Date 17/5/2023

CSM3103

Front-End Development

DR RABIEI B MAMAT

OMAR ISMAIL ABDJALEEL ALOMORY

LAB 4

Task 1

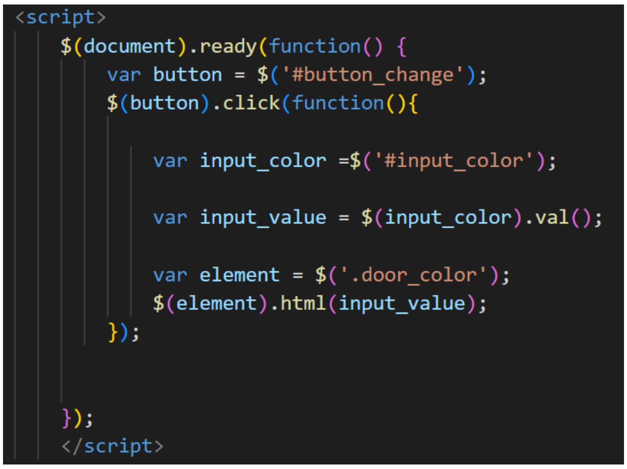
a. Get the code from http://frend.skimtech.my/Lab/Lab4/Task1.html , and evaluate and test the code.

  i. What is the id of button and input field

  ii. What happened when you clicked change button and why?

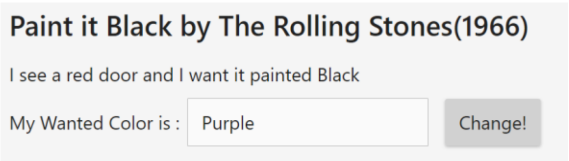
- The id for the button is **button_change**, and the input field is **input_color.**
- Nothing happened because we didn't write any JS to modify it or change its value.

a,b,c,d,e:



```
<script>
    $(document).ready(function() {
        var button = $('#button_change');
        $(button).click(function(){

            var input_color =$('#input_color');

            var input_value = $(input_color).val();

            var element = $('.door_color');
            $(element).html(input_value);
        });


    });
    </script>
```

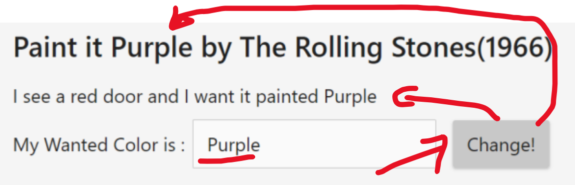
f. Test and evaluate your code. Comment on the new behaviour of your code



Paint it Black by The Rolling Stones(1966)

I see a red door and I want it painted Black

My Wanted Color is : Purple

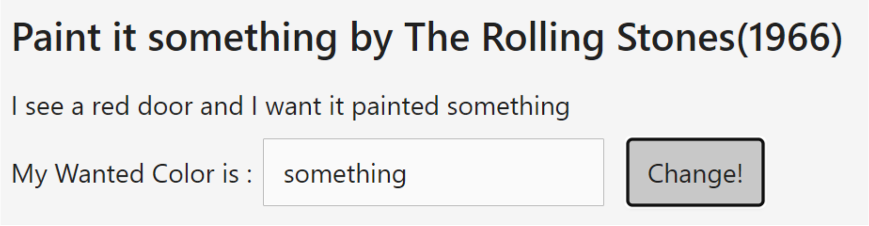


Paint it Purple by The Rolling Stones(1966)

I see a red door and I want it painted Purple

My Wanted Color is : Purple

Different output:



Paint it something by The Rolling Stones(1966)

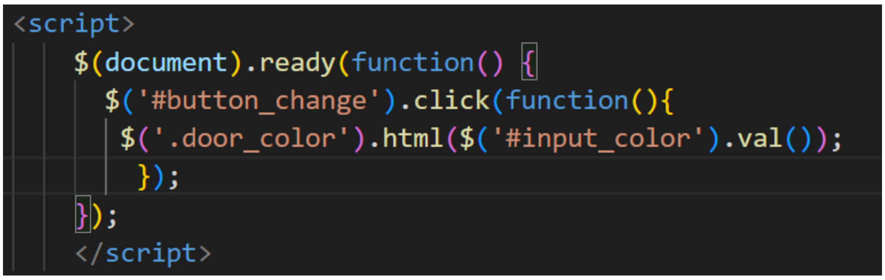
I see a red door and I want it painted something

My Wanted Color is : something

Basically, there is two span tag elements wrapping the words black, and the JS query going to take the id of the input field and the button and change the value entered to the input field to the values of the span element.

g. Basically, all those id and value in the code can be access directly without using any variables. **Rewrite the code by directly using the jQuery selector**

Code



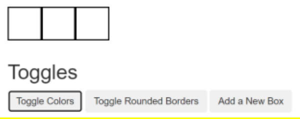
```
<script>
    $(document).ready(function() {
      $('#button_change').click(function(){
        $('.door_color').html($('#input_color').val());
         });
    });
    </script>
```

Task 2

a. Get the code from https://frend.skimtech.my/Lab/Lab4/Task2.html , and evaluate and test the code
   i. What are id's that exist in the code above?
   ii. Open the file inside browser, what have you see in the browser?
   iii. Click any button, what happened? Why

I. Ids: boxes, box1, box2, box3, toggles, button_toggle_colors, button_toggle_roundedges, button_add_box. **Total of 8 id exist in the code above.**

II. Three horizontal boxes, toggles text and below it three buttons.



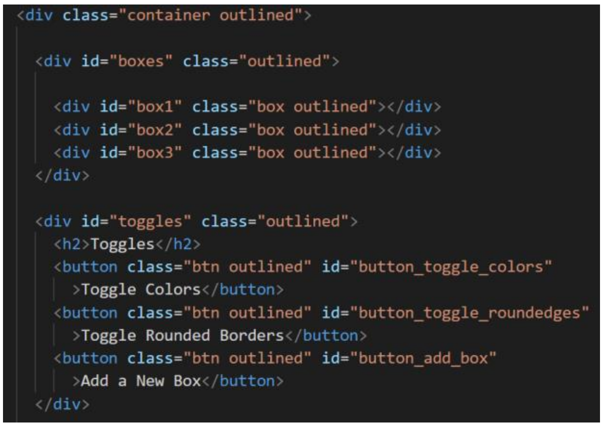
a.

III. Nothing happened because we there is no code to modify the content.

b. Your mission:
   i. Click on Toggle Colors button: Toggles the color between white and cardinal
   ii. Click on Toggle Rounded Borders: Toggles the appearance of a border on the boxes
   iii. Click add a new box: adds one new box to the container

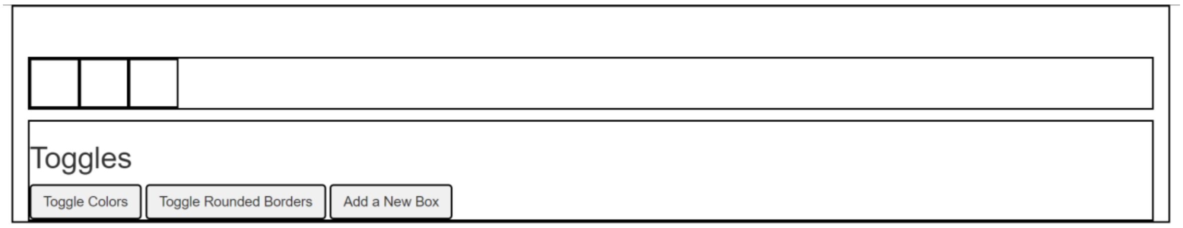
c. Implements the following to accomplished your mission
   i. Add outlined class to each <div>. Refer to <div> for box2 as an example. Save and open the file in browser
      i. What changes have you see?
   ii. Add a click function to button_toggle_color, and implementing change color process

1. **outline is added to each div tag.**



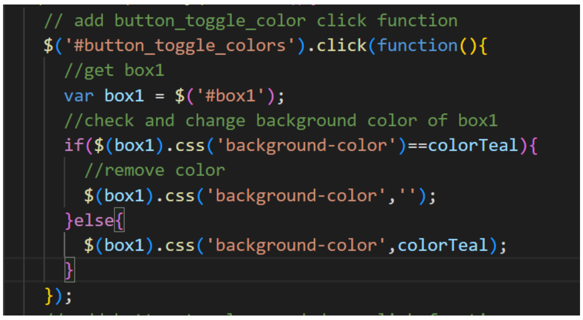
```html
<div class="container outlined">

  <div id="boxes" class="outlined">

    <div id="box1" class="box outlined"></div>
    <div id="box2" class="box outlined"></div>
    <div id="box3" class="box outlined"></div>
  </div>

  <div id="toggles" class="outlined">
    <h2>Toggles</h2>
    <button class="btn outlined" id="button_toggle_colors"
      >Toggle Colors</button>
    <button class="btn outlined" id="button_toggle_roundedges"
      >Toggle Rounded Borders</button>
    <button class="btn outlined" id="button_add_box"
      >Add a New Box</button>
  </div>
```

The changes occur on the border of each div tag, each div tag have now solid black border with 2px width, the figure below shows the result.



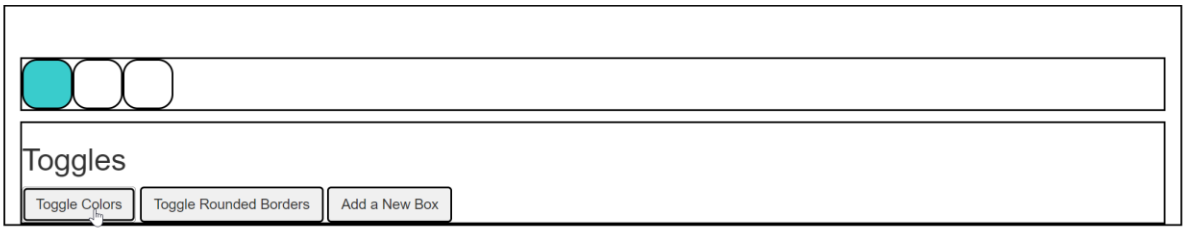
2. **adding buttons functions:**

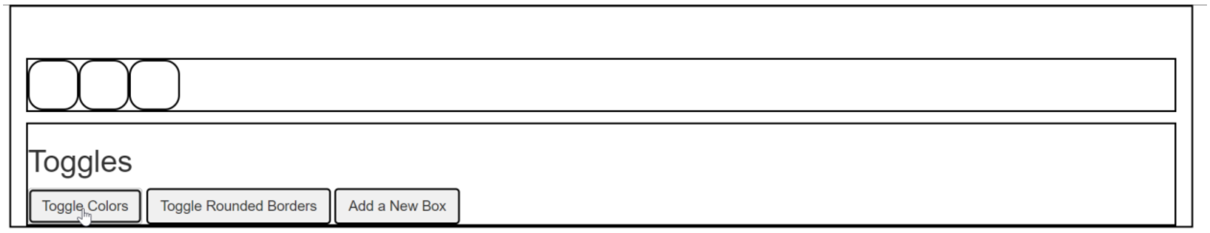
button_toggle_color function is added to implement the changes of color process.



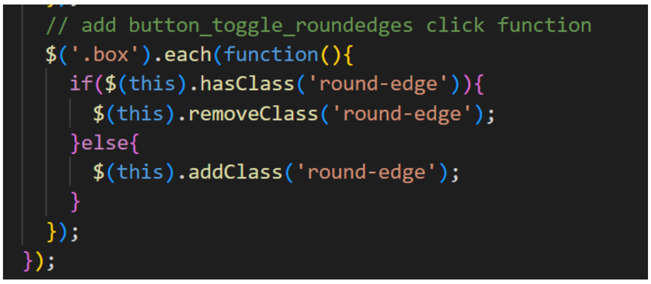
```javascript
// add button_toggle_color click function
$('#button_toggle_colors').click(function(){
  //get box1
  var box1 = $('#box1');
  //check and change background color of box1
  if($(box1).css('background-color')==colorTeal){
    //remove color
    $(box1).css('background-color','');
  }else{
    $(box1).css('background-color',colorTeal);
  }
});
```

Output





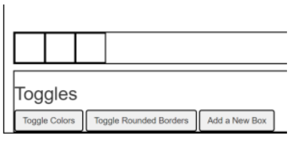
adding an event listener for button_toggle_roundedges.

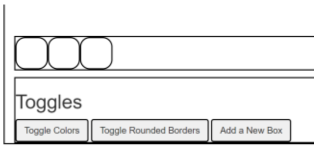


```
  // add button_toggle_roundedges click function
  $('.box').each(function(){
    if($(this).hasClass('round-edge')){
      $(this).removeClass('round-edge');
    }else{
      $(this).addClass('round-edge');
    }
  });
});
```

Output

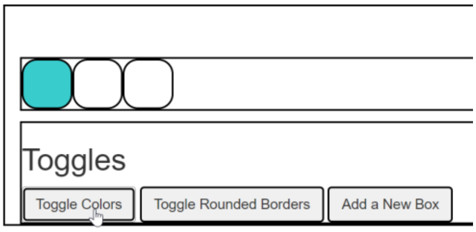
Before **After**

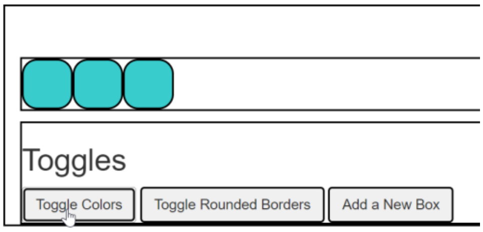




Based on the previous code we can do it on the colors function to be applied to all boxes.

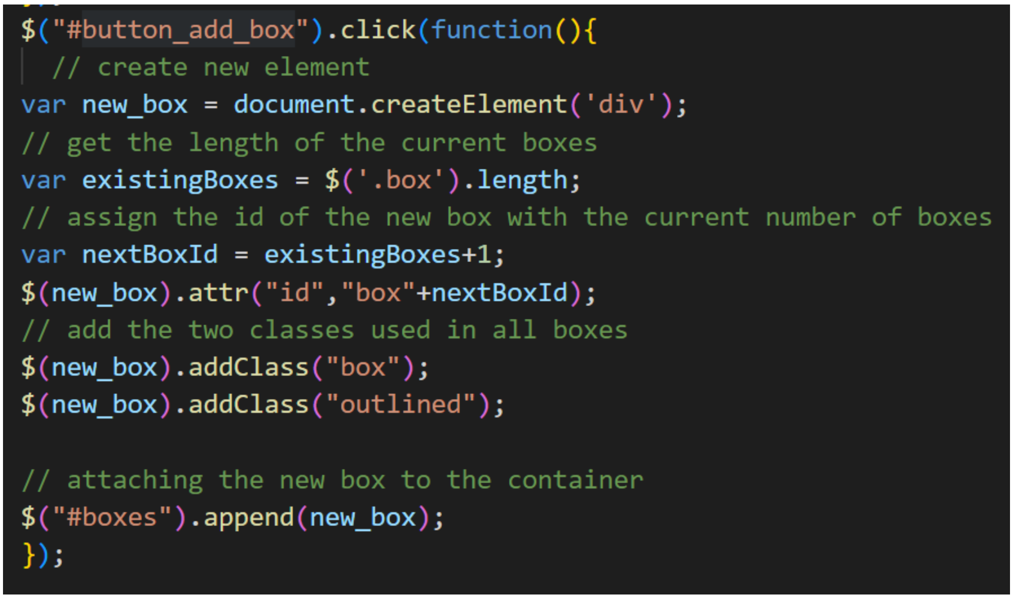
Before **After**





adding an event listener for button_add_box.

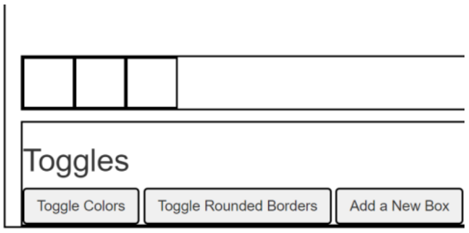
Code

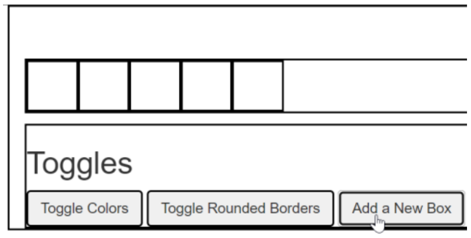


```
$("#button_add_box").click(function(){
  // create new element
var new_box = document.createElement('div');
// get the length of the current boxes
var existingBoxes = $('.box').length;
// assign the id of the new box with the current number of boxes
var nextBoxId = existingBoxes+1;
$(new_box).attr("id","box"+nextBoxId);
// add the two classes used in all boxes
$(new_box).addClass("box");
$(new_box).addClass("outlined");

// attaching the new box to the container
$("#boxes").append(new_box);
});
```

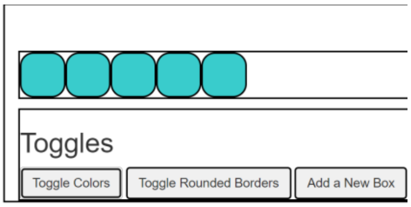
Output

Before **After clicking 2 times**





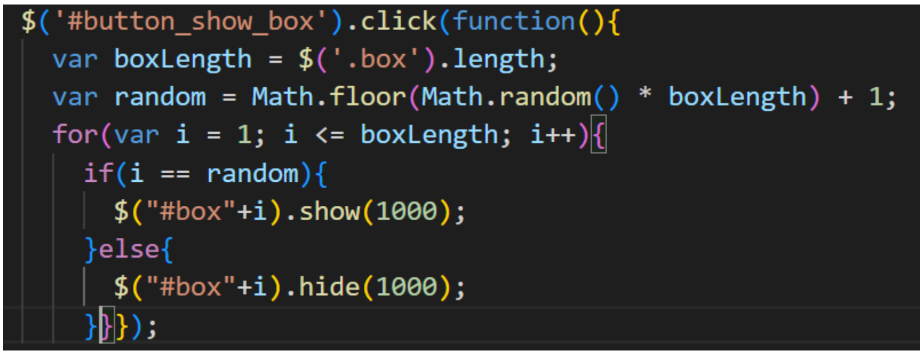
viii. Test and evaluate your code



The code is working just fine: adding new box, rounding the boxes, or toggling the color.

ix. Modify the code above so that when button clicked, it only show one element which selected randomly .

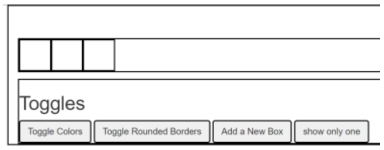
Code

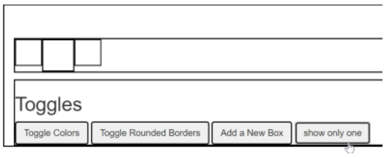


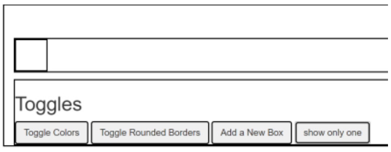
```
$('#button_show_box').click(function(){
  var boxLength = $('.box').length;
  var random = Math.floor(Math.random() * boxLength) + 1;
  for(var i = 1; i <= boxLength; i++){
    if(i == random){
      $("#box"+i).show(1000);
    }else{
      $("#box"+i).hide(1000);
    }}});
```

Output

Before in-between (hide(1000)) after







Task 3

a. Get the code from https://frend.skimtech.my/Lab/Lab4/Task2.html , and evaluate and test the code:
   i. Determine the id's of each element
   ii. Recognise any input element that associated with the form
   iii. Your Task are:
      1. Add DatePicker
      2. Use selectable interaction
      3. Use Sortable widget
      4. Use the tooltip

Ids:

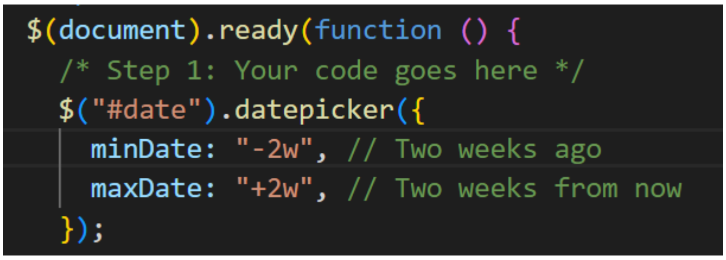
- name_input
- name
- date_input
- date
- year_input
- ugrade
- grade
- color_input
- red
- blue
- hot_pink
- mustard
- dark_green

b. Inserting the datepicker into a form
   i. In the document ready function add this code

```
$('#date').datepicker();
```

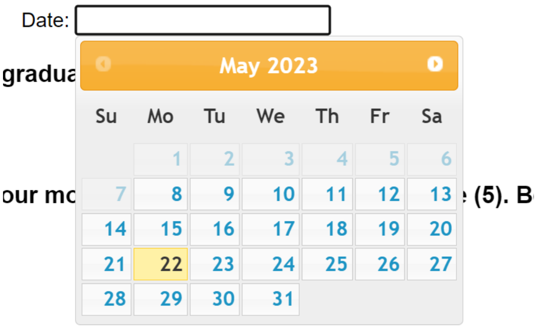
   i. Can you set the date to limit the selectable date range to only include dates from the previous two weeks to the next two weeks.

Code



```
$(document).ready(function () {
  /* Step 1: Your code goes here */
  $("#date").datepicker({
    minDate: "-2w", // Two weeks ago
    maxDate: "+2w", // Two weeks from now
  });
```

Output



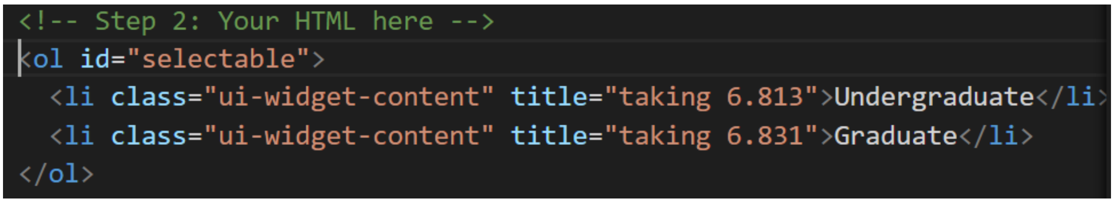
May 2023						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

c. Using selectable for enrolment type

   i. Comment radiobutton input

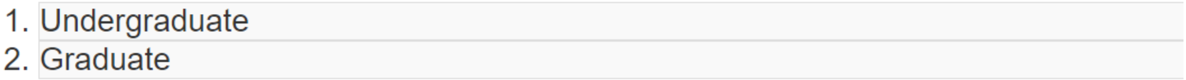
   ii. Add the following code

```
  <ol id="selectable">
    <li class="ui-widget-content" title="taking 6.813">Undergraduate</li>
    <li class="ui-widget-content" title="taking 6.831">Graduate</li>
</ol>
```



```
<!-- Step 2: Your HTML here -->
<ol id="selectable">
  <li class="ui-widget-content" title="taking 6.813">Undergraduate</li>
  <li class="ui-widget-content" title="taking 6.831">Graduate</li>
</ol>
```

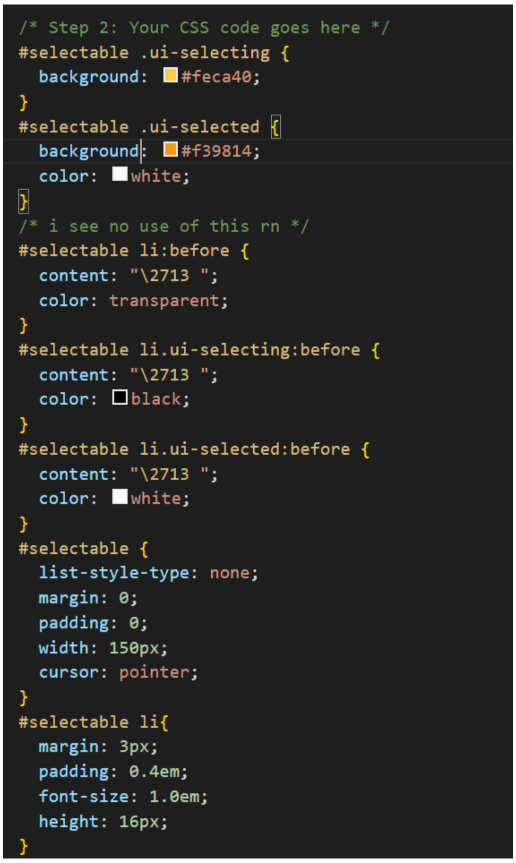
Output



1. Undergraduate
2. Graduate

iii. Add some css in style section

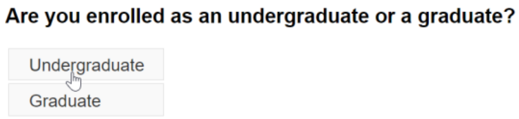
Code



```
/* Step 2: Your CSS code goes here */
#selectable .ui-selecting {
  background: #feca40;
}
#selectable .ui-selected {
  background: #f39814;
  color: white;
}
/* i see no use of this rn */
#selectable li:before {
  content: "\2713 ";
  color: transparent;
}
#selectable li.ui-selecting:before {
  content: "\2713 ";
  color: black;
}
#selectable li.ui-selected:before {
  content: "\2713 ";
  color: white;
}
#selectable {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 150px;
  cursor: pointer;
}
#selectable li{
  margin: 3px;
  padding: 0.4em;
  font-size: 1.0em;
  height: 16px;
}
```

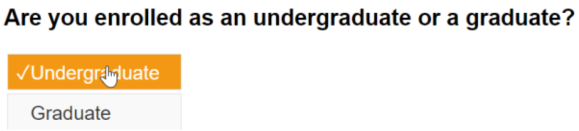
Output

Before



Are you enrolled as an undergraduate or a graduate?

After

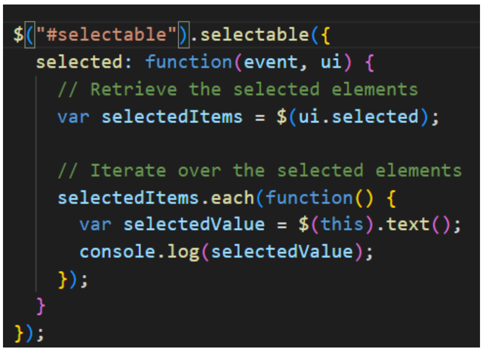


Are you enrolled as an undergraduate or a graduate?

v. How can we retrieve the value of the selected item?

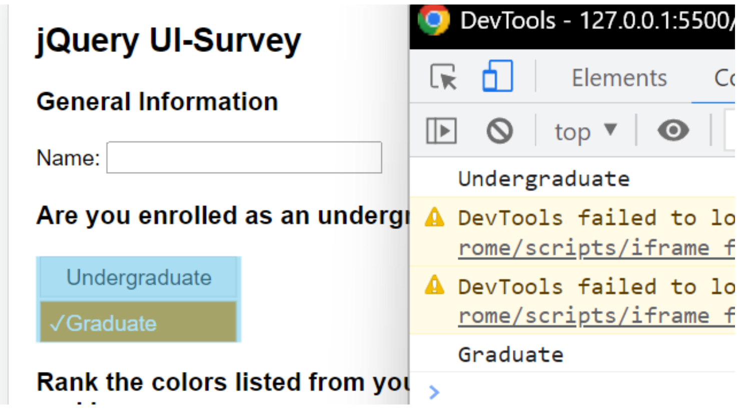
To retrieve the selected values from the **`selectable()`** function in jQuery, you can use the **`selected`** option of the **`selectable()`** method. The **`selected`** option allows you to specify a callback function that will be called whenever an element is selected within the selectable container.

Code



```
$("#selectable").selectable({
  selected: function(event, ui) {
    // Retrieve the selected elements
    var selectedItems = $(ui.selected);

    // Iterate over the selected elements
    selectedItems.each(function() {
      var selectedValue = $(this).text();
      console.log(selectedValue);
    });
  }
});
```

Output



jQuery UI-Survey

General Information

Name:

Are you enrolled as an undergr

Undergraduate

✓Graduate

Rank the colors listed from you

```
Undergraduate
DevTools failed to lo
rome/scripts/iframe_f
DevTools failed to lo
rome/scripts/iframe_f
Graduate
```

d. Using sortable for the color input

   i. Comment all input element that related to color selections

   ii. Replace with the following code

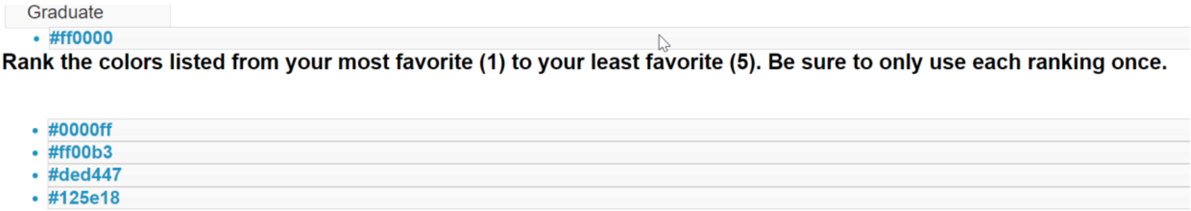
```
<ul id="sortable">
  <li class="ui-state-default">#ff0000</li>
  <li class="ui-state-default">#0000ff</li>
  <li class="ui-state-default">#ff00b3</li>
  <li class="ui-state-default">#ded447</li>
  <li class="ui-state-default">#125e18</li>
</ul>
```

   iii. Adding a control to sort the list and disable the selection

```
$("#sortable").sortable();
$("#sortable").disableSelection();
```

   iv. Check your output, now you can drag any color to a new position

Output



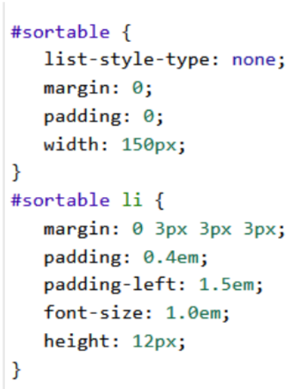
Graduate

- #ff0000

Rank the colors listed from your most favorite (1) to your least favorite (5). Be sure to only use each ranking once.

- #0000ff
- #ff00b3
- #ded447
- #125e18

v. Add the following style to your code

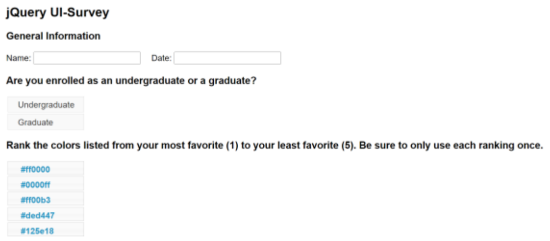


```
#sortable {
   list-style-type: none;
   margin: 0;
   padding: 0;
   width: 150px;
}
#sortable li {
   margin: 0 3px 3px 3px;
   padding: 0.4em;
   padding-left: 1.5em;
   font-size: 1.0em;
   height: 12px;
}
```

vi. What do you think this CSS did? You can reverse the step to see the changes after and before the use of CSS

This CSS have modified the width of the unordered list to 150px, changed the list style (removed the style), added padding, increased the font size and set up the height to 12px to the list items.

Output



jQuery UI-Survey

General Information

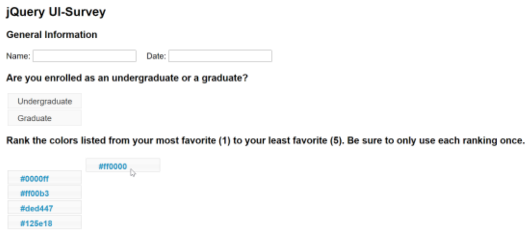
Name: Date:

Are you enrolled as an undergraduate or a graduate?

Undergraduate
Graduate

Rank the colors listed from your most favorite (1) to your least favorite (5). Be sure to only use each ranking once.

#ff0000
#0000ff
#ff00b3
#ded447
#125e18



jQuery UI-Survey

General Information

Name: Date:

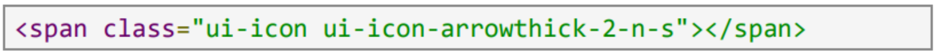
Are you enrolled as an undergraduate or a graduate?

Undergraduate
Graduate

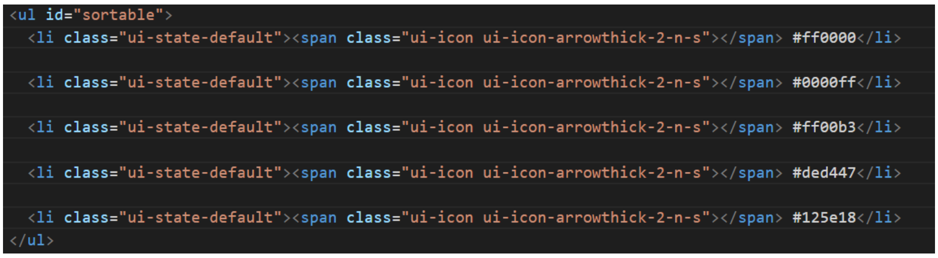
Rank the colors listed from your most favorite (1) to your least favorite (5). Be sure to only use each ranking once.

#ff0000
#0000ff
#ff00b3
#ded447
#125e18

vii. Add the following span with class before color code in the <li> in (ii) above

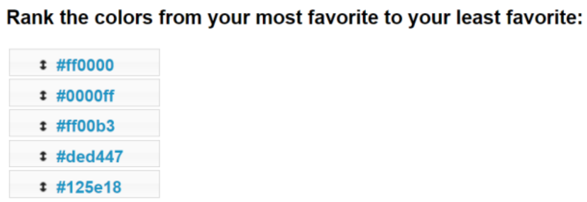


```
<span class="ui-icon ui-icon-arrowthick-2-n-s"></span>
```



```
<ul id="sortable">
  <li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-2-n-s"></span> #ff0000</li>

  <li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-2-n-s"></span> #0000ff</li>

  <li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-2-n-s"></span> #ff00b3</li>

  <li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-2-n-s"></span> #ded447</li>

  <li class="ui-state-default"><span class="ui-icon ui-icon-arrowthick-2-n-s"></span> #125e18</li>
</ul>
```

Output

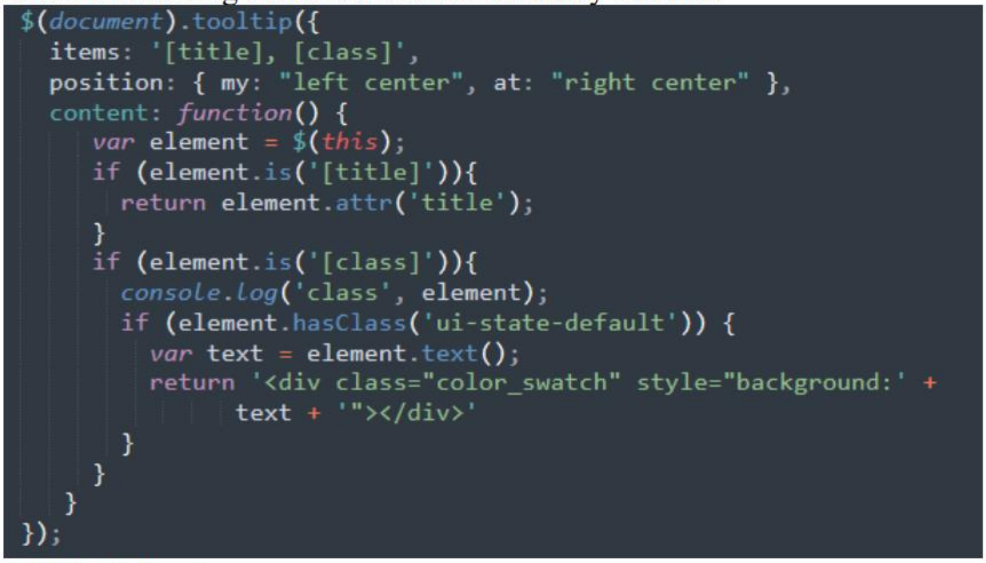


Rank the colors from your most favorite to your least favorite:

↕ #ff0000
↕ #0000ff
↕ #ff00b3
↕ #ded447
↕ #125e18

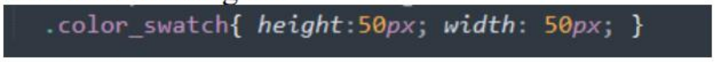
e. Adding a Tooltip

i. Add the following code inside document ready function



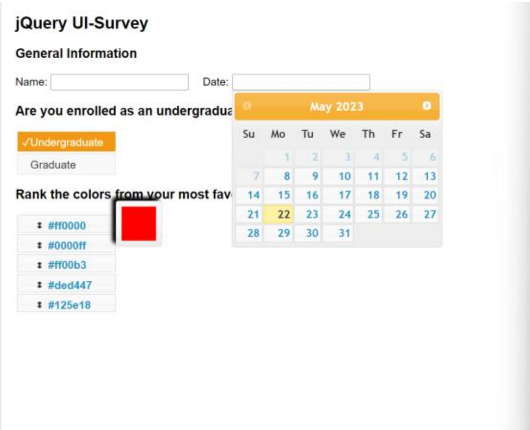
```
$(document).tooltip({
  items: '[title], [class]',
  position: { my: "left center", at: "right center" },
  content: function() {
     var element = $(this);
     if (element.is('[title]')){
       return element.attr('title');
     }
     if (element.is('[class]')){
       console.log('class', element);
       if (element.hasClass('ui-state-default')) {
         var text = element.text();
         return '<div class="color_swatch" style="background:' +
                text + '"></div>'
       }
     }
   }
});
```

ii. Add the following css



```
 .color_swatch{ height:50px; width: 50px; }
```

Final output of task 3



jQuery UI-Survey

General Information

Name: Date:

Are you enrolled as an undergradua

✓Undergraduate

Graduate

Rank the colors from your most fav

May 2023						
Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

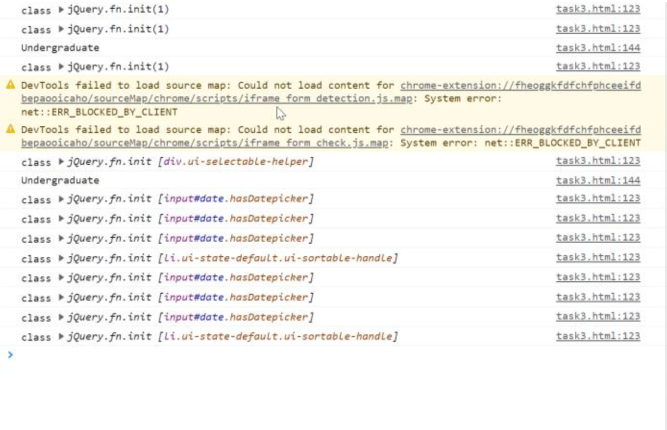
↕ #ff0000

↕ #0000ff

↕ #ff00b3

↕ #ded447

↕ #125e18

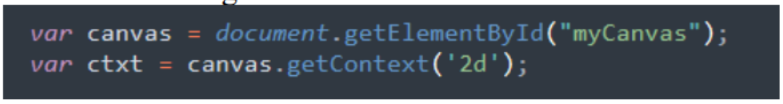


```
class ▸jQuery.fn.init(1)                                             task3.html:123
class ▸jQuery.fn.init(1)                                             task3.html:123
Undergraduate                                                        task3.html:144
class ▸jQuery.fn.init(1)                                             task3.html:123
⚠ DevTools failed to load source map: Could not load content for chrome-extension://fheoggkfdfchfphceeifd
bepaooicaho/sourceMap/chrome/scripts/iframe_form_detection.js.map: System error:
net::ERR_BLOCKED_BY_CLIENT
⚠ DevTools failed to load source map: Could not load content for chrome-extension://fheoggkfdfchfphceeifd
bepaooicaho/sourceMap/chrome/scripts/iframe_form_check.js.map: System error: net::ERR_BLOCKED_BY_CLIENT
class ▸jQuery.fn.init [div.ui-selectable-helper]                     task3.html:123
Undergraduate                                                        task3.html:144
class ▸jQuery.fn.init [input#date.hasDatepicker]                     task3.html:123
class ▸jQuery.fn.init [input#date.hasDatepicker]                     task3.html:123
class ▸jQuery.fn.init [input#date.hasDatepicker]                     task3.html:123
class ▸jQuery.fn.init [li.ui-state-default.ui-sortable-handle]       task3.html:123
class ▸jQuery.fn.init [input#date.hasDatepicker]                     task3.html:123
class ▸jQuery.fn.init [input#date.hasDatepicker]                     task3.html:123
class ▸jQuery.fn.init [input#date.hasDatepicker]                     task3.html:123
class ▸jQuery.fn.init [li.ui-state-default.ui-sortable-handle]       task3.html:123
>
```

Task 4

b. Getting the context of the canvas

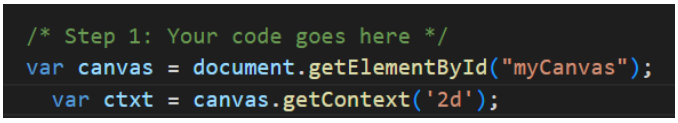
a. Add the following code:



```
var canvas = document.getElementById("myCanvas");
var ctxt = canvas.getContext('2d');
```

i. What is the meaning of context in this case?

Code



```
/* Step 1: Your code goes here */
var canvas = document.getElementById("myCanvas");
  var ctxt = canvas.getContext('2d');
```

Output

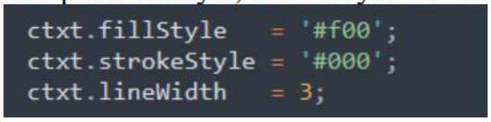
Nothing changed.

Answer(i)

Context here refers to the drawing context of the **`<canvas>`** element. The **`<canvas>`** element is an HTML5 element that provides a space for rendering graphics, animations, and other visual effects using JavaScript.The **`getContext()`** method is used to obtain the drawing context, which allows you to draw and manipulate graphics within the canvas.

c. Drawing a triangle to canvas

a. Setup the fillStyle, strokeStyle and linewidth.



```
ctxt.fillStyle   = '#f00';
ctxt.strokeStyle = '#000';
ctxt.lineWidth   = 3;
```

b. Describe the purpose of each method in (a)

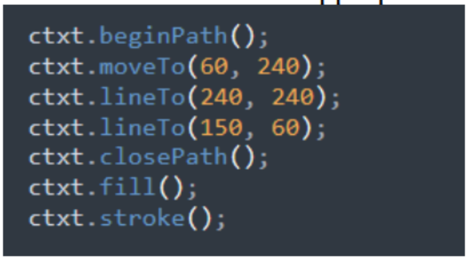
- **`ctxt.fillStyle = "#f00";`**
  This line sets the fill color to "**`#f00`**", which is a red color in hexadecimal format. This means that any subsequent shapes or paths drawn using the **`fill().`**

- **`ctxt.strokeStyle = "#000";`**
  This line sets the stroke color to "**`#000`**", which is a black color in hexadecimal format. This means that any subsequent shapes or paths drawn using the **`stroke()`** method will have an outline or border with this color.

- **`ctxt.lineWidth = 3;`**

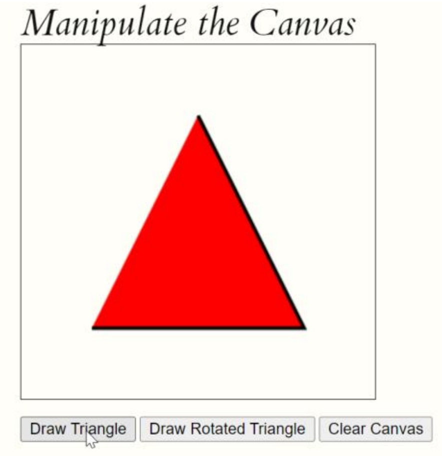
  This line sets the line width to 3 pixels. When drawing paths or outlines using the **`stroke()`** method, the specified line width will determine the thickness of the lines.

c. Put these codes into an appropriate function:



```
ctxt.beginPath();
ctxt.moveTo(60, 240);
ctxt.lineTo(240, 240);
ctxt.lineTo(150, 60);
ctxt.closePath();
ctxt.fill();
ctxt.stroke();
```

d. Test you code and evaluate the output

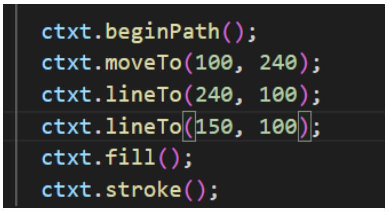


It drew a triangle. First it starts by moving the pointer to coordinate (x,y) in the canvas, then draw the line from coordinate(60,240) to coordinate (240,240), then to coordinate, lastly from coordinate (150, 60). ClosePath to terminate, then fill the shape as well the stroke.

e. Rearrange the position of coordinate and test your code.

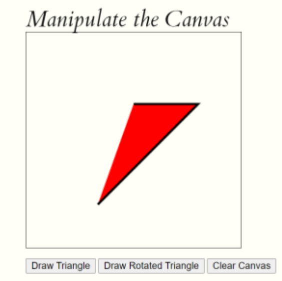
f. What is the use of moveTo methods above?

If we rearrange the coordinates and we got different output, the entered coordinates are:



```
ctxt.beginPath();
ctxt.moveTo(100, 240);
ctxt.lineTo(240, 100);
ctxt.lineTo(150, 100);
ctxt.fill();
ctxt.stroke();
```

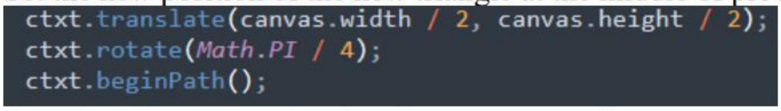
Output



The moveTo() method is used to set the starting point of a new subpath in the canvas. It specifies the coordinates where the path should begin.

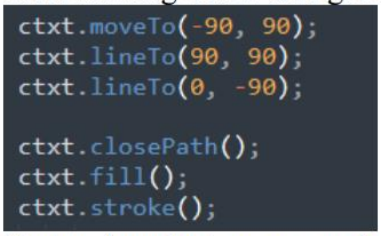
d. a new rotate triangle function

   a. Set the new position of the new triangle at the middle of previous triang



```
ctxt.translate(canvas.width / 2, canvas.height / 2);
ctxt.rotate(Math.PI / 4);
ctxt.beginPath();
```

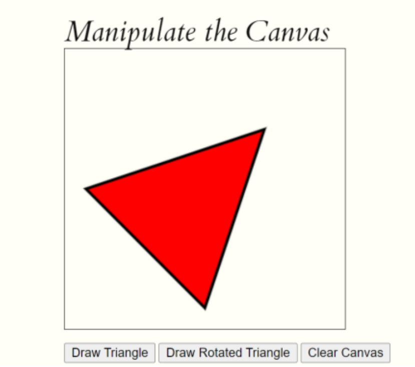
   b. Start drawing a new triangle using a new coordinate



```
ctxt.moveTo(-90, 90);
ctxt.lineTo(90, 90);
ctxt.lineTo(0, -90);

ctxt.closePath();
ctxt.fill();
ctxt.stroke();
```

   c. Test and evaluate your result.

Output



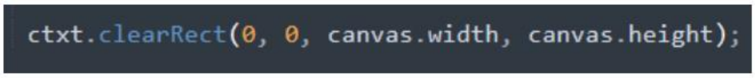
Manipulate the Canvas

d. You can press the rotate many times, and can have different triangle at different location, why it’s happened?

We set up the canvas starting point to the center of the canvas **`ctxt.translate(canvas.width / 2, canvas.height / 2)`** , which is the middle, then we want to rotate 45 degrees **`ctxt.rotate(Math.PI / 4).`** As a result, subsequent triangles will be drawn at different locations and orientations relative to the center of the canvas.

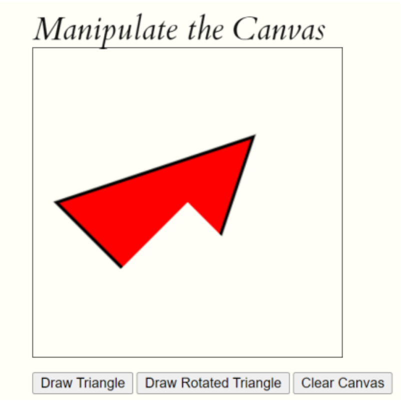
e. Clear the canvas

a. Put this code to clear the canvas



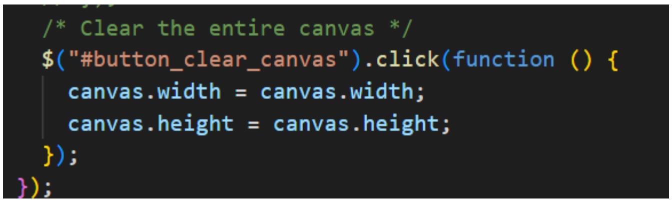
```
ctxt.clearRect(0, 0, canvas.width, canvas.height);
```

Sometimes it delete everything and sometimes it does not like the figure below.



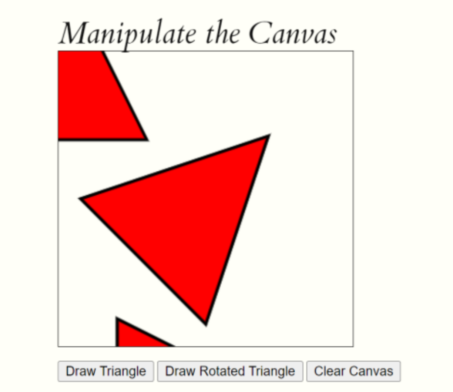
b. Can you write more better clear canvas , to clear all canvas

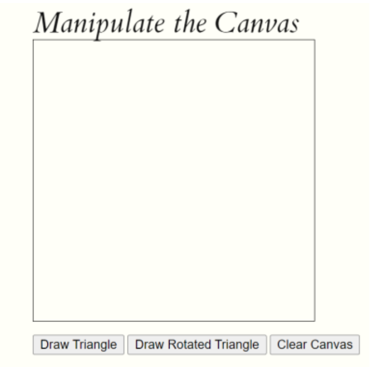
This code is better solution when deleting canvas:



```
  /* Clear the entire canvas */
  $("#button_clear_canvas").click(function () {
    canvas.width = canvas.width;
    canvas.height = canvas.height;
  });
});
```

It can delete everything without any remaining:





GitHub: http://github.com/Alomory/CSM3103