

# Homework 4 - Looping Statements

- Due Oct 7 by 11:59pm
- Points 15
- Submitting a text entry box or a website url
- Available Sep 24 at 5am - Dec 13 at 11:59pm

## Homework 4 - Looping Statements

### What you will do:

- Create a web page that will be the main page for Homework 4 with
  - the heading Homework 4 - Looping Statements
  - a hyperlink to Part 1 - with a short summary about that part
  - a hyperlink to Part 2 - with a short summary about that part
  - a hyperlink to Extra credit (if you will do it) - with a short summary about the extra credit
  - a hyperlink to Homework 1 which is the main page (front page) of your course

**DO NOT USE ALERT BOXES FOR ERROR MESSAGES OR RESULTS!**

### For Part 1:

Create a webpage for Part 1 and it will contain a script, **using the for statement**, that calculates the **product** and **sum** of **every fourth integer from 5 to 25 inclusive** and then displays the results, along with some accompanying text.

For example, when presenting the results, you could show two lines such as:

***The result of  $5 * 9 * 13 * 17 * 21 * 25$  is ....***

***The result of  $5 + 9 + 13 + 17 + 21 + 25$  is ....***

On that same web page for Part 1, you will have another script, now **using the while or do while statement**, that calculates the **product** and **sum** of **every third integer from 3 to 18 inclusive** and then displays the results, along with some accompanying text.

For example, when presenting the results, you could show two lines such as:

***The result of  $3 * 6 * 9 * 12 * 15 * 18$  is ....***

***The result of  $3 + 6 + 9 + 12 + 15 + 18$  is ....***

You should use ***toLocaleString()*** to present the result with the correct punctuation - remember that it is really bad for users to read a big number such as 1342658 instead of reading as 1,342,658 that shows the separation of the thousands.

### For Part 2:

You will create a web page to teach kids about multiplication tables from 5 to 9. There will be 5 tables presented, one below the other, showing the multiplication of the numbers as shown below (showing below just the first table for you to see the pattern):

Number	Multiplier	Result
1	5	5
2	5	10

3	5	15
4	5	20
5	5	25
6	5	30
7	5	35
8	5	40
9	5	45

Notice that the first column of each table, will always shows numbers from 1 to 9.

The second column of each table (multiplier) will show the multiplier being used to build that specific table (in the example it is showing multiplier 5).

The third column is showing the result of the number column multiplied by the multiplier column.

There will be 5 tables in total, one below the other, to show the multipliers from 5 to 9.

The color of the first row of each table (the heading row) will be different from all the other rows of the table. Then, once you start coding the items of the table, every other row will get a light gray color and a white color (as shown in the example above).

As soon as the user opens the web page on the browser, he/she will see a heading such as "**Learn multiplication**"

Below the heading you will code a script to show the 5 tables - 1 table for each multiplier from 5 to 9.

Each of the tables will show the rows **with numbers from 1 to 9 in the column of Number**.

There is **NO INPUT** for this problem! The 5 tables should be shown as soon as the page is opened (loaded in the browser).

**IMPORTANT TIP:** The more efficient way to resolve this problem is to make one loop inside the other (nested loop). It is not ok if you set the 5 tables first and then loop to write only the results on the tables! Think about future maintenance/changes - for example, if the teachers decide to use that same website to show different ranges of multipliers and numbers by giving, for example, input boxes to the teachers or students to put the range they want to see, etc. There are so many possibilities...!!! The less places you have to modify, the better your code is considered. So, use nested loop

**MORE FOR THIS TIP:** the outside loop should be for the **multiplier** that will start a new table tag and the top part of the table (table headings), then the internal loop should be for each **number from 1 to 9** to build the rows of that table!

**TIP FOR BACKGROUND COLORS OF THE ROWS OF THE TABLES:** For alternating the background colors of the rows of each table, you can use CSS - think about **nth-of-child** pseudo-class for the **tr** tag and, in this case, make sure, the first row of each table has the cells marked up as **th**, not **td**!

## Extra Credit

### 2 extra points ONLY IF YOUR SCRIPT WORKS

- Create a webpage with:
  - a heading

- a form with an input box that will allow the user to enter a number (integer), a button to process the script, and a button to clear the form
- a separated div or paragraph element below the form where you will present the result
- You will build a script that:
  - Will read the number typed by the user that should be between 2 and 10
  - Based on that number, you will display a hollow square with that size constructed of asterisks.
  - If the user entered an invalid number (not a number or outside the range of 2 to 10), you should present an error message in the div or paragraph element below the form and you should NOT present the square!
  - If the user entered a valid number, you should present the hollow square in the div or paragraph element below the form

### TIPS:

- Be sure that the output looks like a square as a common error is having the top and bottom row of asterisks appear to be smaller than the sides.
- Use nested while or for statements and follow the final note below when presenting the asterisks to compose the sides of the square

### Final Note:

There should be **NO ASTERISKS (\*)** inside the square - it should be a hollow square like the image below (notice that there is a space between each \* printed asterisk at the top and the bottom of the square which you do not need to provide when printing the asterisks of the side of the square as the browser already provides a space between each line):

```
* * * * *
*           *
*           *
*           *
*           *
*           *
* * * * *
```

### To submit your homework:

1. Create the main page for **Homework 4** as stated at the top
2. Create the pages for **Part 1**, **Part 2**, and **Extra Credit** (if doing it), **make sure you remember to insert a link to the main page of Homework 4 from each of those pages so the user does not need to use the back button of the browser to see your work!**
3. Modify the **main page of the course - Homework 1** - to include a **hyperlink to Homework 4 main page**
4. Upload all the files to a web server
5. Test the URL of your **Homework 1** web page and make sure that the hyperlink to **Homework 4** is working correctly and showing the Homework 4 main page. Then, from **Homework 4 main page**, I should be able to find the hyperlinks to each part of the assignment and be able to open those pages too
6. If everything is ok, submit the **URL of the Homework 1 page** - from there I should use the hyperlink to Homework 4 to start evaluating your work!

## Checklist for grading:

1. Your pages are shown in the browser with no NOT FOUND or FORBIDDEN errors and you updated Homework 1 to have the correct link to this homework and the URL you submitted is the Homework 1 URL - **1 point**
2. No input in any of the pages (exception will be for the extra credit that is evaluated separately) - **1 point**
3. The scripts of the pages are working correctly with what was requested for each page, including the necessary validations requested - **12 points** (6 points for each part)
4. The HTML and CSS of all pages are valid - **1 point**

**The extra credit point will be added to the points when I grade the assignment. If you submit late, then the extra credit will be added and then deductions will be applied according to the syllabus.**