Lab 3: Undo-Redo

Due: October 18, 2020 at 11:59:59 PM

Introduction

This past week, we've been doing Javascript. These past couple years, you've hopefully been paying attention to programming classes, specifically COEN 12. One of the first data structures you encountered was a Stack. A stack is a last-in, first-out data structure. The last thing that's put in is the first thing that comes out.

One practical application of a stack is undo-ing and redo-ing things. This lab is legitimately based on an intern project that was run at Twitch. Oh! That's why you pay attention in Lab!

Submission

Due Date: October 18, 2020
Demo: Yes
Format: 1 compressed file prefixed with your SCU Email with the following files:

- index.html
- index.css
- index.js
- stack.js
- questions.txt

Goals

1. Applying data structures in practical applications
2. Using Javascript to interact with a web page

Task 1: Read Code!

Read through the code in the provided files. There are 6 subtasks (Task 1A -Task1 F) in there for you to do. For anything that you don't understand in the HTML and CSS, see if the TAs can help.

Task 2: The Grid

The grid is already made for you in HTML. It has three rows, each of which are Flexboxes. And in each of the rows, there are 3 buttons. When you click a button in the grid, that specific button should change to a random color (don't worry they're already provided for you, but you can add more).

1. Attach an event handler onto each button (there's a function provided for you already). Don't do this individually, find a way to do this with a loop! Be lazy and smart :D
2. For each button click, change the backgroundColor of that button to a random color from the COLORS array.

Some things to look into:

1. document.querySelector()
2. Javascript Arrays
3. Math.random() (but that'll give you a decimal, not an integer)

Task 3A: Add Elements to the Undo Stack

Once you've gotten those buttons working, augment your event handler to add a new HTMLDivElement to the Undo Stack any time you click a button on the grid. The new element should have the backgroundColor of the button BEFORE you clicked it. Add the position of the button to the Div.

For example, if you click a grey button with Position 0, the undo stack should have a Div with 0 in the middle but the Grid Button with Position 0 should be a different color.

Some things to look into:

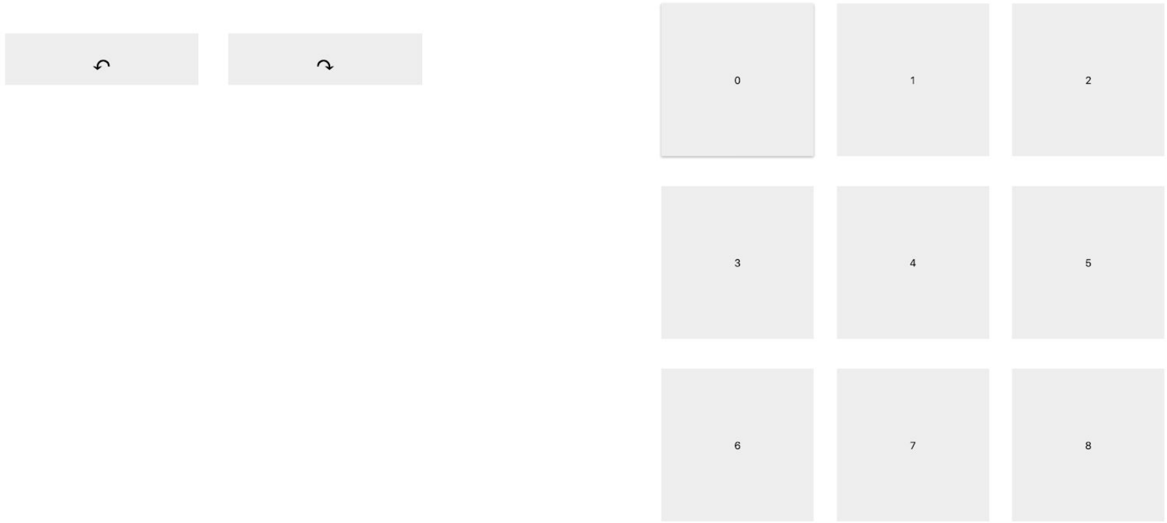
1. Node.textContent
2. Style property in JS
3. document.createElement

Task 3B: Undo and Redo

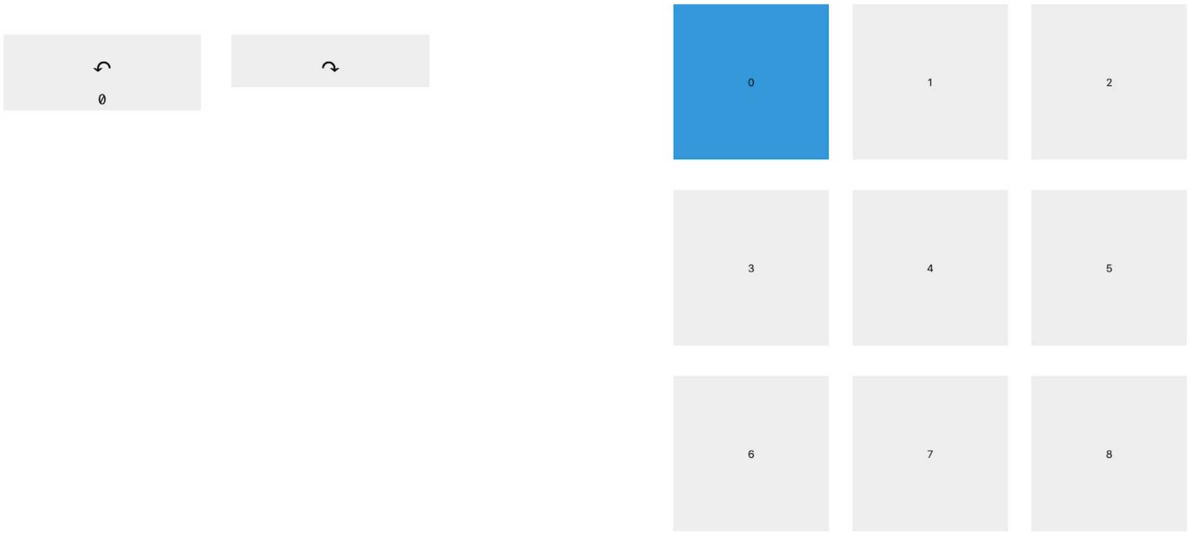
The last part of this lab is to implement undo and redo functionality. This won't be full undo and redo functionality (although feel free to implement that). The limited version of undo/redo will just move things between the two stacks but won't clear the redo stack when a new action is done.

Every time you click a grid button, push some information to undo the transformation you just applied. It's clearer in pictures than words:

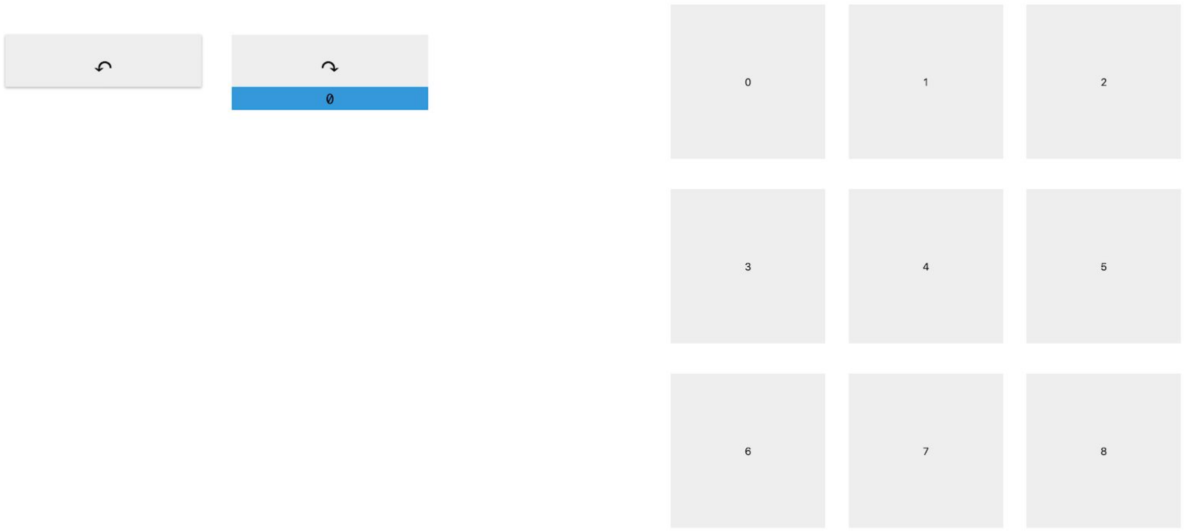
Step 1: The Grid is the The Grid. We haven't done anything to it yet.



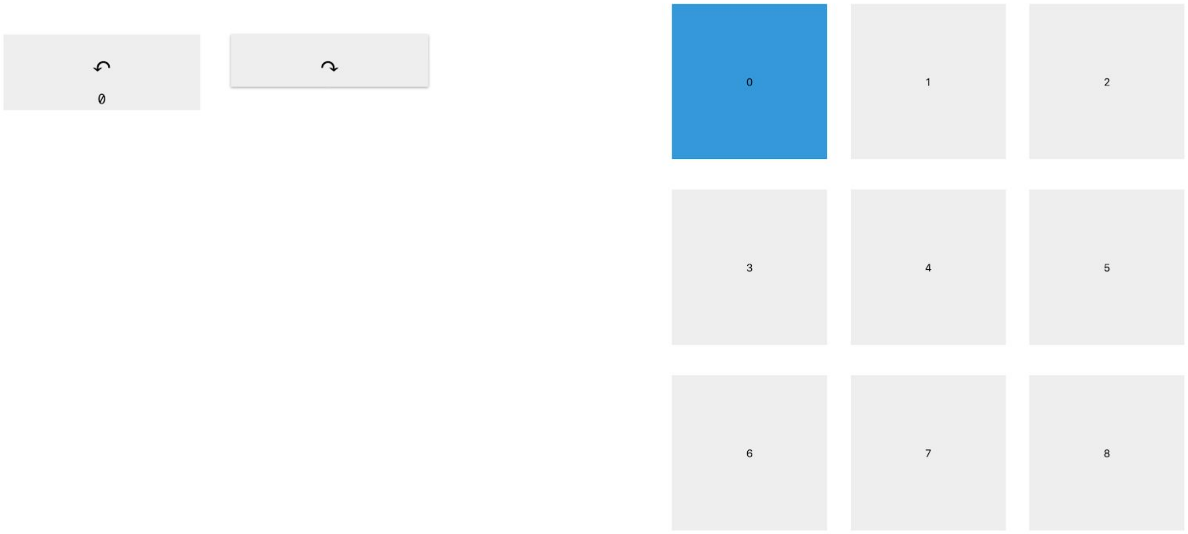
Step 2: We click the 0 button and the grid turns blue. The previous state of the 0 button is pushed onto the undo stack.



Step 3: We click the undo button. The 0 button is turned to grey. The redo stack now has an action to turn the 0 button back to blue.



Step 4: We click the redo button, the grid button turns back to blue, and the action to turn the 0 button to grey is pushed back onto the undo stack.



Some things to look into:

1. window.getComputedStyle
2. Element.classList

Rubric

This lab has a total of 100 points. Each day late is 10 points off your overall score.

Criteria	Explanation	Points
Code Style	Stack implementation is separated, code is readable	5
Questions.txt	Questions correct	5
Task 1	Answer each of the questions	10
Task 2	Each piece of the grid changes color when clicked	20
Task 3A	The stack is visualized properly when undoing and redoing	30
Task 3B	Pressing the undo and redo buttons undoes or redoes an action	20
Task 3 Demo	Demoed and shown to TA	10