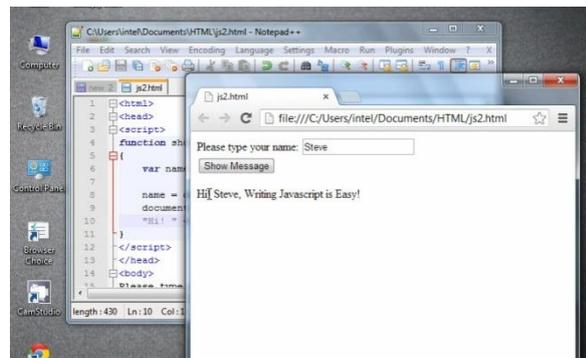


# Using JavaScript to carry out simple arithmetic

In the third video, Steve demonstrates how to carry out simple arithmetic using JavaScript. He codes a page that allows the user to input two numbers and then click a button to add them together. Follow the steps below to do the same.



## Step 1: Creating the file for this activity (video start to 0.54)

Steve starts with a file that already has some HTML5 code in it. You need to create a new file, and make sure you have everything in it that Steve uses in the video.

The screenshot on the right shows the code, identical to the example in the video. Open up a new file, create the html page, head section, and script tags. Create the body section and then insert the code that tells the user to enter two numbers, including the input type and id. Make sure you include the break tags at the end of each input section. Also create a new paragraph that shows where the area is on the web page where the answer to the sum will appear. Make sure your code looks the same as the screenshot above. Watch the video to make sure that your code is the same as Steve's and that you understand what the code is trying to do.

```

1 <html>
2 <head>
3 <script>
4 </script>
5 </head>
6 <body>
7 Please enter a number:
8 <input type="text" id="num1"/><br>
9 Please enter another number:
10 <input type="text" id="num2"/><br>
11 <input type="button" value="Add Numbers" onclick="add()" /><br>
12 <p id="messageArea"></p>
13 </body>
14 </html>
15

```



## Step 2: Adding the function to the script (video 0.55 – 01.23)

To do the next step, you have to stop Steve speaking mid-sentence! Stop the video at 01.23 minutes, after he has added three variables to a function within the script section of the code.

He goes into the script tag and adds the function 'add ()' and then defines three variables – two for each of the numbers and one for the result when they are added together. Insert this syntax into your code so it looks like the screenshot on the right. Note, the way the browser knows to add the two numbers together later is because of the 'function add ()' statement.

```
1 <html>
2 <head>
3 <script>
4 function add()
5 {
6     var num1;
7     var num2;
8     var result;
9
10
11 </script>
12 </head>
13 <body>
14 Please enter a number:
15 <input type="text" id="num1"/><br>
16 Please enter another number:
17 <input type="text" id="num2"/><br>
18 <input type="button" value="Add Numbers" onclick="add()" /><br>
19 <p id="messageArea"></p>
20 </body>
21 </html>
22
```

## Step 3: Telling the browser where to find the numbers (Video 01.24 – 02.07)

Now that Steve has told the browser that there are three variables, he needs to tell it where to find them. The next piece of code defines the variables and tells the browser where to find them.

The code uses the form...

**num1 = document.getElementById("num1").value;**

This should be familiar to you as we used the same syntax in the previous activity. To remind you, this tells the browser to find the value for number 1 (num1) in the HTML document, by getting the element with the id "num1".

In this piece of code, Steve gets the value of num2 as well, and then defines the result as numbers 1 and 2 added together. That is, num1 + num2.

You can see my code in the screenshot on the right. Make sure yours matches it (and Steve's!).

```
1 <html>
2 <head>
3 <script>
4 function add()
5 {
6     var num1;
7     var num2;
8     var result;
9
10     num1 = document.getElementById("num1").value;
11     num2 = document.getElementById("num2").value;
12     result = num1 + num2;
13
14 </script>
15 </head>
16 <body>
17 Please enter a number:
18 <input type="text" id="num1"/><br>
19 Please enter another number:
20 <input type="text" id="num2"/><br>
21 <input type="button" value="Add Numbers" onclick="add()" /><br>
22 <p id="messageArea"></p>
23 </body>
24 </html>
```



## Step 4: Telling the browser where to print the result (Video 2.08 – 2.35)

Steve now needs to tell the browser where to print the output of the sum – the result. He tells the browser to look in the HTML document and find the element with the id “messageArea” which is in the inner.HTML. This is defined as the result.

Type in the syntax that tells the document to produce the result in the message area. It should look like the screenshot on the right. Add the closing curly bracket to make sure the browser knows where to read the instructions you have given it.

```
1 <html>
2 <head>
3 <script>
4 function add()
5 {
6     var num1;
7     var num2;
8     var result;
9
10    num1 = document.getElementById("num1").value;
11    num2 = document.getElementById("num2").value;
12    result = num1 + num2;
13
14    document.getElementById("messageArea").innerHTML = result;
15 }
16 </script>
17 </head>
18 <body>
19 Please enter a number:
20 <input type="text" id="num1"/><br>
21 Please enter another number:
22 <input type="text" id="num2"/><br>
23 <input type="button" value="Add Numbers" onclick="add()"/><br>
24 <p id="messageArea"></p>
25 </body>
26 </html>
```

## Step 5: Saving and publishing the page (video 2.36 to 2.56)

In the next video, Steve saves and publishes his page. And you would hope that the published page does what he intends it to – that is, adds the two inputted numbers together.

Please enter a number:   
Please enter another number:   
  
45

Follow Steve’s instructions to save and publish your page, and then try to add two numbers together. You will see that rather than adding the numbers, the browser creates another number that is number 1 followed by number 2 (2 + 2 = 22!). So there is something wrong with the code. Make sure your code produces the effect that Steve demonstrates whenever you input two numbers.



## Step 6: Debugging the code by telling the browser to look for numbers, not text (Video 2.57 to end)

Steve explains that the browser thinks that the variables that are numbers 1 and 2 (num1 and num2) are text. So when they are added together, it reads the first number followed by the second, rather than adding them together.

He introduces a new command 'parseInt' which tells the browser to check whether the variables are numbers (integers) and if they are, they are then treated as numbers, and are added together.

```
1 <html>
2 <head>
3 <script>
4 function add()
5 {
6   var num1;
7   var num2;
8   var result;
9
10  num1 = parseInt(document.getElementById("num1").value);
11  num2 = parseInt(document.getElementById("num2").value);
12  result = num1 + num2;
13
14  document.getElementById("messageArea").innerHTML = result;
15  }
16 </script>
17 </head>
18 <body>
19 Please enter a number:
20 <input type="text" id="num1"/><br>
21 Please enter another number:
22 <input type="text" id="num2"/><br>
23 <input type="button" value="Add Numbers" onclick="add()"/><br>
24 <p id="messageArea"></p>
25 </body>
26 </html>
```

To add the code, he types in the new command and then puts everything about the numbers into a bracket. So number 1 is defined...

**num1=parseInt(document.getElementById("num1").value);**

This is telling the browser to check that the variable num1 is a number, and then treat it as a number. By typing this in for both number 1 and 2, as the screenshot above shows, this should make the sum work – and add the numbers together, rather than writing them in one after the other.

Save and publish your code. You should now get something that asks for two numbers and then adds them together! Good work if you did. If you didn't, debug your code and try again. Take a copy of your code and the published page and enter them into your Student Log.

