Lab 4

Task 1

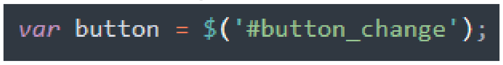
a. Get the code from http://frend.skimtech.my/Lab/Lab4/Task1.html , and evaluate and test the code.

   i. What is the id of button and input field. button id = " button_change" & input id = "input_color"

   ii. What happened when you clicked change button and why? nothing happened

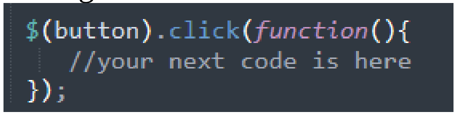
b. Your mission: when the button is clicked, all occurrences of the word Black to be changed to the text in the input field.

   a. Select the change button



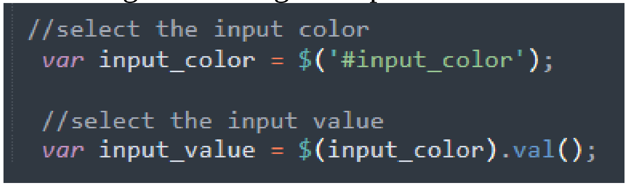
```
var button = $('#button_change');
```

   b. Setting a click listener for button



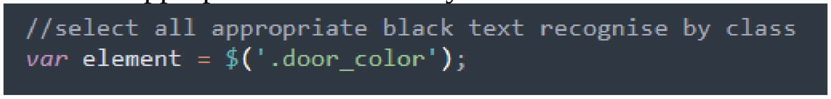
```
$(button).click(function(){
    //your next code is here
});
```

   c. Retrieving and storing the input value



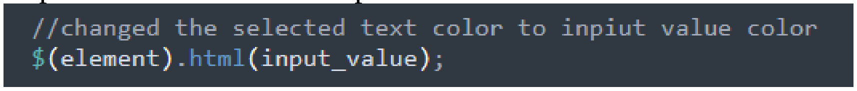
```
//select the input color
 var input_color = $('#input_color');

 //select the input value
 var input_value = $(input_color).val();
```

   d. Select all appropriates black text by class selector



```
//select all appropriate black text recognise by class
var element = $('.door_color');
```

   e. Replace the text with the input value



```
//changed the selected text color to inpiut value color
$(element).html(input_value);
```

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

   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**

   h. Rewrite the code: can refer to task 2 if in doubt

      i. Instead of input field, change it to dropdown which display available color understood by HTML (use select tag)

      ii. Retrieve the value from the select tag

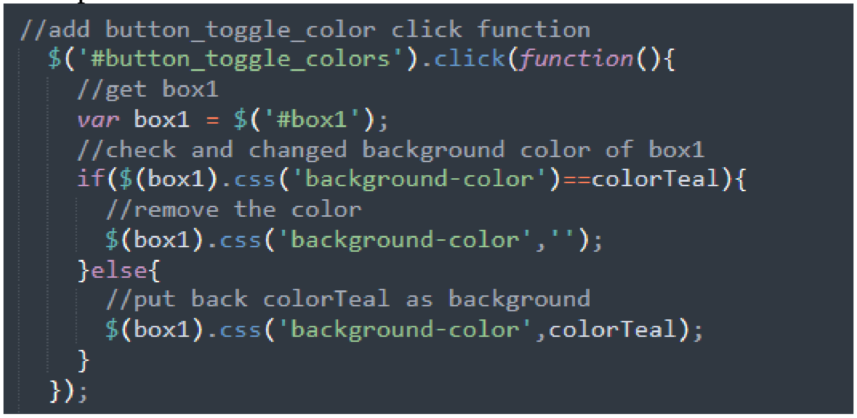
      iii. Change the color of all appropriate text to selected color

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

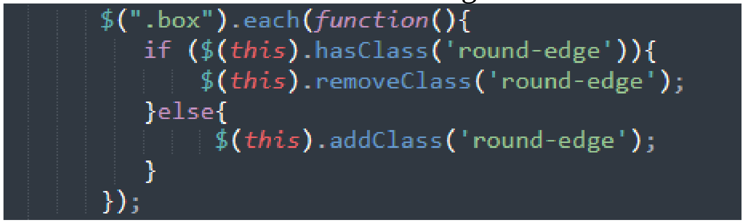
b. Your mission:
    i. Click on Toggle Colors button: Toggles the color between white and cardinal
    ii. Click on Toggle Rounded Borders: Toggles the appearance of a border on the boxes
    iii. Click add a new box: adds one new box to the container

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



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

    iii. Create an event listener for button_toggle_roundedges
    iv. Inside the function add the following code



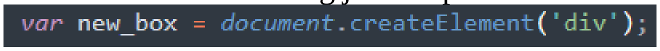
```
$(".box").each(function(){
   if ($(this).hasClass('round-edge')){
       $(this).removeClass('round-edge');
   }else{
        $(this).addClass('round-edge');
   }
});
```

    v. Based on code in iv, can you modify a code in ii, so it change all box background and returned all to vanilla as well!

vi. Create an event listener for button_add_box.

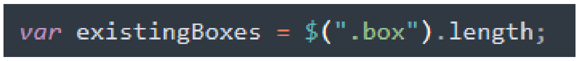
vii. Inside the function put the following code

   i. Create new element using javascript



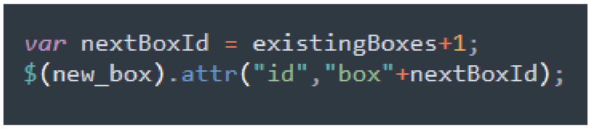
```
var new_box = document.createElement('div');
```

   ii. Get current number of element with class .box



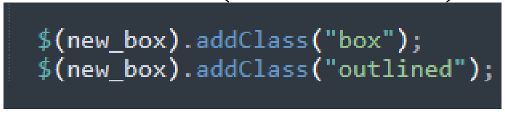
```
var existingBoxes = $(".box").length;
```

   iii. With the current number of box create new id for new box



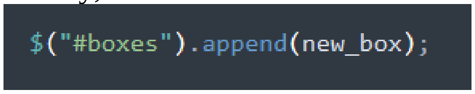
```
var nextBoxId = existingBoxes+1;
$(new_box).attr("id","box"+nextBoxId);
```

   iv. Add two classes (box and outlined) to new_box



```
$(new_box).addClass("box");
$(new_box).addClass("outlined");
```

   v. Finally, attached new box into box container



```
$("#boxes").append(new_box);
```

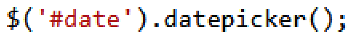
viii. Test and evaluate your code

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

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

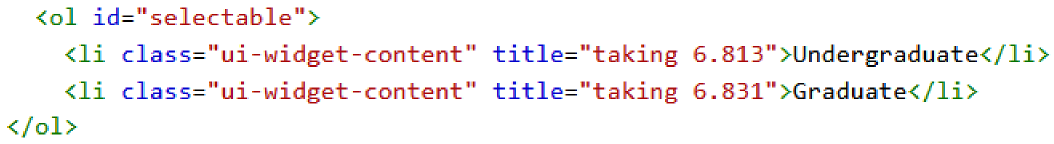
b. Inserting the datepicker into a form
   i. In the document ready function add this code



```
$('#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.

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>
```

   iii. Add some css in style section

```
#selectable .ui-selecting {
  background: #FECA40; }

#selectable .ui-selected {
  background: #F39814; color: white; }

#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; }
```

iv. And control the behaviour using jQuery selector

```
$("#selectable").selectable();
```

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

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

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;
}
```

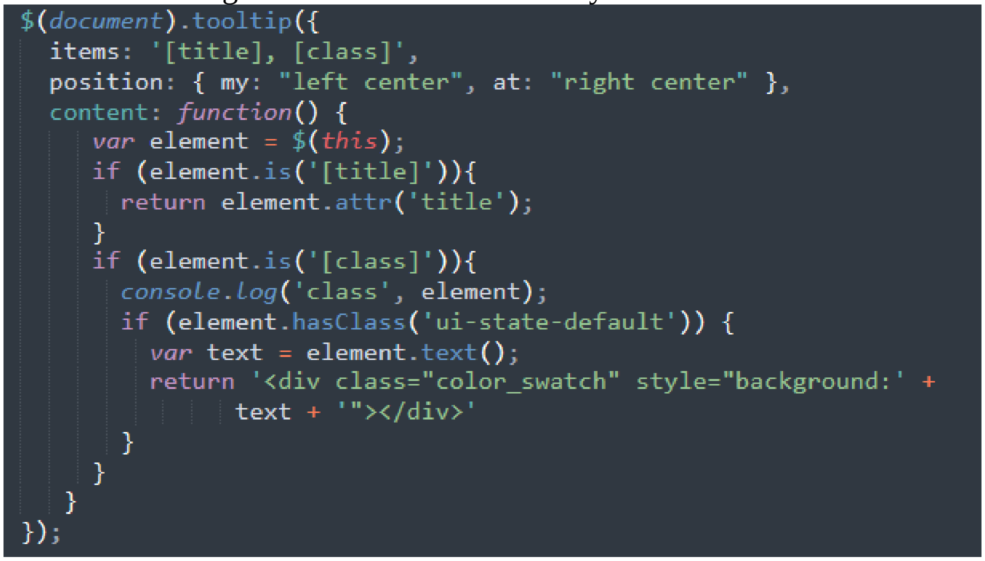
vi. What do you think this CSS did? You can reverse the step to see the changes after and before the use of CSS

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>
```

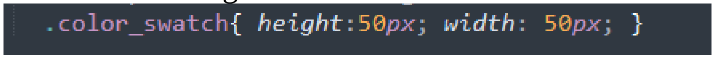
e. Adding a Tooltip

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; }
```