

# The Do-While Loop

You'll now learn about the do-while loop which is another way to repeat code.

We have a few more loops to go. The **do-while** loop executes the loop body **at least once**, and checks the condition before the end.

```
let numbers = [19, 65, 9, 17, 4, 1, 2, 6, 1, 9, 9, 2, 1];

function sumArray( values ) {
  if ( values.length == 0 ) return 0;
  let sum = 0;
  let i = 0;
  do {
    sum += values[i];
    i += 1;
  } while ( i < values.length );
  console.log( 'The loop was executed ' + i + ' times' );
  return sum;
}

sumArray( numbers );
```



Notice the first line in the function. We have to exit the **sumArray** function before reaching the **do-while** loop in case the array is empty. Otherwise, the code works in the exact same way as the while loop, except that the condition for running the loop again is at the bottom. If the condition is true, then it will repeat the loop as long as the condition is true.

## Do-while for input validation

We mainly use do-while loops for input validation. The user inputs a value, then we check its validity.



Let's use the **prompt** function to get a value.

**prompt( 'message' )** opens a dialog displaying message. A textfield appears in

the dialog, where you can enter a value. Once you press the OK button, the `prompt` function returns a string.

Example:

Output
JavaScript
HTML
CSS (SCSS)
<div>Run Code</div>



This is where the do-while loop makes sense, because we can be sure we need to enter data at least once. In the example summing array members, using the do-while loop is technically possible, but it does not make sense, because the simple while loop is easier.