းမှာ နောက်ဆုံး ကိန်းကို ရိုက်ထည့်ခိုင်းပါတယ်။ ထိုနောက် လုပ်ဆောင်ချက် အားလုံးကို do statement ထဲမှာ လုပ်ကိုင်ခွင့် ပေးထားပြီး နောက်ဆုံးမှာ while(x<=num)ဆိုပြီးတော့ while condition နဲ့ စစ်လိုက် တာပါ။ Condition မှန်နေသေးသရွေ့ x ကိုတစ်တိုးတိုးပေးပြီးပြီး Sum ထဲကို ပေါင်းပေါင်းထည့်ခဲ့တာပါ။ နောက်ဆုံးတော့ ကျနော်တို့ series တခုလုံးရဲ့ ပေါင်းလဒ်ကို ရရှိပါလိမ့်မယ်။ နားလည်မရ ထင်ပါတယ်။ နောက်တပုဒ် ဆက်လေ့လာကြည့်ကြအုံးစို့ဗျာ။

```

<html>
<head><title>While Loop</title></head>
<body>
<script type="text/javascript">
var a, b, c, d;
do{
    a=prompt("Enter your first num: ");
    b=prompt("Enter your second num:");
    c=parseInt(a)+parseInt(b);
    alert("Addition of " +a+ " and " +b+ " = " +c);
    d=prompt("Do you want to do this again(Y/N):");
}while(d=='Y' || d=='y');
alert("Good Bye....!!!!");
</script>
</body>
</html>

```

Result:

Enter your first num:

OK Cancel

Enter your second num:

☐ Prevent this page from creating additional dialogs

OK Cancel

Addition of 456 and 263= 719

☐ Prevent this page from creating additional dialogs

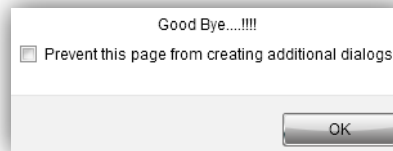
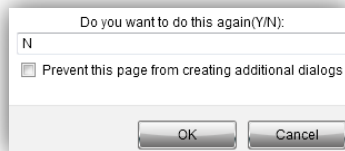
OK

အခုလို operation တခုကို လုပ်ဆောင်ပြီးသွားပြီဆိုရင်တော့ ကျနော်တို့ကို နောက်ထပ် operation တခု ဆက်လုပ်အုံးမလား(Y/N) ဆိုပြီး ဆက်မေးပါလိမ့်မယ်။ ကျနော်တို့က Y (or) y လို့ ရိုက်ထည့်

လေ့လာခြင်းဖြင့် ကျွန်ုပ်တို့၏ မနက်ဖြန်များကို ဖြတ်သန်းကြပါစို့....။



လိုက်ရင်တော့ နောက်ထပ် operation တခု ဆက်အလုပ်လုပ်ရန်အတွက် Enter your first num ဆိုတဲ့ prompt box ထပ်ပေါ်လာပါလိမ့်မယ်။ ဒီလိုပဲ မိမိ လုပ်ဆောင်ချင်သလောက် Y/N ရွေးပေးပြီး လုပ်ဆောင် သွားနိုင်ပါတယ်။ N(or)တခြားကီးတခုခု နှိပ်လိုက်ရင်တော့ do..while loop က ထွက်သွားပြီး တခြား statement တခုကို ဆက်အလုပ်လုပ်သွားပါလိမ့်မယ်။



ဒီပုစ္ဆာ အလုပ်လုပ်သွားပုံကလဲ ခက်ခက်ခဲခဲမရှိပါဘူး။ User ကိုနှစ်သက်ရာ ကိန်းနှစ်လုံး ရိုက်ထည့်ခိုင်းပြီး ပေါင်းခြင်းအလုပ်လုပ်ကို လုပ်သွားတာပါ။ User input အတွက် prompt box ကို အသုံးပြုထားပြီး ရိုးရိုးပဲ ပေါင်းပြထားတာပါ။ ဒီမှာ ကျနော်တို့က integer type တွေအဖြစ် number တွေကြီးကို အလုပ်လုပ် စေမှာဖြစ်တဲ့အတွက် ကျနော်တို့ prompt box တွေက ရိုက်ထည့်လိုက်တဲ့ string type တွေကို parseInt() ဆိုတဲ့ string to integer ပြောင်းနိုင်တဲ့ function လေးကို ခေါ်သုံးထားပါတယ်။ ရှေ့က Data conversion အခန်းမှာ ရှင်းပြခဲ့ဖူးပါတယ်။ ပြန်လေ့လာကြည့်လိုက်ပေါ့ဗျာ။ ဒီမှာ do...while loop ကိုတော့ နောက်ထပ် operation ကို ထပ်လုပ်မလား၊ လုပ်အုံးမလားဆိုတဲ့ နေရာမှာ စစ်ထားပါတယ်။ ကျနော်တို့ variable d ရဲ့ တန်ဖိုးက Y/y လို့ ရိုက်ထည့်နေသရွေ့ ပေါင်းခြင်းအလုပ်ကို ထပ်ခါတလဲလဲ လုပ်နိုင်ပါတယ်။ N/other key တခုခု ရိုက်ထည့်လိုက်ရင်တော့ loop ကထွက်သွားပြီး Good Bye ဆိုတဲ့ statement ကိုထုတ်ပြပေးသွားပါလိမ့်မယ်။ အခုလောက်ဆိုရင်တော့ Javascript မှာ do....while loop ကို နားလည်ပြီး၊ အသုံးပြုနိုင်ပြီလို့ ယူဆပါတယ်။ မိတ်ဆွေအားလုံး လေ့လာခြင်းဖြင့် ကျေနပ်နိုင်ကြပါစေ။

ဒီနေ့တော့ ဒီလောက်နဲ့ပဲ နားကြအုံးစို့ဗျာ။ အားလုံးက အခြေခံလေးတွေ ဖြစ်လို့ စတင်လေ့လာသူများ အတွက်သာ ရည်ရွယ်ပါတယ်။ နောက်နေ့ အပိုင်းလေးမှာတော့ Javascript statements သင်ခန်းစာလေးတွေထဲက for, break, continue,... စတဲ့ statement လေးတွေအကြောင်း ဆွေးနွေးသွားပါမယ်။

ဒါကတော့ Javascript အားအသုံးပြုခြင်း အပိုင်း(13)လေးပါ။ အားလုံးက အခြေခံလေးတွေဖြစ်လို့ 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.totalh.com