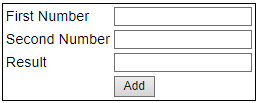
**different methods that are available in JavaScript to convert strings to numbers**. Let's design the web form as shown below.   
  
  
  
  
This form allows the user to enter 2 numbers and add them. Along the way we will learn the use of the following functions  
1. **parseInt()**  
2. **parseFloat()**  
3. **isNan()**  
  
**Here is the HTML for the web form**

<table style="border:1px solid black; font-family:Arial">

    <tr>

        <td>First Number</td>

        <td><asp:TextBox ID="txtFirstNumber" runat="server"></asp:TextBox></td>

    </tr>

    <tr>

        <td>Second Number</td>

        <td><asp:TextBox ID="txtSecondNumber" runat="server"></asp:TextBox></td>

    </tr>

    <tr>

        <td>Result</td>

        <td><asp:TextBox ID="txtResult" runat="server"></asp:TextBox></td>

    </tr>

    <tr>

        <td>

        </td>

        <td>

            <input type="button" value="Add" id="btnAdd"/>

        </td>

    </tr>

</table>

**Include the following JavaScript in the head section of the web form.**

<script type="text/javascript">

    function addNumbers()

    {

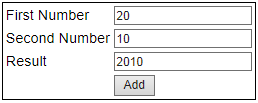
        var firstNumber = document.getElementById("txtFirstNumber").value;

        var secondNumber = document.getElementById("txtSecondNumber").value;

        document.getElementById("txtResult").value = firstNumber + secondNumber;

    }

</script>

Set the **onclick**attribute of the Add button to call the **addNumbers()** function. The HTML for the button should be as shown below.  
  
<inputtype="button"value="Add"id="btnAdd"onclick="addNumbers()"/>  
  
Run the application and enter 20 as the first number and 10 as the second number. Click Add buton. Notice that JavaScript concatenates the numbers instead of adding them. This is because the value property of the textbox is returning the number in a string format.   
  
  
So we need to **explicitly do the conversion**. This is when parseInt() function is useful. Modify the addNumbers() JavaScript function as shown below.

function addNumbers()

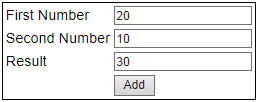
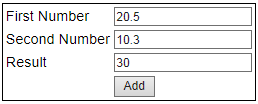
{

    var firstNumber = parseInt(document.getElementById("txtFirstNumber").value);

    var secondNumber = parseInt(document.getElementById("txtSecondNumber").value);

    document.getElementById("txtResult").value = firstNumber + secondNumber;

}

Run the application and test. Notice that we now get 30 as expected.  
  
  
Let's do another test. Enter 20.5 as the first number and 10.3 as the second number. Click the Add button. Notice that the decimal part is ignored.  
  
  
To retain the decimal places, use **parseFloat() function**.

function addNumbers()

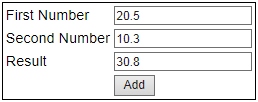
{

    var firstNumber = parseFloat(document.getElementById("txtFirstNumber").value);

    var secondNumber = parseFloat(document.getElementById("txtSecondNumber").value);

    document.getElementById("txtResult").value = firstNumber + secondNumber;

}

  
  
If you leave first number and second number textboxes blank or if you enter text instead of a number, and when you click the Add button, **NaN**is displayed in result textbox.  
  
**NaN in JavaScript stands for Not-a-Number.** In JavaScript we have isNaN() function which determines whether a value is an illegal number. This function returns true if the value is not a number, and false if not.  
  
Modify the addNumbers() JavaScript function as shown below.

function addNumbers()

{

    var firstNumber = parseFloat(document.getElementById("txtFirstNumber").value);

    if (isNaN(firstNumber))

    {

        alert("Please enter a valid number in the first number textbox");

        return;

    }

    var secondNumber = parseFloat(document.getElementById("txtSecondNumber").value);

    if (isNaN(secondNumber))

    {

        alert("Please enter a valid number in the second number textbox");

        return;

    }

    document.getElementById("txtResult").value = firstNumber + secondNumber;

}

Now, when you leave first number and second number textboxes blank or if you enter text instead of a number, and when you click the Add button, you get relevant validation error messages as expected.  
  
**Let's make the validation error message a little more relevant:**  
If the first number and second number textboxes are left blank, then we want to display the following validation messages  
a) First Number is required  
b) Second Number is required  
  
If you enter text instead of number  
a) Please enter a valid number in the first number textbox  
b) Please enter a valid number in the second number textbox  
  
To achieve this modify addNumbers() function as shown below.

function addNumbers()

{

    var firstNumber = document.getElementById("txtFirstNumber").value;

    var secondNumber = document.getElementById("txtSecondNumber").value;

    if (firstNumber == "")

    {

        alert("First Number is required");

        return;

    }

    firstNumber = parseFloat(firstNumber);

    if (isNaN(firstNumber))

    {

        alert("Please enter a valid number in the first number textbox");

        return;

    }

    if (secondNumber == "")

    {

        alert("Second Number is required");

        return;

    }

    secondNumber = parseFloat(secondNumber)

    if (isNaN(secondNumber))

    {

        alert("Please enter a valid number in the second number textbox");

        return;

    }

    document.getElementById("txtResult").value = firstNumber + secondNumber;

}