Joseph Camacho-Terrazas
Chapter 9 binding of global variables during subroutine passing
10/16/2020

**Problem Description**

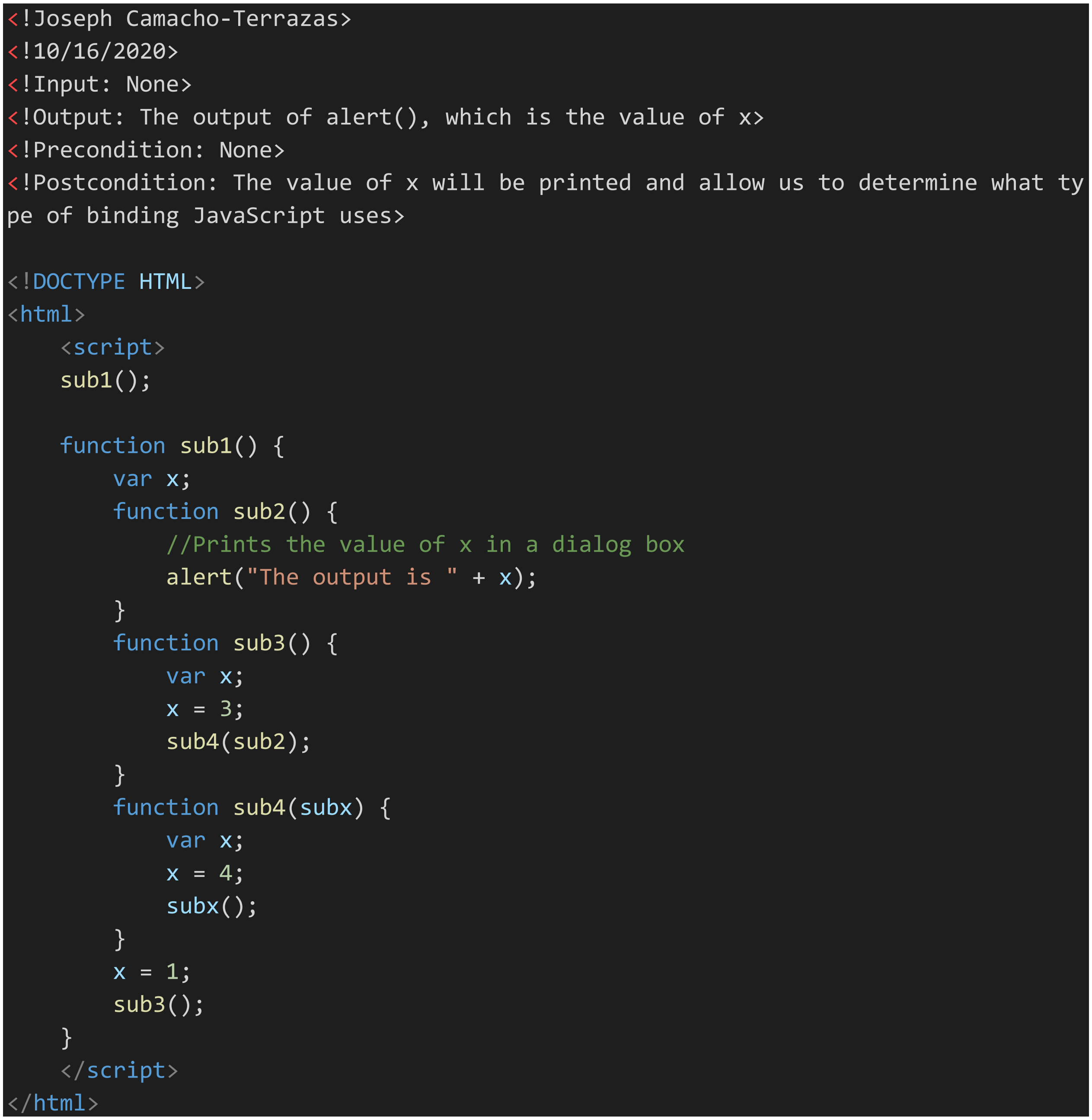
The problem that we are trying to solve is finding out whether JavaScript uses shallow binding, deep binding, or ad hoc binding. To solve this problem, I implemented the code from the book. First, I created a small HTML page that implements the JavaScript code. The alert() function creates a popup window when the HTML page is run in a browser and gives us the result to determine the binding type. Then I wrote a call to sub1() to run the whole program. This gave me an output of 1 when run in Google Chrome.

**Conclusion**

Based on the output of the program, I can conclude that JavaScript uses deep binding. This is because when we call alert() within sub2(), it prints out “1”. We can see in the code that x = 1 under sub1()’s environment. This means that when sub2() is called, it’s referencing environment is that of sub1(). So, the “x” referenced in sub2() is bound to the local “x” of sub1(), therefore we get the output of “1”.

Joseph Camacho-Terrazas
Chapter 9 binding of global variables during subroutine passing
10/16/2020

**HTML with embedded JavaScript**



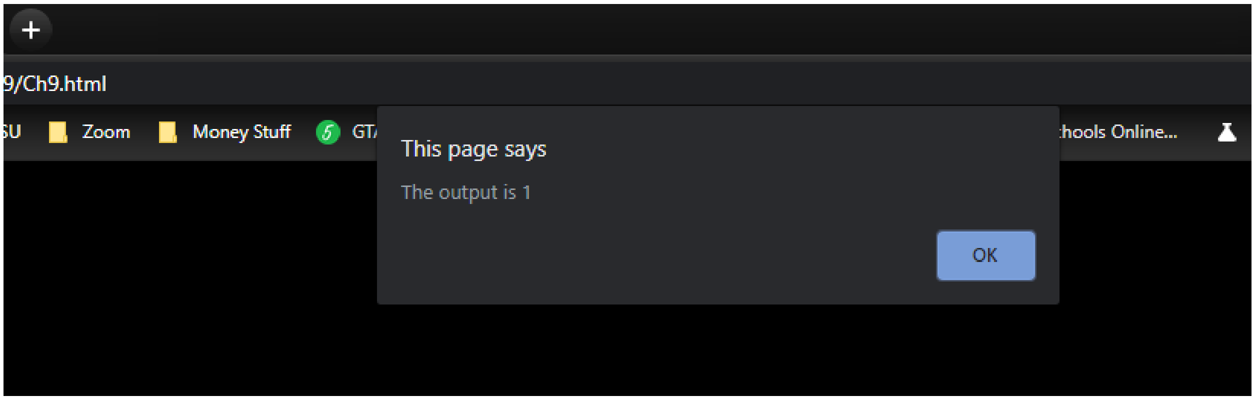
```
<!Joseph Camacho-Terrazas>
<!10/16/2020>
<!Input: None>
<!Output: The output of alert(), which is the value of x>
<!Precondition: None>
<!Postcondition: The value of x will be printed and allow us to determine what type of binding JavaScript uses>

<!DOCTYPE HTML>
<html>
    <script>
    sub1();

    function sub1() {
        var x;
        function sub2() {
            //Prints the value of x in a dialog box
            alert("The output is " + x);
        }
        function sub3() {
            var x;
            x = 3;
            sub4(sub2);
        }
        function sub4(subx) {
            var x;
            x = 4;
            subx();
        }
        x = 1;
        sub3();
    }
    </script>
</html>
```

Joseph Camacho-Terrazas
Chapter 9 binding of global variables during subroutine passing
10/16/2020

**Output Run in Google Chrome**



This page says

The output is 1

Console output