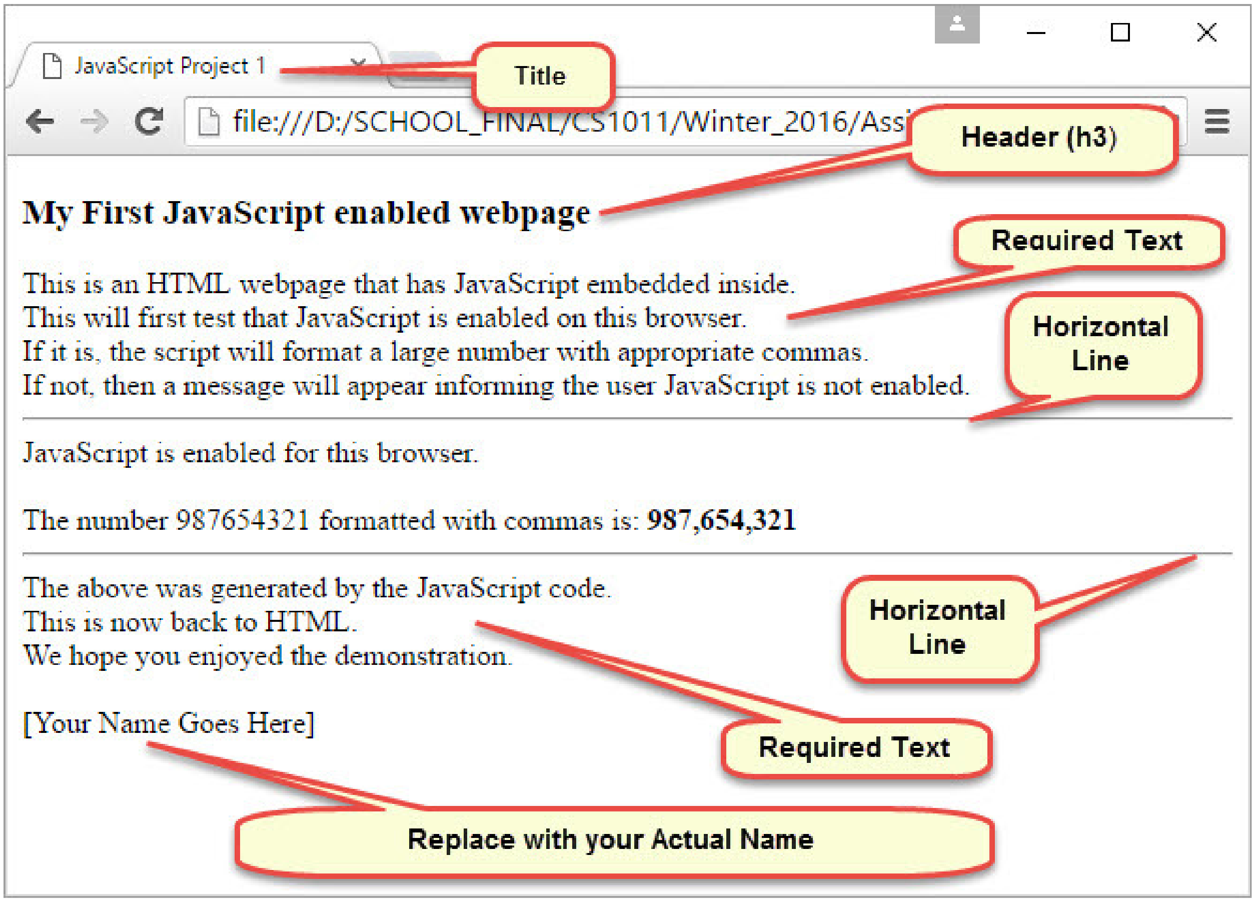
CS 1011b - The Internet: Behind the Curtain
Assignment 3

Instructor: L. Magguilli
Due Date: April 06, 2017, 11:55pm
Total marks: 100
Percentage of final mark: 15%
Accepted up to ONE (1) day late.

Project 1: Use JavaScript imbedded in HTML [15 marks]

Using a plain text editor (example: NotePad or Notepad++ or Brackets), create a web page that contains JavaScript code by typing in the HTML tags and JavaScript commands as shown in the lecture notes and in class.

The web page must look exactly as shown [You **MUST** use these words and spacing]:



My First JavaScript enabled webpage

This is an HTML webpage that has JavaScript embedded inside.
This will first test that JavaScript is enabled on this browser.
If it is, the script will format a large number with appropriate commas.
If not, then a message will appear informing the user JavaScript is not enabled.

JavaScript is enabled for this browser.

The number 987654321 formatted with commas is: **987,654,321**

The above was generated by the JavaScript code.
This is now back to HTML.
We hope you enjoyed the demonstration.

[Your Name Goes Here]

Title

Header (h3)

Required Text

Horizontal Line

Horizontal Line

Required Text

Replace with your Actual Name

Everything above and below the horizontal lines is plain **HTML** code.
Everything between the two lines is **JavaScript**.

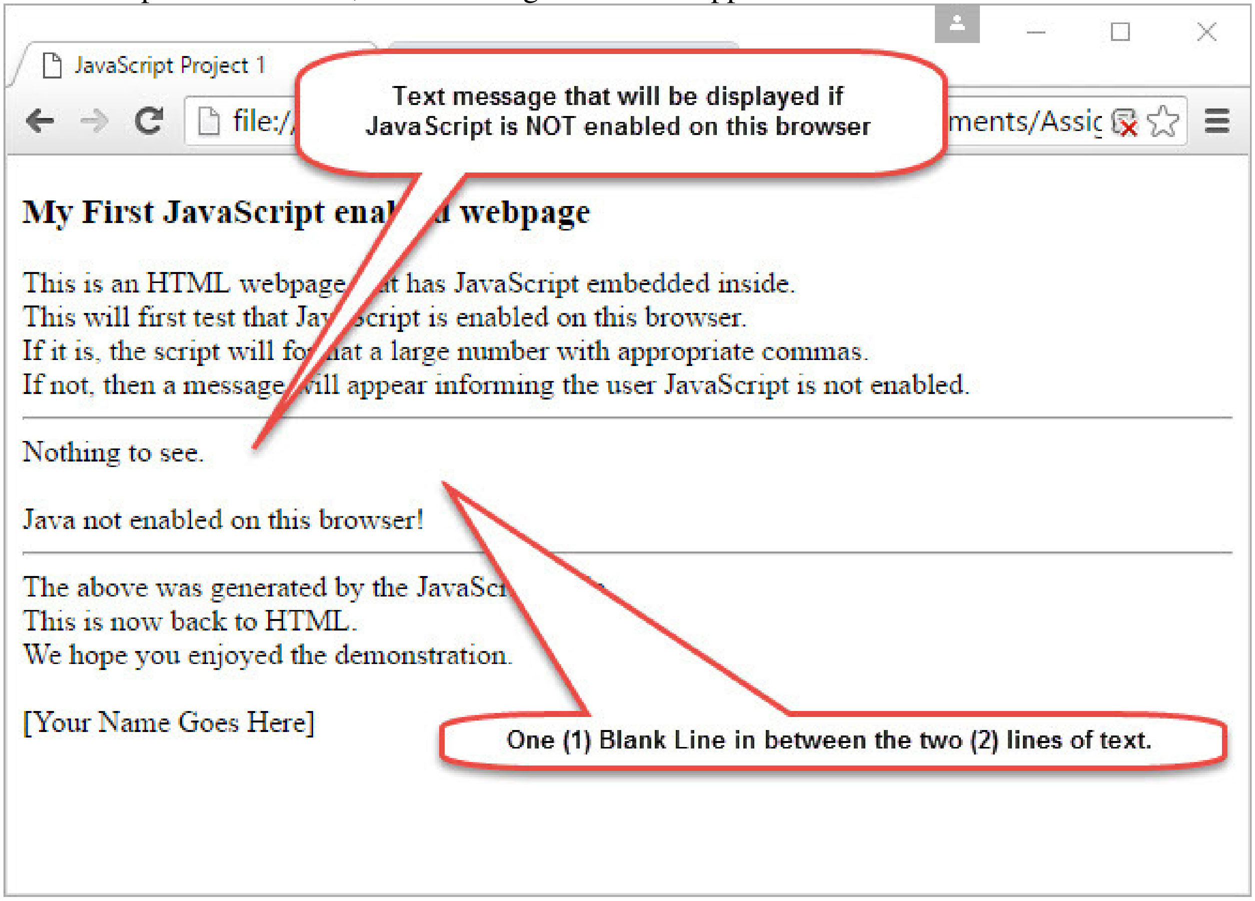
The number to be formatted must be: **987654321**.

You will include the following function in your web page to add the appropriate commas:

```
// This simple method adds commas to numbers (thousands, millions …)
  function formatNumber(num) {
    return num.toString().replace(/(\d)(?=(\d{3})+(?!\d))/g, "$1,")
  }
```

The JavaScript will test if JavaScript is enabled by using the <noscript> tag as shown in class:

If JavaScript is not enabled, the following text should appear:



Text message that will be displayed if JavaScript is NOT enabled on this browser

My First JavaScript ena webpage

This is an HTML webpage t has JavaScript embedded inside.
This will first test that Ja cript is enabled on this browser.
If it is, the script will fo at a large number with appropriate commas.
If not, then a message ill appear informing the user JavaScript is not enabled.

Nothing to see.

Java not enabled on this browser!

The above was generated by the JavaScr
This is now back to HTML.
We hope you enjoyed the demonstration.

[Your Name Goes Here]

One (1) Blank Line in between the two (2) lines of text.

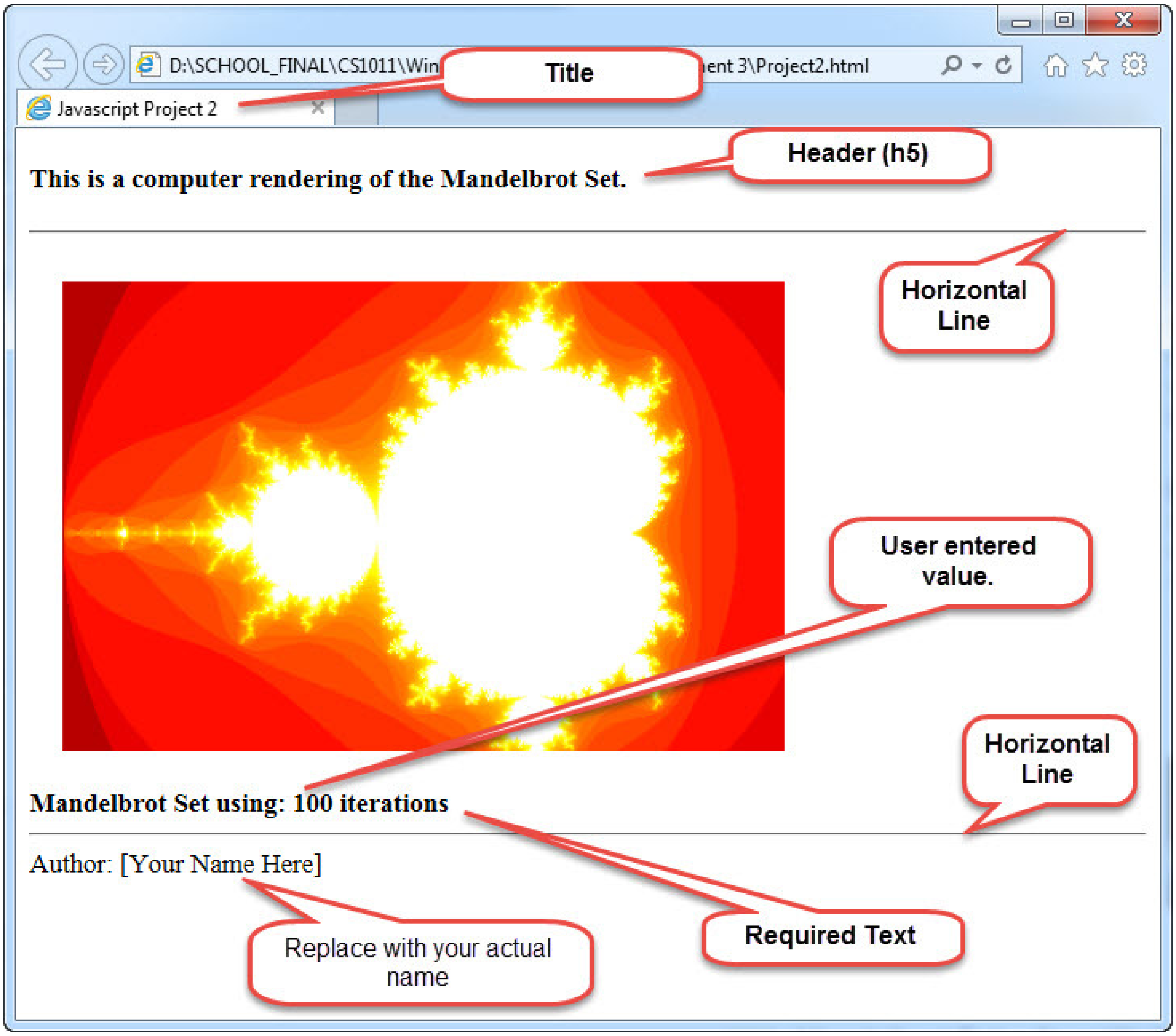
Project 2: Using JavaScript to interact with a user and the use of Function Calls [25 marks]

Using a plain text editor (example: Brackets or Notepad or Brackets), create a web page that contains JavaScript code by typing in the HTML tags and JavaScript commands as shown in the lecture notes.

You will ask the user if they wish to draw the Mandelbrot Set.
If they do, you must prompt for the number of iterations (how fine of detail) they want.

Then you will render the Mandelbrot Set on your webpage.
If they enter a value less than 100, or greater than 5,000 or decline, you will display the fact the user declined on to the webpage.

The rendered webpage must look like this [You **MUST** use these words and spacing]:



This is a computer rendering of the Mandelbrot Set.

Mandelbrot Set using: 100 iterations

Author: [Your Name Here]

Everything above and below the horizontal lines are plain HTML code.
Everything between the two lines is JavaScript.
The JavaScript for the Mandelbrot Set is in a supplied file named Mandelbrot.js

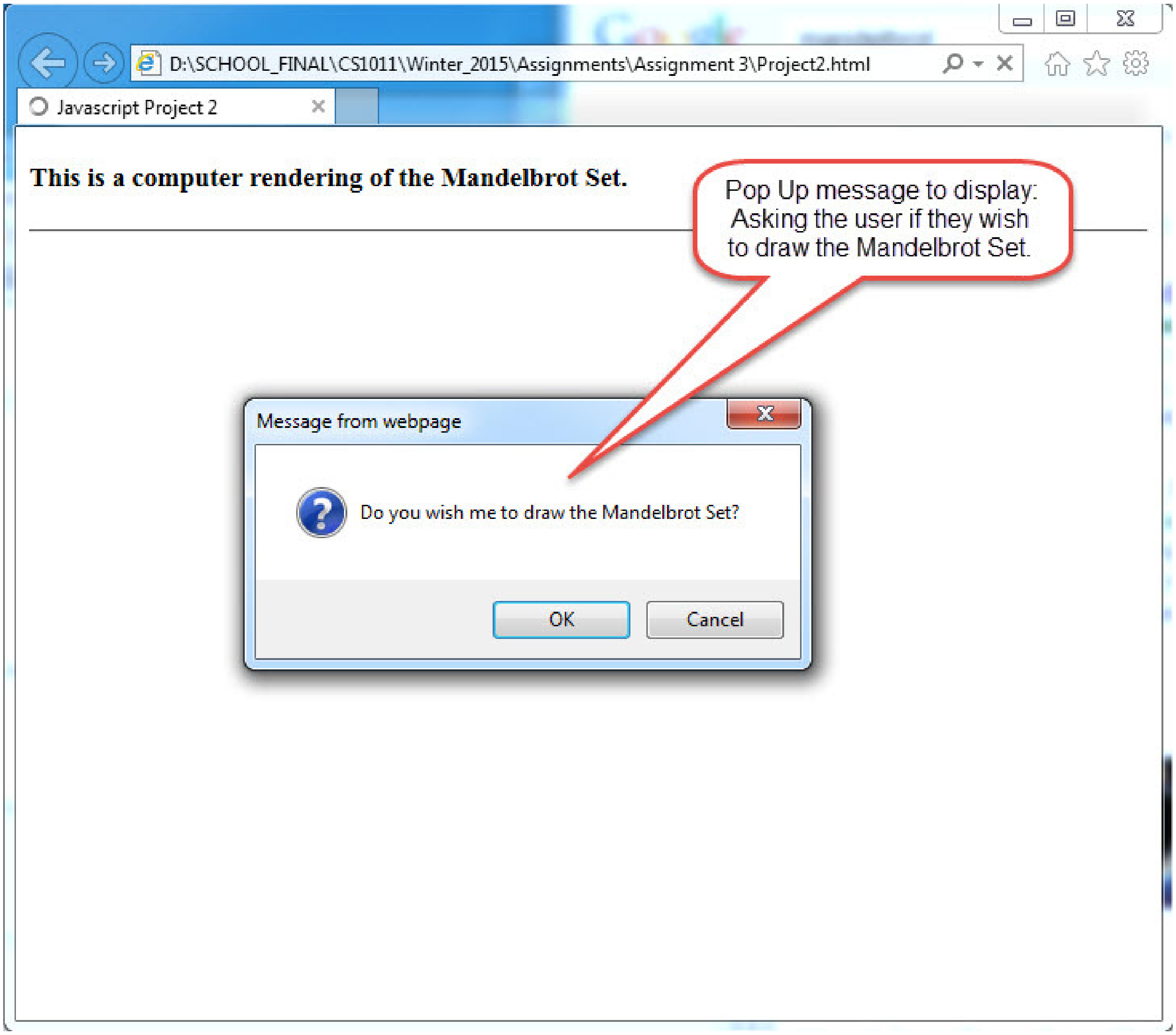
You must set up the connection inside your webpage to this file so the included function can be called from your JavaScript code: [do **NOT** copy the code in to your webpage]

The function is:

```
Mandelbrot( 450,300,  -2,1, -1,1, NumIterations );
```

where `NumIterations` is the value your user enters.
Just copy and paste the line of code above into your code exactly as given here.
Note: the higher the number of iterations – the longer it takes to render the set.

First, you must confirm with the user they want to render the Set.
The confirmation message must look like:



D:\SCHOOL_FINAL\CS1011\Winter_2015\Assignments\Assignment 3\Project2.html

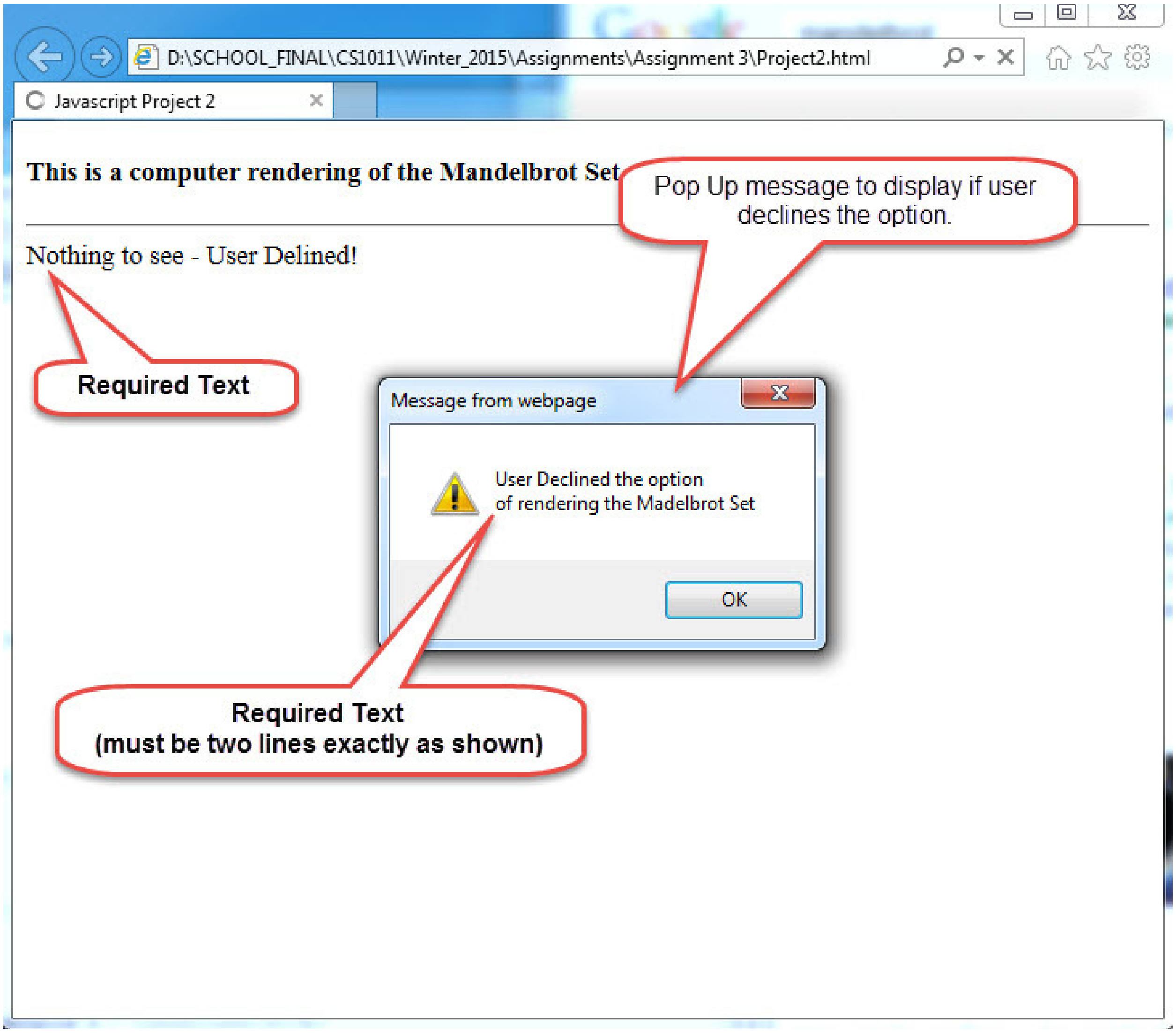
This is a computer rendering of the Mandelbrot Set.

Pop Up message to display: Asking the user if they wish to draw the Mandelbrot Set.

Message from webpage

Do you wish me to draw the Mandelbrot Set?

If they **Cancel** – display the following:



D:\SCHOOL_FINAL\CS1011\Winter_2015\Assignments\Assignment 3\Project2.html
Javascript Project 2
Message from webpage
OK

This is a computer rendering of the Mandelbrot Set

Nothing to see - User Delined!

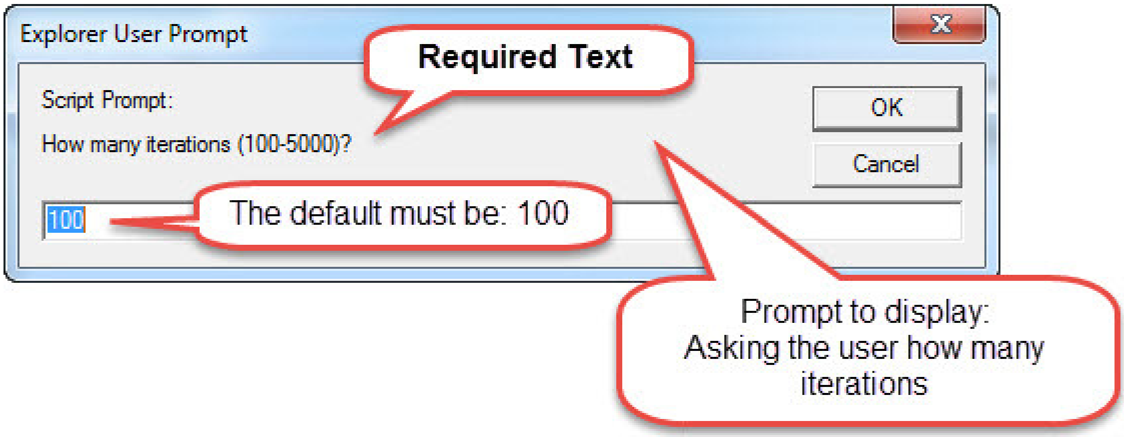
User Declined the option
of rendering the Madelbrot Set

Pop Up message to display if user declines the option.

Required Text

Required Text
(must be two lines exactly as shown)

If they accept, then prompt them for the number if iterations.



Script Prompt:

How many iterations (100-5000)?

100

Required Text

The default must be: 100

Prompt to display:
Asking the user how many iterations

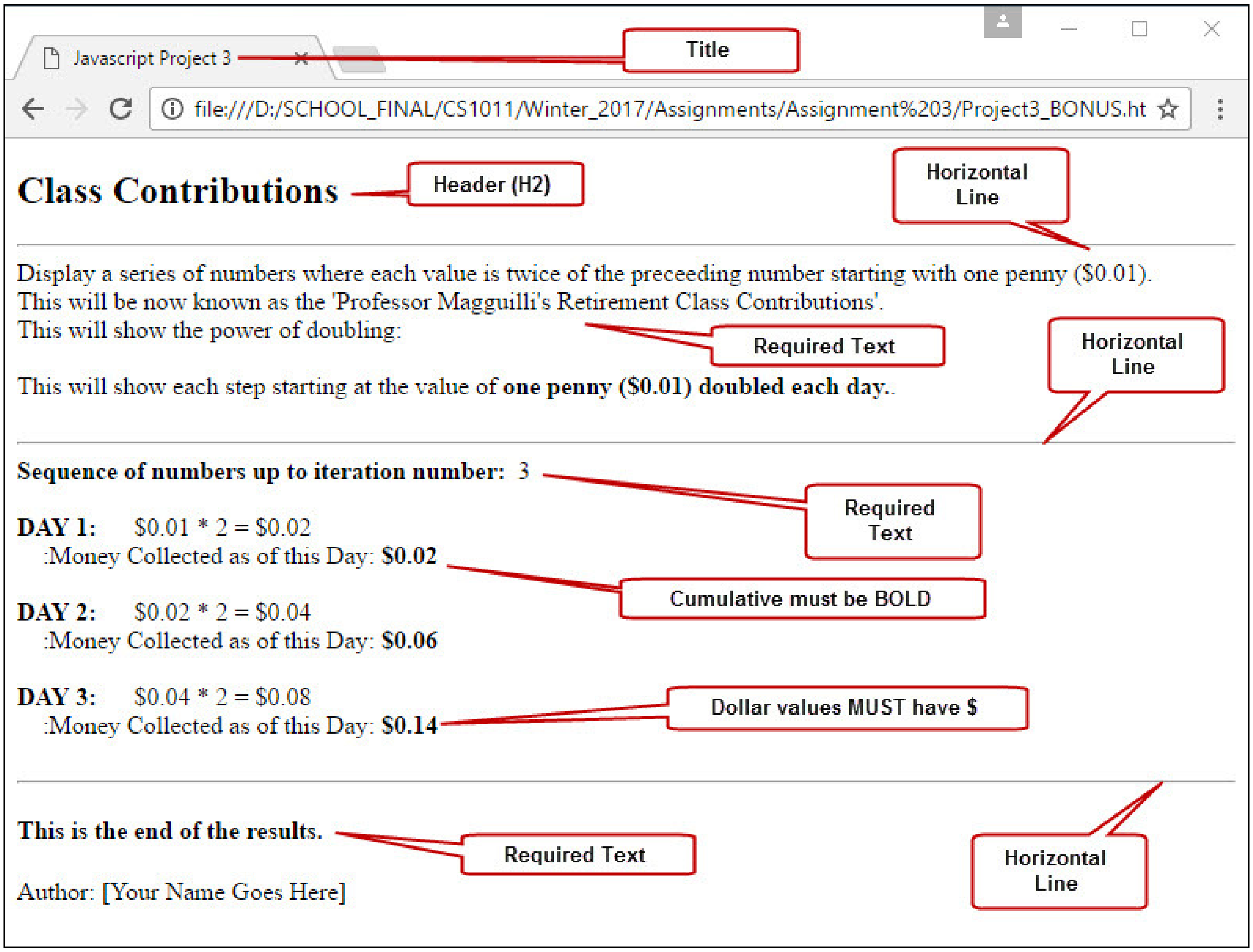
Project 3: Using JavaScript to compute and display the Professor Magguilli Retirement Fund [40 marks]

Prf. Magguilli always talks about the power of doubling values and how quickly it accumulates. One example he frequently uses is if the class starts with a penny and then doubles it every day, how much will he accumulated each day. This means add the result to a running total.

The Magguilli Retirement Fund is a series of numbers based on multiplying the next sequential number by two (double the number at each step).

Simply stated:
 Start with a value A and a running total X that starts with the value of A
 Double this value A (A x 2 = B) then add B to the running total of X
 Then Double B (B x 2 = C) then add C to the running total of X
 Then Double C (C x 2 = D) then add D to the running total of X
 (write code that allows you to do this with just two (2) variables.)

The rendered webpage must look like this [You MUST use these words and spacing]:


Javascript Project 3
file:///D:/SCHOOL_FINAL/CS1011/Winter_2017/Assignments/Assignment%203/Project3_BONUS.ht

Class Contributions

Display a series of numbers where each value is twice of the preceeding number starting with one penny ($0.01).
This will be now known as the 'Professor Magguilli's Retirement Class Contributions'.
This will show the power of doubling:

This will show each step starting at the value of **one penny ($0.01) doubled each day.**.

Sequence of numbers up to iteration number: 3

DAY 1: $0.01 * 2 = $0.02
 :Money Collected as of this Day: **$0.02**

DAY 2: $0.02 * 2 = $0.04
 :Money Collected as of this Day: **$0.06**

DAY 3: $0.04 * 2 = $0.08
 :Money Collected as of this Day: **$0.14**

This is the end of the results.

Author: [Your Name Goes Here]

Title

Header (H2)

Horizontal Line

Required Text

Horizontal Line

Required Text

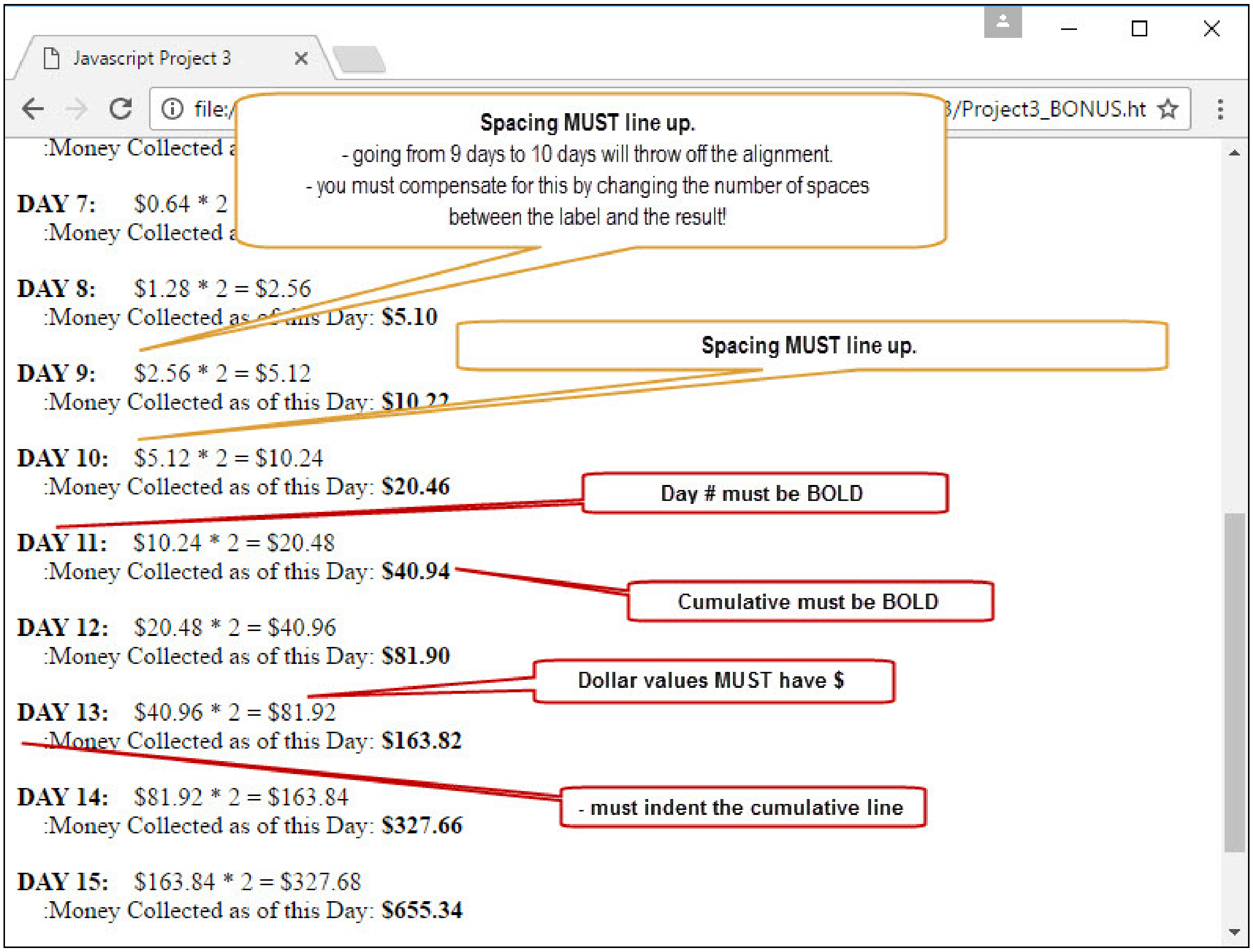
Cumulative must be BOLD

Dollar values MUST have $

Required Text

Horizontal Line

Your results must be formatted using the **FORMATTING INSTRUCTIONS**:



```
   :Money Collected a

DAY 7:     $0.64 * 2
   :Money Collected a

DAY 8:     $1.28 * 2 = $2.56
   :Money Collected as of this Day: $5.10

DAY 9:     $2.56 * 2 = $5.12
   :Money Collected as of this Day: $10.22

DAY 10:    $5.12 * 2 = $10.24
   :Money Collected as of this Day: $20.46

DAY 11:    $10.24 * 2 = $20.48
   :Money Collected as of this Day: $40.94

DAY 12:    $20.48 * 2 = $40.96
   :Money Collected as of this Day: $81.90

DAY 13:    $40.96 * 2 = $81.92
   :Money Collected as of this Day: $163.82

DAY 14:    $81.92 * 2 = $163.84
   :Money Collected as of this Day: $327.66

DAY 15:    $163.84 * 2 = $327.68
   :Money Collected as of this Day: $655.34
```

Spacing MUST line up.
- going from 9 days to 10 days will throw off the alignment.
- you must compensate for this by changing the number of spaces between the label and the result!

Spacing MUST line up.

Day # must be BOLD

Cumulative must be BOLD

Dollar values MUST have $

- must indent the cumulative line

- The Number of Iterations must be displayed before the results (as seen in the first image).

- All equations MUST line up
  - so: as the DAY goes from single digit to two – the spacing must make room

- **DAY Number:** must be in bold – but the equations not bolded (plain)

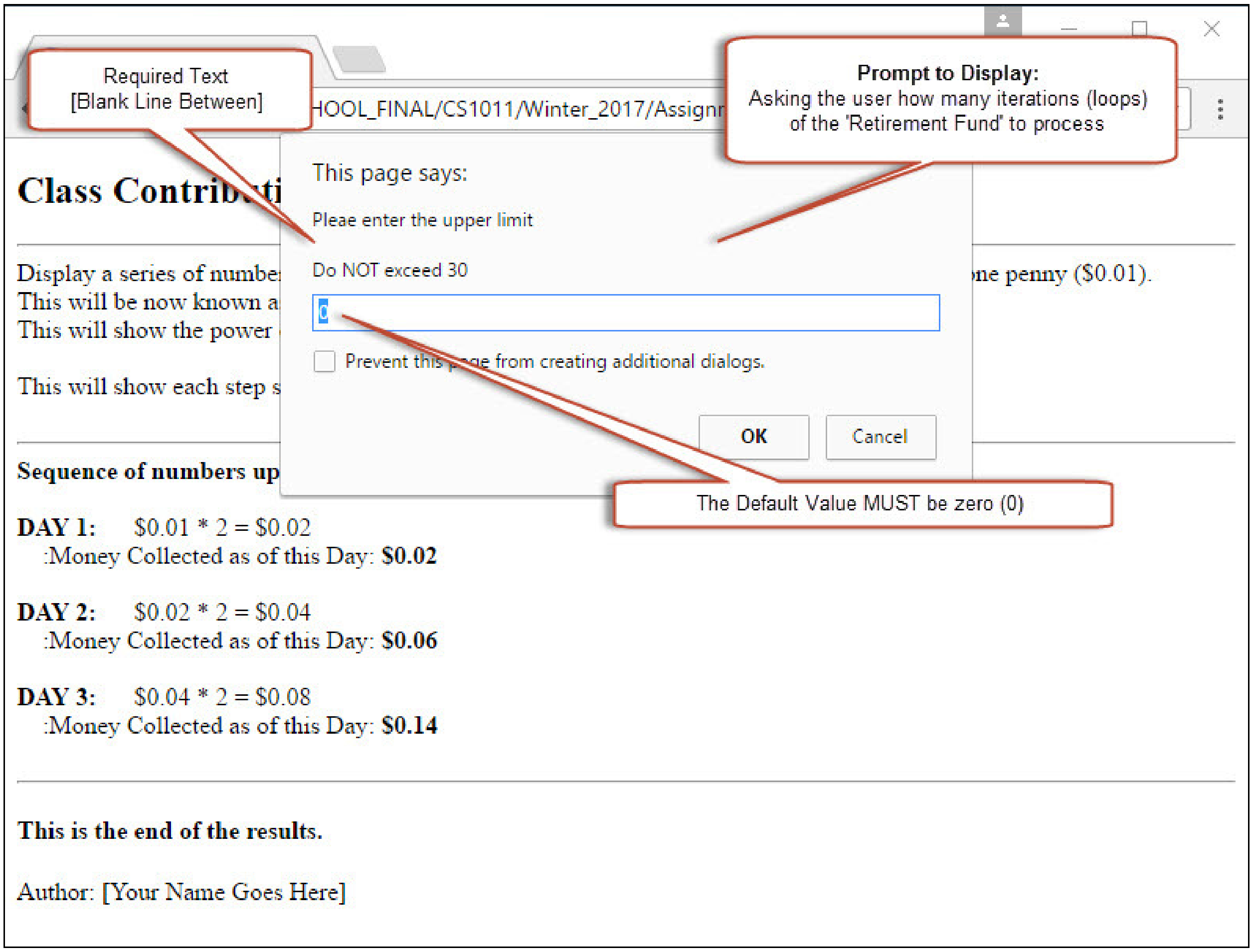
- All numbers must have the appropriate commas inserted.
  - (hint: you have seen this before…)

- The dollar sign ($) is added as part of the text output line.

<u>HINT</u>: to show a number with EXACTLY two (2) decimal places in JavaaScript add **.toFixed(2)**
so –if you name the variable that stores the number is something like **imaVaribale**
then the following will show it with two decimal places: **imaVaribale.toFixed(2)**

PROMPT: You must prompt the user for the number of iterations they wish to see computed:

The prompt must look like:



Required Text
[Blank Line Between]

Prompt to Display:
Asking the user how many iterations (loops) of the 'Retirement Fund' to process

The Default Value MUST be zero (0)

HOOL_FINAL/CS1011/Winter_2017/Assignr

This page says:

Pleae enter the upper limit

Do NOT exceed 30

0

☐ Prevent this page from creating additional dialogs.

Class Contributi

Display a series of numbe ... ne penny ($0.01).
This will be now known a
This will show the power

This will show each step s

Sequence of numbers up

DAY 1: $0.01 * 2 = $0.02
:Money Collected as of this Day: **$0.02**

DAY 2: $0.02 * 2 = $0.04
:Money Collected as of this Day: **$0.06**

DAY 3: $0.04 * 2 = $0.08
:Money Collected as of this Day: **$0.14**

This is the end of the results.

Author: [Your Name Goes Here]

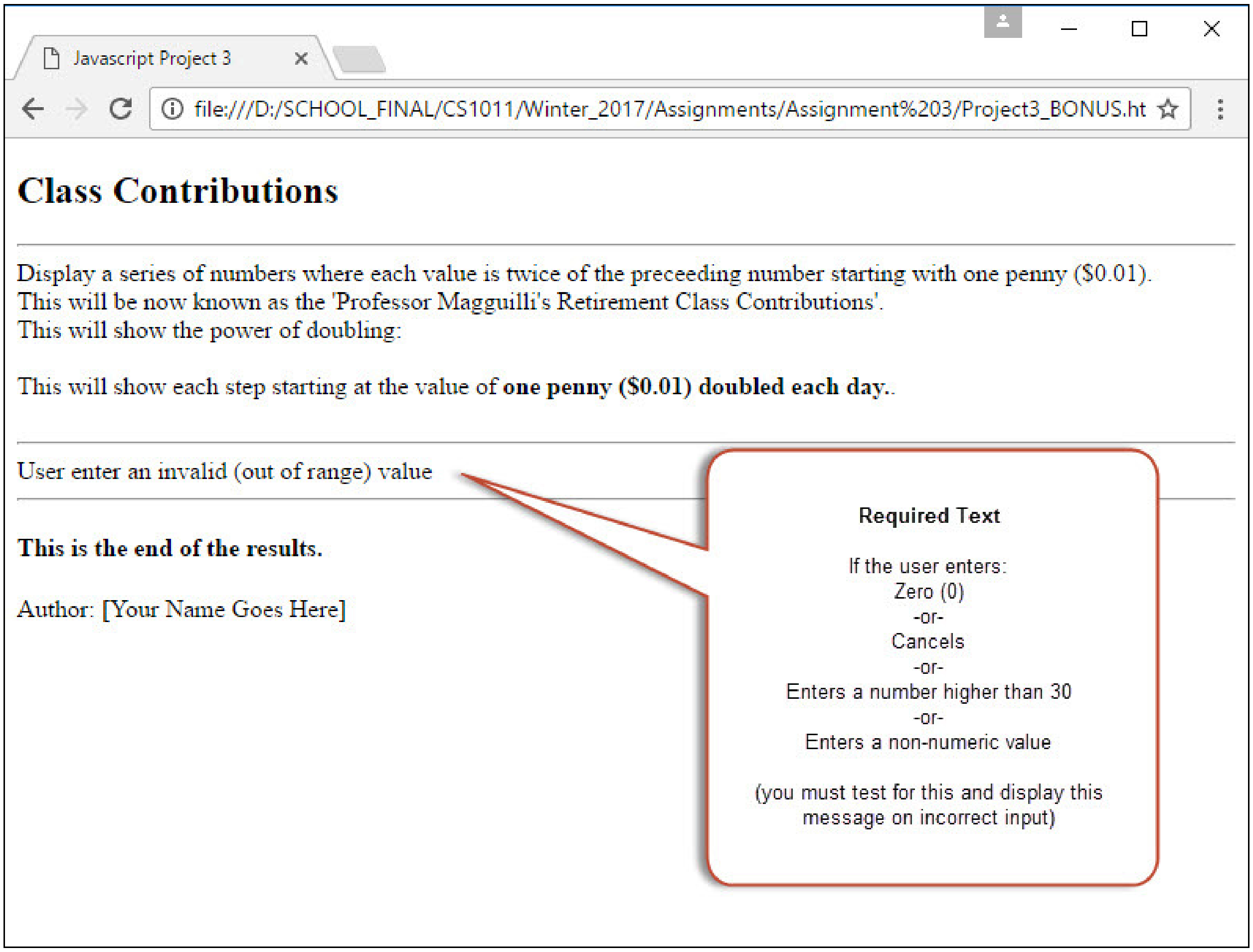
Everything above the second horizontal line and below the third horizontal line is plain HTML code.

Everything between the bottom two lines is JavaScript.

You need to test if the user has entered a zero (0) or a negative number or has Canceled the request or entered a number out of range.

If the user does any of these actions you will simply print this information out on the webpage as seen in the image below.

Example of Screen when user cancels or enters an invalid entry:


Javascript Project 3
file:///D:/SCHOOL_FINAL/CS1011/Winter_2017/Assignments/Assignment%203/Project3_BONUS.ht

Class Contributions

Display a series of numbers where each value is twice of the preceeding number starting with one penny ($0.01).
This will be now known as the 'Professor Magguilli's Retirement Class Contributions'.
This will show the power of doubling:

This will show each step starting at the value of **one penny ($0.01) doubled each day**..

User enter an invalid (out of range) value

This is the end of the results.

Author: [Your Name Goes Here]

Required Text

If the user enters:
Zero (0)
-or-
Cancels
-or-
Enters a number higher than 30
-or-
Enters a non-numeric value

(you must test for this and display this message on incorrect input)

Project 3 BONUS: [10 additional (bonus) marks]

This is an optional addition to the Project Three (3) for bonus marks (added to the assignment grade)

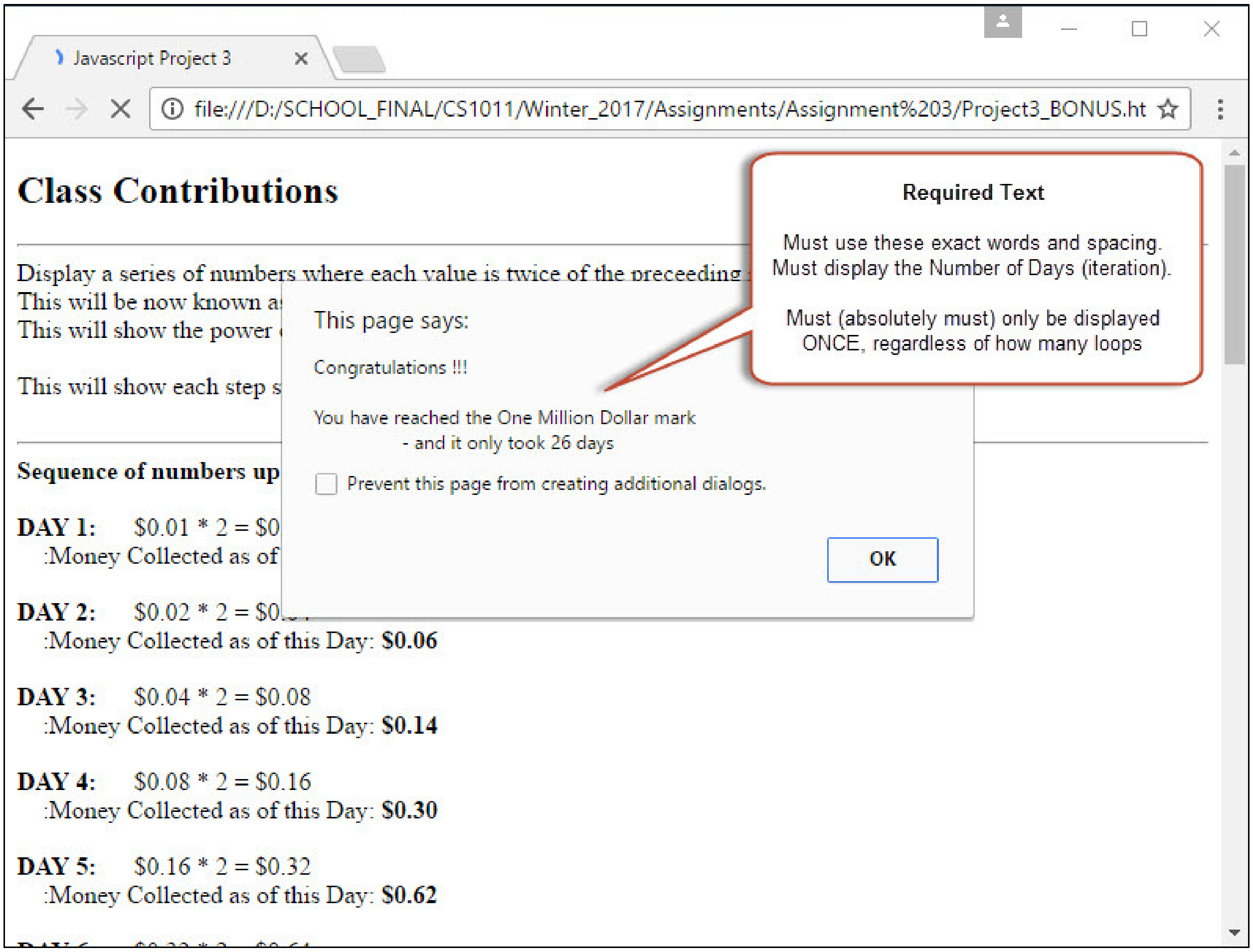
The class wants to know how many days have to pass before $1,000,000.00 is reached.

Add code to the web page that will send an alert message to the user when this milestone is achieved.

BUT!! This message must only appear once, even if more days beyond this point are calculated.

SO, if this goal is reached on day 26, this message should only appear once. It cannot repeat or be shown again. Once and once only!

This is what the **BONUS pop up message** must say, exactly.


Javascript Project 3
file:///D:/SCHOOL_FINAL/CS1011/Winter_2017/Assignments/Assignment%203/Project3_BONUS.ht

Class Contributions

Display a series of numbers where each value is twice of the preceding

This will be now known a

This will show the power

This will show each step s

Sequence of numbers up

DAY 1: $0.01 * 2 = $0

:Money Collected as of

DAY 2: $0.02 * 2 = $0.

:Money Collected as of this Day: **$0.06**

DAY 3: $0.04 * 2 = $0.08

:Money Collected as of this Day: **$0.14**

DAY 4: $0.08 * 2 = $0.16

:Money Collected as of this Day: **$0.30**

DAY 5: $0.16 * 2 = $0.32

:Money Collected as of this Day: **$0.62**

This page says:

Congratulations !!!

You have reached the One Million Dollar mark

- and it only took 26 days

□ Prevent this page from creating additional dialogs.

OK

Required Text

Must use these exact words and spacing.
Must display the Number of Days (iteration).

Must (absolutely must) only be displayed
ONCE, regardless of how many loops

Only show this prompt at when the total accumulated of one million dollars reached.

Do not show any message or prompt if the number of iterations is too low to reach the one million mark.

If you elect to do the bonus do **NOT** submit two different copies of Project Three.

The Teaching Assistants will be able to see you attempted the bonus by looking at the code.

Project 4: Publish your Web Pages [10 marks]

Go to the web site http://publish.uwo.ca/ to get step by step instructions for posting a web page at Western. Complete the instructions provided at this site.

Upload the following files to that site:

- youraccountname_Project1.html
- youraccountname_Project2.html
- youraccountname_Project3.html
- Mandelbrot.js
- Any other files that are required for your pages to operate.

Go the following site: http://publish.uwo.ca/~youraccountname/youraccountname_Project1.html and ensure that your pages are published and work properly

Project 5: Final Exam [10 marks – 2 each]

Write five (5) multiple choice questions you think would be appropriate for the final exam.

Base these questions on material covered in the slides or in class since the midterm.

[The final will not be cumulative]

- Ensure that each multiple choice question has five choices
- Ensure that the majority of questions are framed in a positive context
  (do not use questions that ask "Choose the option that does not belong)
- Avoid using "all of the above" or "none of the above" answer options, or answers consisting of specific option numbers as possible responses.
- Provide, for each question the correct answer

(…and again, if I use your question(s) you will already know the answer(s))

Submission Instructions:

You must upload and submit, via the CS1011 OWL Web Site, the following file:

youraccountname_Assignment3.docx or *youraccountname* _Assignment3.doc
- this will contain your Final Exam questions.
youraccountname_Project1.html
youraccountname_Project2.html
youraccountname_Project3.html
Mandelbrot.js (just to make it easier to mark – the file will be included in your download)
Any other files you wrote (you created or used some other files) so your pages work.

- It is your responsibility to have all the file required for your web pages to function.

(example: my UWO email is dernt373@uwo.ca

so, my file name that is to be used for submission is:
dernt373 _Project1.html
-and-
dernt373 _Project2.html
-and-
dernt373 _Project3.html

NOTE: Please do not cheat.
Do not use another student's work.
We will check, we will be able to verify.
And besides, you will need to know how to write JavaScript for the final.
So, please, do not cheat. Do the work yourself. Do the work independently.