« JS Function Return

JS Function Private »

Javascript Closures and Recursions

More advanced applications of Javascript functions.

Javascript Function: Closures

In Javascript, closures are functions that have access to variables from another functions scope.This is achieved by creating a function within a function

The outer function creates a ***reference*** to the inner anonymous function, and thus making it possible to call the inner function via the reference.

The idea of closures it that the local variables must remain ***accessible*** to inner functions even when it seems it is out of scope.

Syntax: Javascript Function Closures

```
<script>

</script>
```

Example: Javascript Function Return Values

```
<!DOCTYPE html>
<html>
<head>
  <title>Javascript Functions - Closures</title>
    <script>
      function Cars(brand, name){
      var text = "Car's brand is " + brand + " and name is " + name; //local variable

      var output = function() //Anonymous function
      {
      document.write(text + "<hr>");
      }
      return output;
      }
    </script>
</head>
<body>
<script>
var input1 = Cars("Honda", "Accord"); /*A reference to the anonymous function*/

var input2 = Cars("Toyota", "Camry");

alert(input1);

input1(); //Anonymous function called.
input2();
</script>
</body>
</html>
```

Note: All javascript functions are closures, as they are objects and have a scope chain associated with them.

Give it a TRY! »

Javascript Functions: Recursion

By definition, a *recursive function* is created when a function calls itself by name.

In Recusion, the program starts executing at the beginning of the function and backs up to where it was when it called the function and starts executing from that point, if the test condition is satisfied.

But most importantly, there must be a way to stop the recursion at some point, or else it would be infinite, ultimately causing the program to crash.

Syntax: Javascript Anonymous Function as a Variable

```
<script>
var variable = function()
{
```

```
4  ......
5  ......
6  }
7  </script>
```

Example: Javascript Functions - Recursion

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Javascript Functions : Recursion</title>
5      <script>
6       var counter = 0;
7       function fibo(num)
8       {
9       counter++;
10      switch(num)
11      {
12      case 0:
13         return(0);
14         break;
15      case 1:
16         return(1);
17         break;
18      default:
19         return(fibo(num - 1) + fib( num -2));
20         break;
21         }
22     }
23     </script>
24 </head>
25 <body>
26 <script>
27 for (i=0; i < 30 ; i++)
28 {
29 output = fibo(i);
30 document.write(output + "   ");
31 }
32 alert("The Recursion Function was called " + output + " times");
33 </script>
34 </body>
35 </html>
```

Note: For each iteration the function ***fibo()*** is called.

Give it a TRY! »

« JS Function Return

JS Function Private »