JavaScript

Tutor Marked Assignment (TMA)

Introduction

The TMA, which carries 25% of the total marks for this module, requires you build a small JavaScript web application based on the following scenario.

Big Biz are starting a work-based lottery scheme for their employees, and want a application that will allow their employees to check their lottery tickets online. Your job is to build a working prototype of this application.

TMA Specifications

You must complete tasks 1 to 6 below. Read the instructions for each task carefully, until you understand exactly what you are being asked to do. Make sure that you submit your solutions for each task exactly as instructed (e.g. submit the code for each task individually rather than as one single, completed solution).

Task 1. Creating and Linking to a JavaScript File (5 marks)

Create a new JavaScript file named *lottotask1.js* using Notepad++ or a similar editor. Now open the TMA Resources Folder in Moodle. Here you will find an HTML file called *lottotask1.html*. Open this file and create a link to the JavaScript file you have just created (e.g. *lottotask1.js*).

When you have done this, save both files and zip them in a file called *jv_tma_task_1.zip ready* for submission.

Task 2. Variables and Arrays (10 Marks)

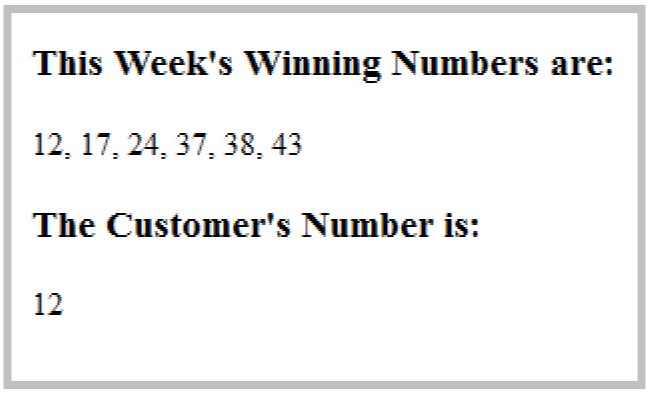
Open your task 1 files, *lottotask1.html* and *lottotask1.js,* in Notepad++ and save them as *lottotask2.html* and *lottotask2.js*. Remember to update the link between the two files.

In *lottotask2.js* create a variable called *customerNumbers* and assign it a value of 12.

Now, create an array called *winningNumbers*. When you have created it, use an array object method to add the following six numbers:

12, 17, 24, 37, 38, 43

Finally, use any method you want to write *customerNumbers* and *winningNumbers* to the screen. Your screen output should look like the following.



```
This Week's Winning Numbers are:

12, 17, 24, 37, 38, 43

The Customer's Number is:

12
```

Add at least one explanatory comment to your code and indent your code where required.

When you have completed the above tasks, save both files and zip them in file called *jv_tma_task_2.zip* for submission.

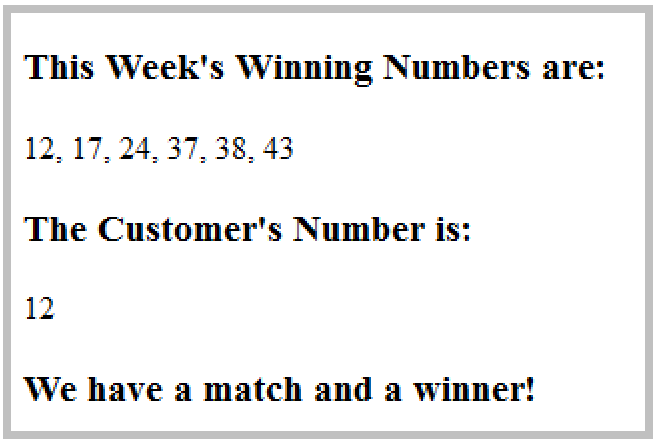
Task 3. Conditionals (18 Marks)

Big Biz's lottery system involves issuing each employee with a single weekly number. If this number is in the weekly series of six winning numbers, the winner recieves a bonus of one hour's pay on top of his/her monthly salary.

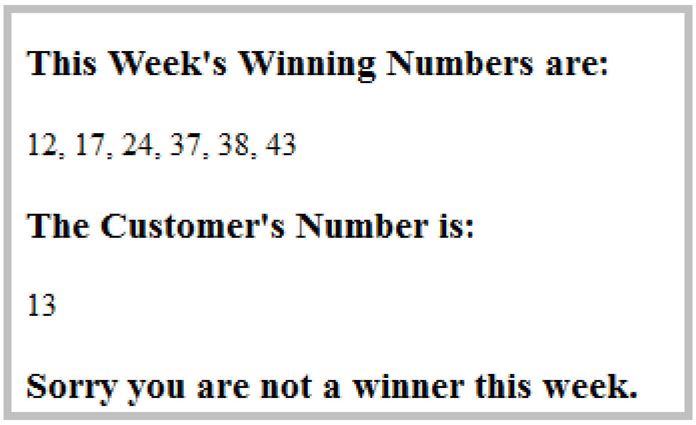
Open your task 2 files, *lottotask2.html* and *lottotask2.js,* in Notepad++ and save them as *lottotask3.html* and *lottotask3.js*. Remember to update the link between the two files.

Now enable your program to search the array of winning numbers to check if a customer number is a winner or not. To do this, you will need to compare the customer number to each of the six winning number to see if there is a match. This should be done using (a) an *if* statement with a series of conditions linked by compound *OR* operators (b) a boolean variable called *match*, which is set to a value of true if the customer's number matches one of the winning numbers.

If a match is found, a message similar to the one below should be written to the screen.


```
This Week's Winning Numbers are:

12, 17, 24, 37, 38, 43

The Customer's Number is:

12

We have a match and a winner!
```

If a match is not found, then a message similar to the one below should be written.


```
This Week's Winning Numbers are:

12, 17, 24, 37, 38, 43

The Customer's Number is:

13

Sorry you are not a winner this week.
```

Add at least one explanatory comment to your code and indent your code where required.

When you have completed the above tasks, save both files and zip them in a zip file called *jv_tma_task_3.zip* for submission.

Task 4. Loops (20 Marks)

Open your task 3 files, *lottotask3.html* and *lottotask3.js,* in Notepad++ and save them as *lottotask4.html* and *lottotask4.js*. Remember to update the link between the two files.

An *if* statement with a series of compound *OR* conditions is one way to look for a number match. However, it is rather long-winded. A more succinct way to achieve the same result would be to use a loop to step through each position in the winning numbers array to check whether the customer number matches any of the winning numbers.

Use a For loop to step through each position in the winning numbers array and to compare the customer number to each number the array contains.

To complete this, you will need to set up the following.

1. A counter variable (e.g. i) for the loop.
2. A boolean variable (e.g. *match*) to flag if a match has been found or not.
3. A compound *AND* condition that allows the loop to continue to iterate only if a match is not found, *and*, the end of the array has not been reached.
4. An *if* statement nested inside the For loop which checks the customer number against each winning number in the array, each time the loop iterates, and sets the boolean, *match*, to true if a match is found.

The output of the program when you are done should be exactly the same as the output indicated in task 3.

Add at least one explanatory comment to your code and indent your code where required.

When you have completed the above tasks, save both files and zip them in a zip file called *jv_tma_task_4.zip* for submission.

Task 5. Functions (22 Marks)

Open your task 4 files, *lottotask4.html* and *lottotask4.js,* in Notepad++ and save them as *lottotask5.html* and *lottotask5.js*. Remember to update the link between the two files.

Now update your program so that the work of checking the lottery result is done by a function called *checkNumbers()*. This function should take the customer number and the array of winning numbers as parameters.

To complete this part of the program you will need to give careful thought to the scope of your variables (e.g. Whether a particular variable should be local or global in scope). Remember, you should always aim to limit the visibility of your variables, keeping them local in scope where possible.

Add at least one explanatory comment to your code and indent your code where required.

When you have completed the above tasks, save both files and zip them (.zip format) in a zip file called *jv_tma_task_5.zip* for submission.

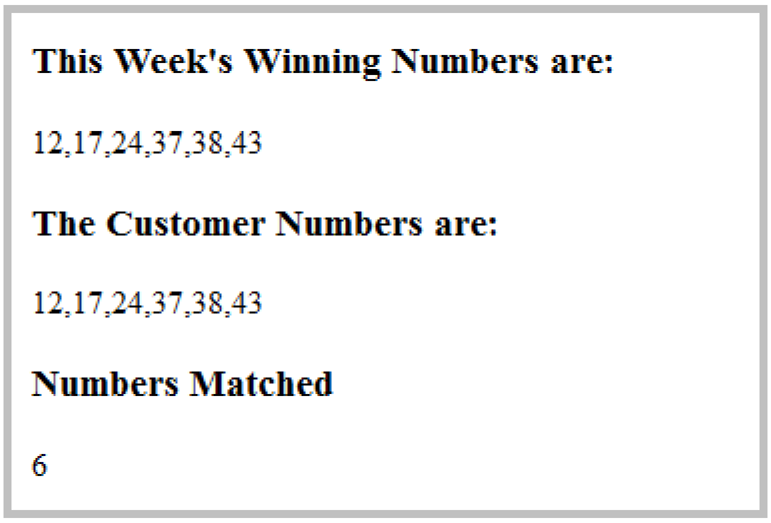
Task 6. Advanced Programming Techniques (25 Marks)

Open your task 5 files, *lottotask5.html* and *lottotask5.js,* in Notepad++ and save them as *lottotask6.html* and *lottotask6.js*. Remember to update the link between the two files.

After successfully completing the Big Biz project, you have been asked by the National Loterry of Zedland to provide a solution for their lottery. Their lottery works differently. In fact, it works exactly like the UK National Lottery. A customer chooses six numbers and they are matched against six winning numbers.

Update your function from task 5 so that it compares all six of the customer's numbers to the six winning numbers to see if there are any matches.

The output of the program would look something like the following, if, for example, there were six numbers matched:



This Week's Winning Numbers are:

12,17,24,37,38,43

The Customer Numbers are:

12,17,24,37,38,43

Numbers Matched

6

Add at least one explanatory comment to your code and indent your code where required.

When you have completed the above tasks, save both files and zip them (.zip format) in a zip file called *jv_tma_task_6.zip* for submission.

Deliverables for Assessment

TMA documentation

The completed TMA documentation must be submitted electronically in a .zip file (*username_jv_tma.zip*) in the assignment dropbox in Moodle BEFORE the start of Session 5. Do NOT submit your TMA in other compressed formats (e.g. .rar).

username_jv_tma.zip should contain the following files:

- jv_tma_task_1.zip
- jv_tma_task_2.zip
- jv_tma_task_3.zip
- jv_tma_task_4.zip
- jv_tma_task_5.zip
- jv_tma_task_6.zip

Note: If a required file is not submitted, the examiners will not search for missing files and 0% will be awarded for any missing components.

Completing the TMA

The TMA must be completed and submitted electronically in the assignment dropbox in Moodle BEFORE the start of session 5.

You should work on your TMA after class and during the self-study session scheduled after Session 4. Begin your work early, as the TMA is a substantial task that requires planning and effort to complete satisfactorily. The TMA prepares you for the FMA, so you greatly reduce your risk of a poor overall mark by completing and submitting a TMA.

Getting support

Support for the TMA work will be available in class during Session 4.

Getting feedback

Feedback on the marked TMA can be downloaded from Moodle and will normally be returned to you within 2 weeks of submission. The feedback on your TMA and any issues that arise can be discussed with your tutor in Session 7, or optionally you can make a tutorial appointment for individual support within 2 weeks of the return of your marked TMA.

Backing up files

Always keep a back-up copy of all work submitted for assessment in case of unforeseen submission problems.

Plagiarism

Plagiarism, which is claiming the work of others as your own, is a serious offence and can result in your exclusion from all colleges of the University of London. You should be aware that we use a range of automated tools to spot potential plagiarism in spreadsheets, databases, programme code and text documents. Providing you clearly reference work done by others that you have included in your TMA you will not be penalised.

In the course of completing the assignment, we acknowledge that you will research code from books and from online sources. *Ideas* and *techniques* from these sources may be used in the completion of your own work. HOWEVER, your own work MUST differ significantly from any third-party sources. If it does not, this will constitute plagiarism. You must also clearly reference any third-party sources you have used.

Likewise, we acknowledge that some students will work together collaboratively to solve problems. Again, if you do this, each student's final submission must be markedly different. If your work is not markedly different from another student's work, again, this will consitute plagiarism.