

Computing GCSE Coursework

Thomas Bass
Candidate 4869
Centre 52423
OCR A452 Practical Investigation

Made with L^AT_EX

2017

Summary

1 Task 1	3
1.1 Explain how you ran this script:	3
1.2 Explain what each line of the script does:	4
1.2.1 The HTML Code:	4
1.2.2 Commentry:	4
2 Task 2	5
2.1 Set up an array to include the items shown above, plus a few extras of your choice.	5
2.1.1 Products:	5
2.1.2 Array:	5
2.1.3 Array in code editor:	5
2.2 Write a script that:	5
2.2.1 Outputs the items in alphabetical order	5
2.2.2 Counts the number of items in the array.	6
2.2.3 Full code:	6
2.2.4 Output:	7

1 Task 1

Often, a web designer wants a change to happen when a user clicks on a screen object or moves the mouse over it. JavaScript can make changes to the HTML elements.

Enter and run this script:

```
<!DOCTYPE html>
<html>
<body>
<h1>Change an HTML element</h1>
<p id="msg">Now you see me.</p>
<button type="button"
onclick=document.getElementById( 'msg' ).innerHTML = 'Gone!' ">
Click Me!</button>
<button type="button"
onclick=document.getElementById( 'msg' ).innerHTML = 'Back again!' ">
Bring me back!</button>
</body>
</html>
```

1.1 Explain how you ran this script:

This script was copied into a HTML document, and opened into a web browser.
When the script ran it gave the following output:

Change an HTML element

Now you see me.

The web page took the code entered, ran it, and produced this output.

1.2 Explain what each line of the script does:

1.2.1 The HTML Code:

```
01      <!DOCTYPE html>
02      <html>
03      <body>
04      <h1>Change an HTML element</h1>
05      <p id="msg">Now you see me.</p>
06      <button type="button"
07      onclick="document.getElementById('msg').innerHTML = 'Gone!'" >
08      Click Me!</button>
09      <button type="button"
10      onclick="document.getElementById('msg').innerHTML = 'Back again!'" >
11      Bring me back!</button>
12      </body>
13      </html>
```

1.2.2 Commentary:

```
01      This declares that the document is a HTML document
02      This opens the <html> tag, and shows that the code enclosed
      is HTML code
03      This opens the <body> tag, and shows that the code enclosed
      is placed in the body of the document
04      This opens the <h1> tag, showing that the text enclosed
      ("Change an HTML Element") is placed in the highest header, and closes
      the tag
05      This opens a <p> tag, showing that the text enclosed ("Now you see me.")
      is paragraph text, and it has the ID "msg", and then the tag is closed
06      This opens a <button> tag, showing that the information enclosed is
      a button, and has type "button"
07      This declares that following an onclick event (when the button is clicked),
      the program will execute a Javascript function to find the elements with
      the ID of "msg" (the body text in line 05) and change its HTML code to "Gone".
08      This line provides the text of the button ("Click Me!") and closes the
      <button> tag
09      This opens a <button> tag, showing that the information enclosed is
      a button, and has type "button"
10      This declares that following an onclick event (when the button is clicked),
      the program will execute a Javascript function to find the elements with
      the ID of "msg" (the body text in line 05) and change its
      HTML code to "Back Again!".
11      This line provides the text of the button ("Bring me back!") and closes the
      <button> tag
12      This closes the <body> tag
13      This closes the <body> tag and ends the document
```

2 Task 2

As is the case with most programming languages, in JavaScript you can use arrays in order to store multiple values under the same identifier. For example, an array of products can be set up as below for use on an ecommerce web site.

```
var products = ["Printer","Tablet","Router"];
```

2.1 Set up an array to include the items shown above, plus a few extras of your choice.

2.1.1 Products:

Printer, Tablet, Router, Network Switch, Monitor, Keyboard, Mouse, 500GB Hard Drive, ATX Motherboard, Memory Card, Flash Drive, Network Switch, Bluetooth Adaptor, Modem, Wireless Speaker, 256GB SSD.

2.1.2 Array:

```
var products = ["Printer","Tablet","Router","Network Switch","Monitor",  
"Keyboard","Mouse","500GB Hard Drive","ATX Motherboard","Memory Card",  
"Flash Drive","Network Switch","Bluetooth Adaptor","Modem",  
"Wireless Speaker", "256GB SSD"];
```

=16 items

2.1.3 Array in code editor:

```
26 var products = ["Printer","Tablet","Router","Network Switch","Monitor",  
27 "Keyboard","Mouse","500GB Hard Drive","ATX Motherboard","Memory Card",  
28 "Flash Drive","Network Switch","Bluetooth Adaptor","Modem","Wireless Speaker", "256GB SSD"];
```

2.2 Write a script that:

2.2.1 Outputs the items in alphabetical order

```
25 <script>  
26 var products = ["Printer","Tablet","Router","Network Switch","Monitor",  
27 "Keyboard","Mouse","500GB Hard Drive","ATX Motherboard","Memory Card",  
28 "Flash Drive","Network Switch","Bluetooth Adaptor","Modem","Wireless Speaker", "256GB SSD"];  
29 function sortProducts(){  
30     temp = products.sort();  
31     document.write(temp.join('<br/>'));  
32 }  
33 sortProducts();  
34 </script>
```

This code snippet is written in JS embedded in an HTML document. Line 25 opens the `<script>` tag, which indicates to the HTML code that the following is JS script. Lines 26 to 28 then declares `products` to be a global array with the contents as described in part 2.1. Line 29 declares `sortProducts()` to be a function, and opens the code for the function. Line 30 declares `temp` to be the array `products`, sorted alphabetically with the `.sort` function built into JS. Line 31 then outputs the `temp` array, joined by a `
` line break, onto the document body with the `.write` function. Line 32 closes the code for the `sortProducts()` function, and Line 33 calls the `sortProducts()` function its self. Line 34 then closes the `<script>` tag, indicating the end of this JS script.

2.2.2 Counts the number of items in the array.

```
36 <script>
37     function printLength(){
38         document.write(products.length);
39     }
40     printLength()
41 </script>
```

This code snippet is again written in JS, and embedded in the same HTML document as in 2.2.1. Line 36 opens the `<script>` tag to declare the following as JS code. Line 37 then declares `printLength()` to be a function, and opens the code for the function. Line 38 outputs the length of the array with the built-in `.length` function, and writes it to the body with `.write`. The code is then closed in line 39, and the `printLength()` function is called in line 40. Line 41 then closes the `<script>` tag to show the end of the JS script.

2.2.3 Full code:

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width">
6 <title>JS Bin</title>
7 </head>
8 <body>
9 <style>
10     body{
11         background-color: #ecec;
12     }
13     h3 {
14         font-family: arial;
15         color: #3581df;
16     }
17     body {
18         font-family: arial;
19         color: #39ccd2;
20     }
21 </style>
22
23 <h3>Alphabetical:</h3>
24
25 <script>
26     var products = ["Printer","Tablet","Router","Network Switch","Monitor",
27         "Keyboard","Mouse","500GB Hard Drive","ATX Motherboard","Memory Card",
28         "Flash Drive","Network Switch","Bluetooth Adaptor","Modem","Wireless Speaker", "256GB SSD"];
29     function sortProducts(){
30         temp = products.sort();
31         document.write(temp.join('<br/>'));
32     }
33     sortProducts();
34 </script>
35 <h3>Length of Array:</h3>
36 <script>
37     function printLength(){
38         document.write(products.length);
39     }
40     printLength()
41 </script>
42 </body>
43 </html>
```

2.2.4 Output:**Alphabetical:**

256GB SSD
500GB Hard Drive
ATX Motherboard
Bluetooth Adaptor
Flash Drive
Keyboard
Memory Card
Modem
Monitor
Mouse
Network Switch
Network Switch
Printer
Router
Tablet
Wireless Speaker

Length of Array:

16