JAVASCRIPT FUNCTIONS

CIT 260:03 – MOBILE APPLICATION DEVELOPMENT

FUNCTIONS

It is a group of reusable code which can be called anywhere in our program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes. Functions allow a programmer to divide a big program into a number of small and manageable functions.

FUNCTIONS

Like any other advanced programming language, JavaScript also supports all the features necessary to write modular code using functions. We must have seen functions like **alert()** and **write()** in the earlier chapters. We were using these functions again and again, but they had been written in core JavaScript only once.

FUNCTIONS

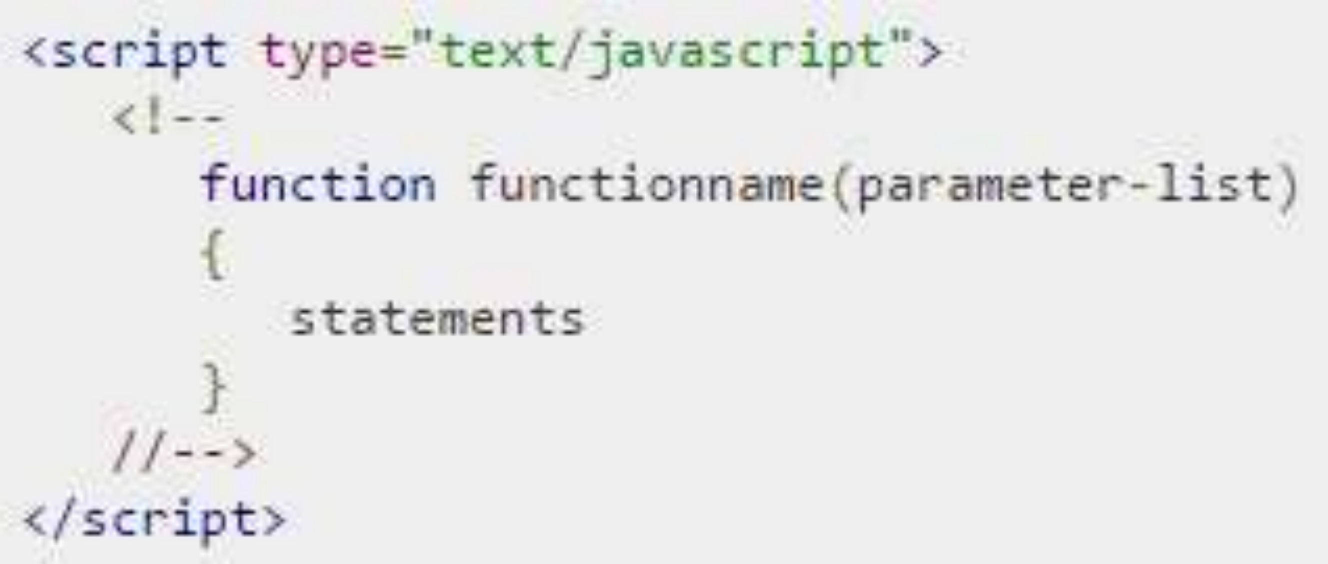
JavaScript allows us to write our own functions as well.

DEFINITION

Before we use a function, we need to define it. The most common way to define a function in JavaScript is by using the **function** keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

SYNTAX

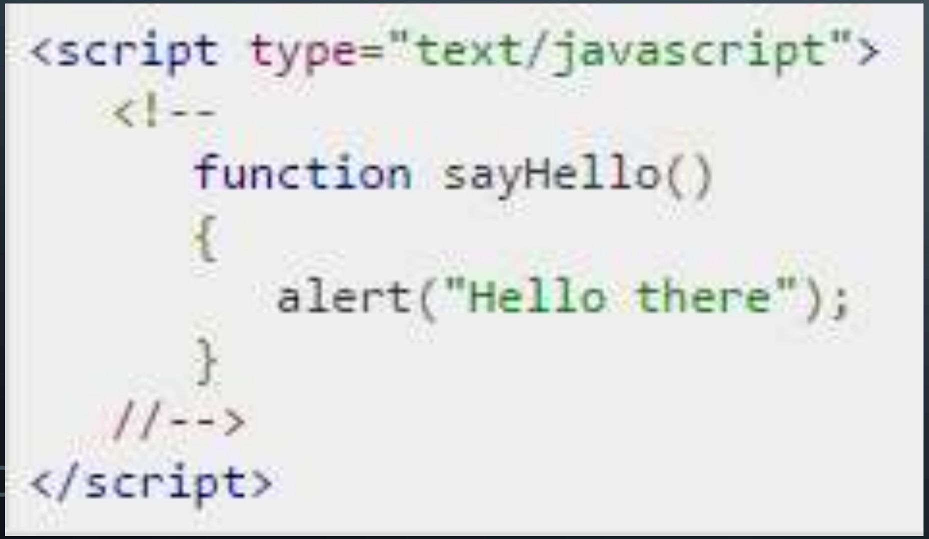
The basic syntax is shown below.



```
<script type="text/javascript">
   <!--
      function functionname(parameter-list)
      {
         statements
      }
   //-->
</script>
```

EXAMPLE

The following exampled defines a function called sayHello that takes no parameters.

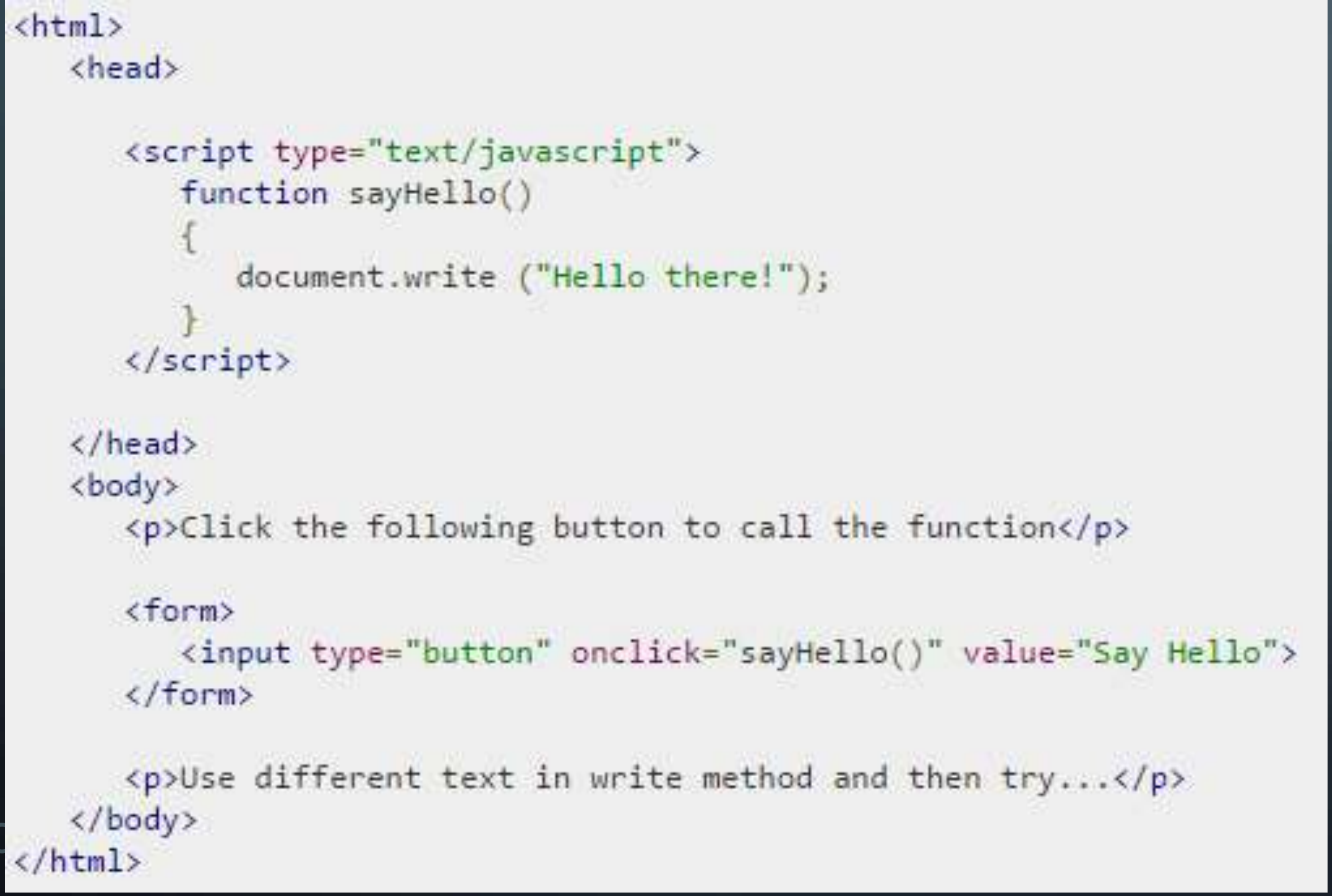


```
<script type="text/javascript">
   <!--
      function sayHello()
      {
         alert("Hello there");
      }
   //-->
</script>
```

CALLING A FUNCTION

To invoke a function somewhere later in the script, we would simply need to write the name of that function as shown in the following code.

CALLING A FUNCTION



```
<html>
   <head>

      <script type="text/javascript">
         function sayHello()
         {
            document.write ("Hello there!");
         }
      </script>

   </head>
   <body>
      <p>Click the following button to call the function</p>

      <form>
         <input type="button" onclick="sayHello()" value="Say Hello">
      </form>

      <p>Use different text in write method and then try...</p>
   </body>
</html>
```

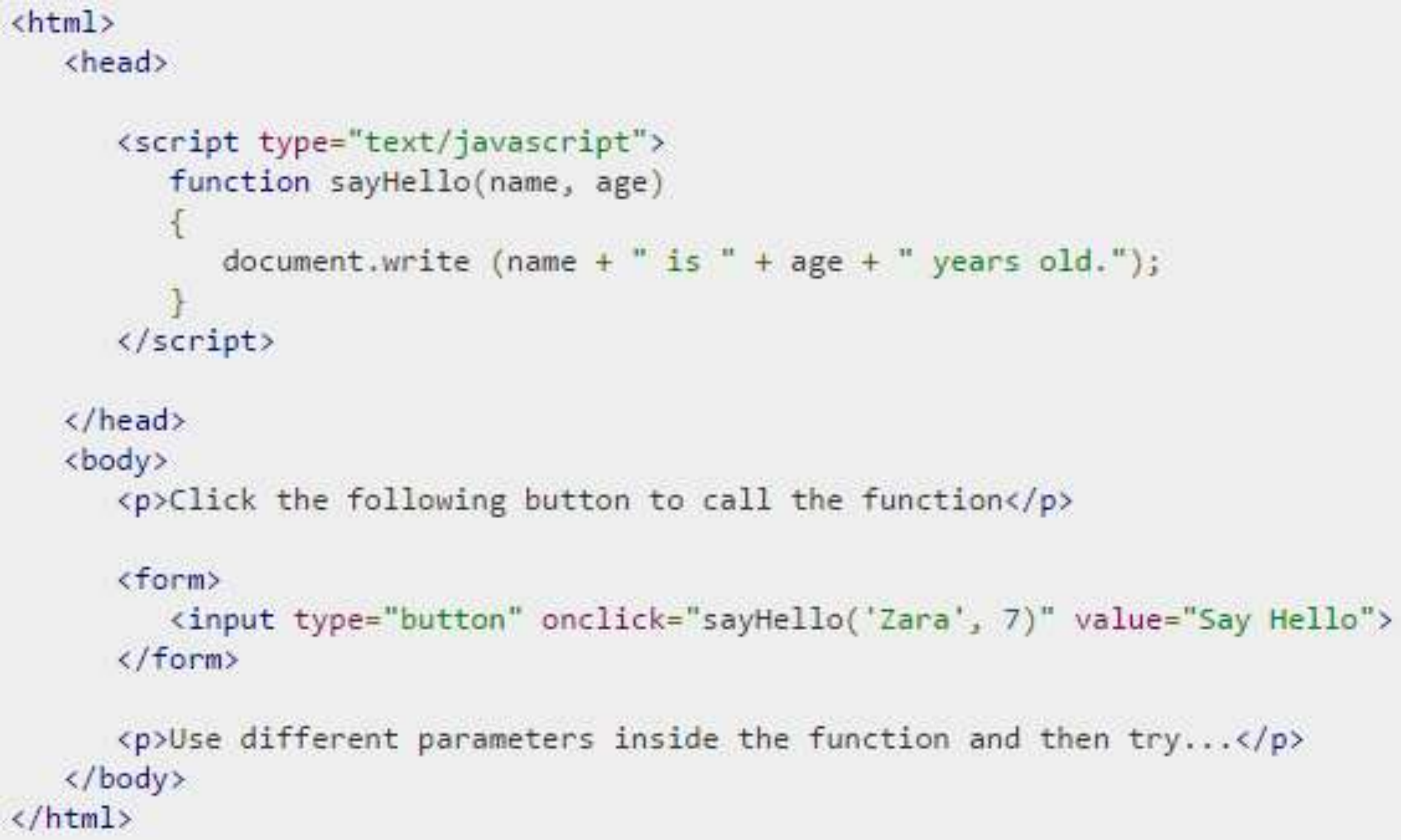
PARAMETERS

Till now, we have seen functions without parameters. But there is a facility to pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters.

PARAMETERS

A function can take multiple parameters separated by comma.

EXAMPLE



```
<html>
   <head>

      <script type="text/javascript">
         function sayHello(name, age)
         {
            document.write (name + " is " + age + " years old.");
         }
      </script>

   </head>
   <body>
      <p>Click the following button to call the function</p>

      <form>
         <input type="button" onclick="sayHello('Zara', 7)" value="Say Hello">
      </form>

      <p>Use different parameters inside the function and then try...</p>
   </body>
</html>
```

THE RETURN STATEMENT

A JavaScript function can have an optional **return** statement. This is required if we want to return a value from a function. This statement should be the last statement in a function.

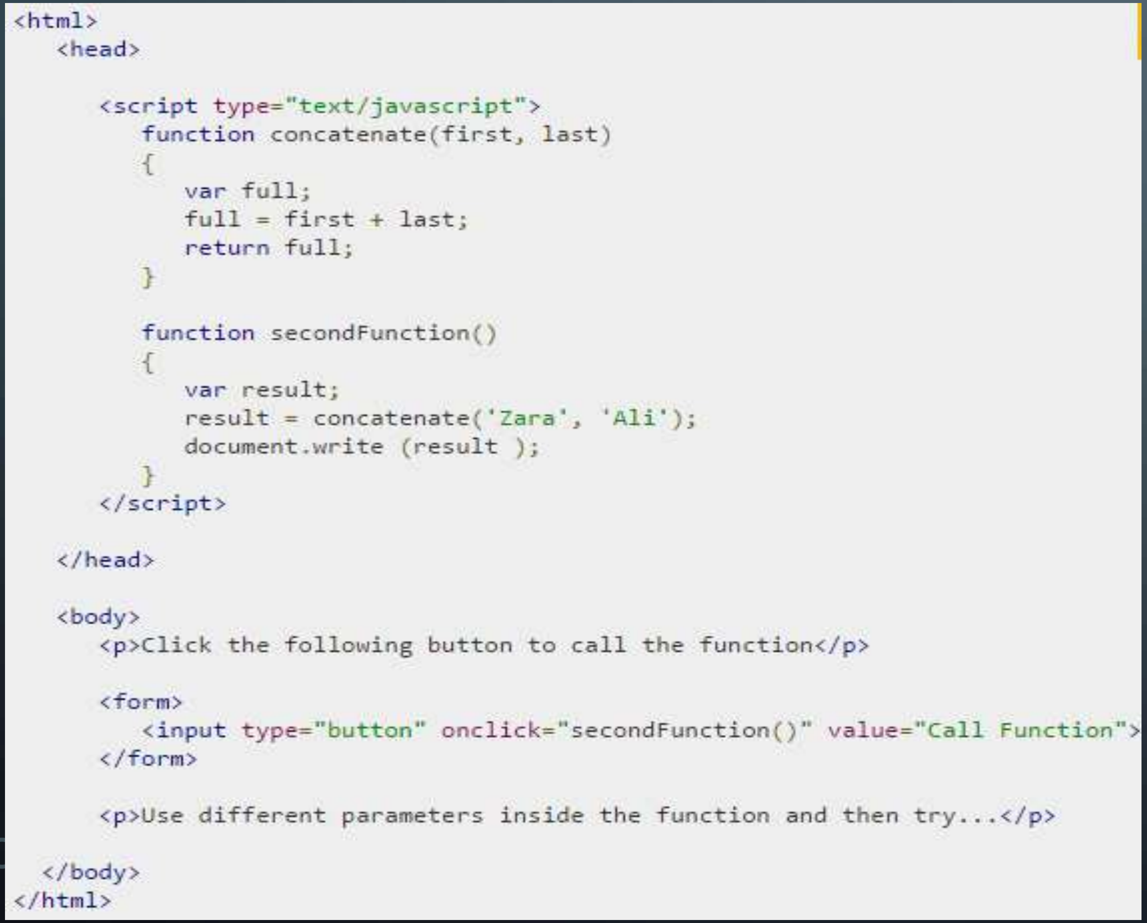
THE RETURN STATEMENT

For example, we can pass two numbers in a function and then we can expect the function to return their multiplication in your calling program.

EXAMPLE

The following example defines a function that takes two parameters and concatenates them before returning the resultant in the calling program.

EXAMPLE



```
<html>
   <head>

      <script type="text/javascript">
         function concatenate(first, last)
         {
            var full;
            full = first + last;
            return full;
         }

         function secondFunction()
         {
            var result;
            result = concatenate('Zara', 'Ali');
            document.write (result );
         }
      </script>

   </head>

   <body>
      <p>Click the following button to call the function</p>

      <form>
         <input type="button" onclick="secondFunction()" value="Call Function">
      </form>

      <p>Use different parameters inside the function and then try...</p>

  </body>
</html>
```

JAVASCRIPT FUNCTIONS

SOURCE: JAVASCRIPT FUNCTIONS:

HTTPS://WWW.TUTORIALSPOINT.COM//JAVASCRIPT/JAVASCRIPT_FUNCTIONS.HTM