

Assignment 02

ITEC 3020, Summer 2021

Due Time: Monday - July 26 – 11:55 pm

Submission

You may work individually or in pairs.

- If you are submitting as a pair, **only one of the members** should submit the assignment. Please, include the name of the teammates in the submission (in a different text (.txt) file). Otherwise, if no information is received, then an individual assignment will be assumed.
- To submit your solutions to this assignment, you will use the submission facility on the course website on eClass.
- Submission should be a zipped folder including a text file with the names of the teammates and four folders for four exercises of this assignment. Each folder includes one html file, one or more css files, and one or more js files.
- You are allowed to use frameworks like Bootstrap, SemanticUI, etc but you are **not allowed to use templates**.
- You do not need to use Ajax for this assignment.
- You should abide by the policies set by the Senate of York University and by the Faculty of Liberal Arts & Professional Studies regarding the Academic Honesty.
- This assignment is worth **20% of your final grade**.

Exercise One: Design and implement a web page to receive a number and map it to a letter grade based on York standard. You may need to look at this reference: <http://calendars.registrar.yorku.ca/2012-2013/academic/grades/index.htm>. Assume if the grade is between 40 and 49, it's mapped to an E.

1. In this exercise you need at least three files (ex1.html, ex1.css, ex1.js).
2. You need an input html tag with type 'text', because this program receives a number and a proper html tag to show the result.
3. Use proper styles for your html document.

Exercise Two: Implement a web page to receive a number and continue receiving numbers until a zero is entered. When a zero is entered, the program should output how many positive and how many negative numbers have been entered, and then stop.

1. In this exercise you need at least three files (ex2.html, ex2.css, ex2.js).
2. You need an input tag with type 'text', because this program receives a number and a proper html tag to show the result.
3. Use proper styles for your html document.

Exercise Three: Implement a program to receive a positive number and output "yes" if it's equal to its reverse; otherwise, output "no". For instance, if the input is 63936, the program should output "yes", because if you read the digits from left to right or from right to left, it's the same number. But, if the input is 632, the program should output "no" because 632 is not the same as 236.

You may separate the digits of the input number and make the reverse of the input number along the way—i.e., in loop iterations. For instance, assume the input number is 235, before entering the loop, you should declare a variable 'reverse' and initialized it to 0 as well as storing the input number in a variable, let's call it temp.

In the first iteration of the loop, you will have:

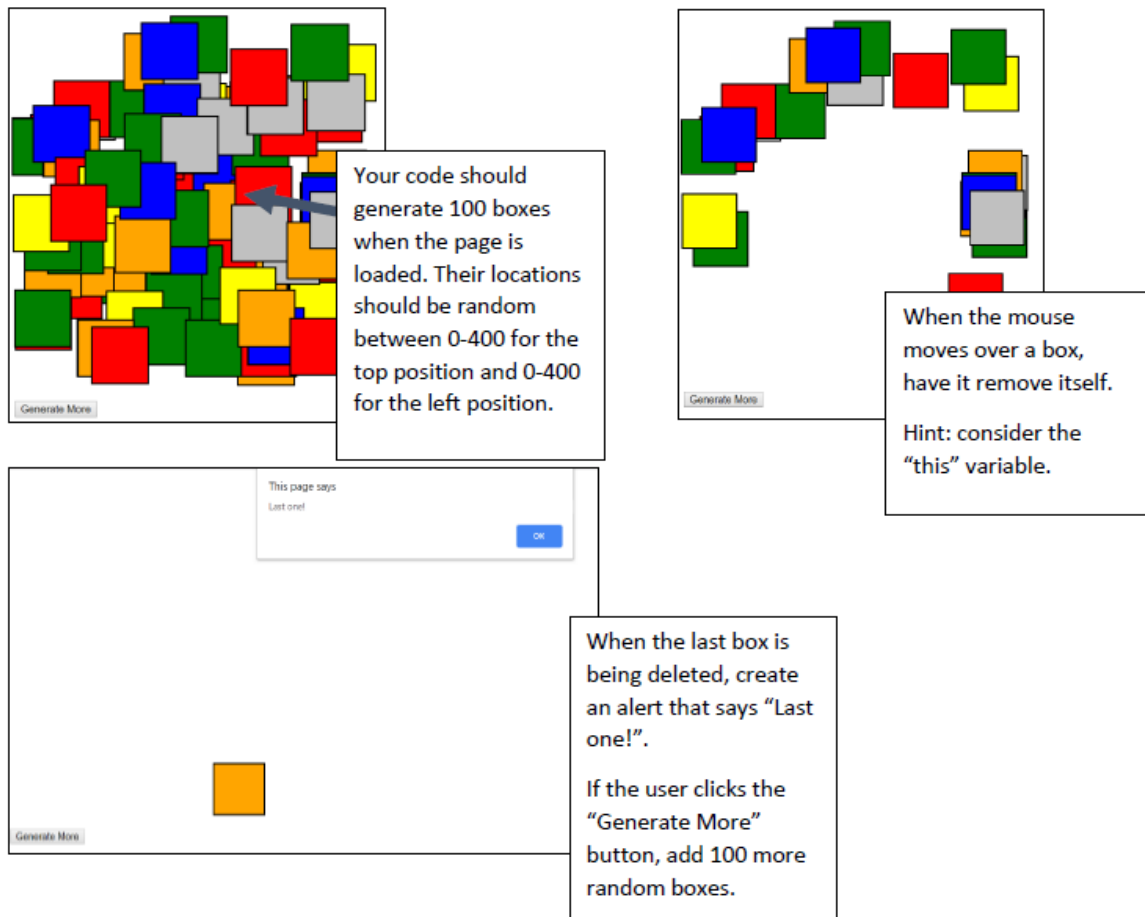
235 modulus 10 is 5; and, the reverse number that you currently have, which is 0, times 10 plus 5 is 5 and you store it in variable reverse; then, you reduce 235 to 23 by dividing it by 10.

Now, in the second iteration of the loop, you will have: 23 modulus 10 is 3; and, the reverse number that you currently have, which is 5, times 10 plus 3 is 53 and you store it in variable reverse; then, you reduce 23 to 2 by dividing it by 10. Now, in the third iteration of the loop, you will have:

2 modulus 10 is 2; and, the reverse number that you currently have, which is 53, times 10 plus 2 is 532 and you store it in variable reverse; then, you reduce 2 to 0 by dividing it by 10.

Because the number is already reduced to zero, there is no more iteration. In other words, we exit the loop. Now, we should compare the initial number (235) by its reverse (532)—that we built within loop iterations. If they are equal, we output "yes"; otherwise we output "no".

Exercise Four: Create 100 boxes at random locations inside a div with the id container (see below). When the mouse moves over a box, the box should be removed from the div. When the div has one child box, pop up an alert that says, "last child!". Add a button to create more boxes. If the user clicks on the button, your JS code should generate 100 more boxes (even if you have existing boxes on the screen).



Pay attention that:

- Your html pages should be dynamic. Use DOM and event handling to design and implement your programs.
- In creating your programs, you might need to use more than what we explained in class. You can search to learn more about them, W3schools and tutorialspoint are good resources.
- Proper implementation of the html tags and their styles is important.
- Examples we saw during the lectures about HTML documents and JavaScript are quite useful for this assignment.