

JavaScript Function Invocation

The code inside a JavaScript **function** will execute when "something" invokes it.

Invoking a JavaScript Function

The code inside a function is not executed when the function is **defined**.

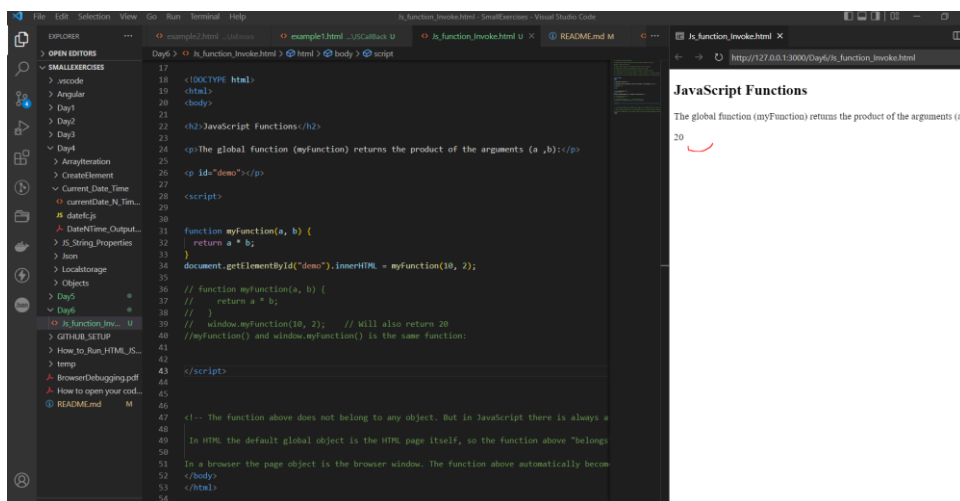
The code inside a function is executed when the function is **invoked**.

It is common to use the term "**call a function**" instead of "**invoke a function**".

It is also common to say "call upon a function", "start a function", or "execute a function".

In this tutorial, we will use **invoke**, because a JavaScript function can be invoked without being called.

Example1:



```
17 <!DOCTYPE html>
18 <html>
19 <body>
20 <body>
21
22 <h2>JavaScript Functions</h2>
23
24 <p>The global function (myFunction) returns the product of the arguments (a,b)</p>
25
26 <p id="demo"></p>
27
28 <script>
29
30
31 function myfunction(a, b) {
32   return a * b;
33 }
34 document.getElementById("demo").innerHTML = myfunction(10, 2);
35
36 // function myfunction(a, b) {
37 //   return a * b;
38 // }
39 // window.myfunction(10, 2); // Will also return 20
40 // myfunction() and window.myfunction() is the same function.
41
42 </script>
43
44
45
46
47 <!-- The function above does not belong to any object. But in JavaScript there is always a
48
49 In HTML the default global object is the HTML page itself, so the function above "belongs
50
51 In a browser the page object is the browser window. The function above automatically becom
52 </body>
53 </html>
54
```