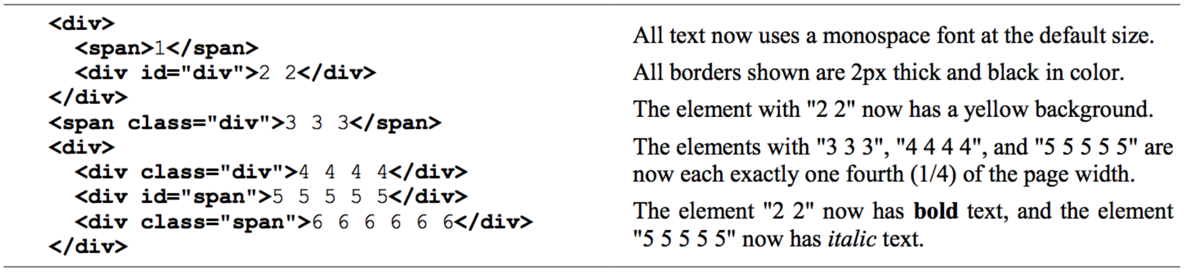
**CSC 435 Web Programming, Spring, 2018 Mid-term exam**
**Prof. Bei Xiao**

Total points (50pts + 15pts extra).

Instructions:

1. Download the midterm code folder in blackboard. Each question contains starter codes that you need.
2. Please upload each question as firstname_lastname_QuestionX.zip that contains .js, .css, .html and your resources files. Make sure you upload correctly.
3. The exam is open notes, open source and open book. But you can't discuss with anyone and **you can't share your solutions**. Please do not copy solutions from the Internet. **If caught, you will be reported to the authority.**
4. If you finish the exam, please make sure you finalize the submission and leave the room quietly.
5. Please make sure your code compile and run on the browser (use Firefox and chrome to look at your solution).
6. Cell phone and other smart phone devices are forbidden during the exam.
7. No late exam can be graded.
8. Partial credits will be given to all the steps involved. But code which doesn't compile will receive little points. Make sure your JS code works step by step.

1. (10pts) Write the CSS code necessary to recreate the following appearance on-screen, exactly as shown. The page uses the same HTML code below. You are NOT allowed to modify the HMTL code.



```
<div>
  <span>1</span>
  <div id="div">2 2</div>
</div>
<span class="div">3 3 3</span>
<div>
  <div class="div">4 4 4 4</div>
  <div id="span">5 5 5 5 5</div>
  <div class="span">6 6 6 6 6 6</div>
</div>
```

All text now uses a monospace font at the default size.

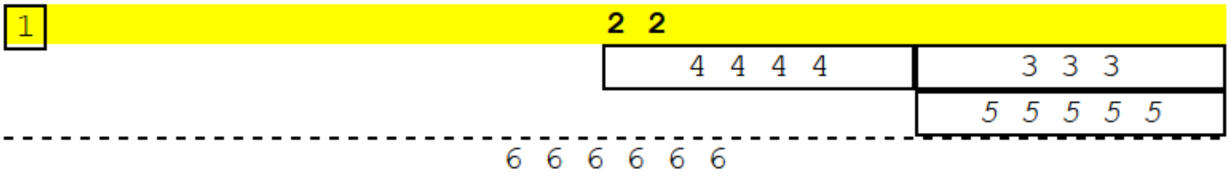
All borders shown are 2px thick and black in color.

The element with "2 2" now has a yellow background.

The elements with "3 3 3", "4 4 4 4", and "5 5 5 5 5" are now each exactly one fourth (1/4) of the page width.

The element "2 2" now has **bold** text, and the element "5 5 5 5 5" now has *italic* text.

Page appearance (span across entire page):



2. (10pts)

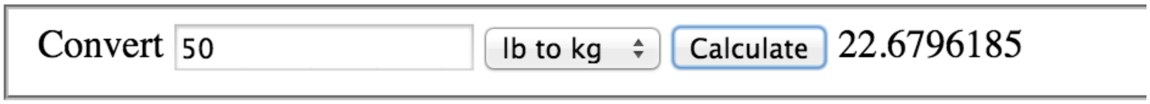
Create the file Question2.js referenced by Question2. html. Write a function convert() in Questeion2.js that takes the value in the text input and convert it from either from kilograms to pounds or from pounds to kilograms depends on the dropdown box.

The results should be displayed in the empty span with the id #answer.

The conversion factor from pounds to kilograms is 0.45359. The conversion factor from kilograms to pounds is 2.2045.

You can edit the queston2.html to add ids and to the elements as necessary and you may assume valid input.

The following is an example of desirable output:



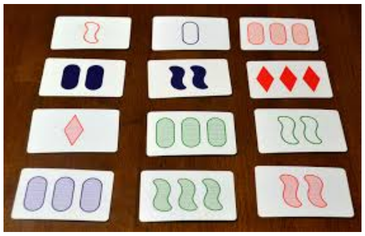
Convert 50 lb to kg 22.6796185

3. (10pts total). JS/OOP
   (3pts) Create  simple JS object called SetGameCard using constructor function.
    Each card has a properties color (‘red’,’green’,’blue’,’purple’) and shape (‘diamond’,’squiggle’,’tube’) and each has a number ( [1, 2,3]) of symbols  and texture (‘solid’, ‘stripped’, ‘open’).

   (4pts) Write a function that create a 12 card objects that each has a particular sets of the above properties using a random function. Notice each card must be unique (see image blow as an example).

   (4pts) Write a function to return True if the 3 randomly selected cards fulfill the following features:
   1) They have ALL different colors
   2) They have ALL different shapes
   3) They have the same numbers.
   4) They have ALL different textures.

An example draw of set cards.   You don’t have to use any images. You simply create a .html file with .js code in the <script> tag for this question.



4. (20pts) JS/Dom Blend Names

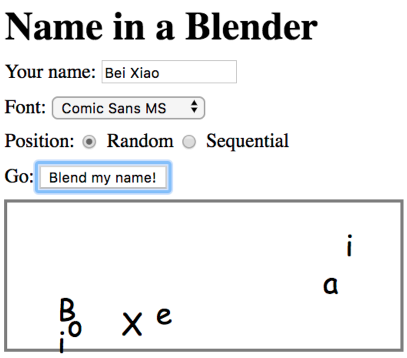
Write the JavaScript code to add behavior to the following page (.html in the starter folder) that has a user interface for "blending" a name. The UI allows the user to type his/her name into a text box. The user can click a button to "blend" the name, which causes each character of the name to be individually injected into a div with the id of output. Your code should enable the user to choose a font to use for displaying the characters of the name, which is either Arial, Comic Sans MS, or Times New Roman.

The letters of the name will be displayed in the font chosen. The letters can be positioned differently by choosing one of two radio buttons. If the "Random" button is checked, each letter of the name is positioned randomly with an x-coordinate between 0-300 pixels from the left edge of the output area, and a y-coordinate between 0-100 pixels from the top edge of the output area. If the "Sequential" button is checked, letters are placed at 15-pixel intervals, with the first at (x=15, y=15) within the output area, and the second at (x=30, y=30), the third at (x=45, y=45), and so on. (If there are a lot of letters, they might extend outside of the output area, but you don't need to worry about that.)

The code should work for multiple clicks of "Blend". On each click it should clear any previously inserted letters.

Hint: you might find Math.random() useful.
document.createElement() can be used to create an span element for the output of words.

These screenshots show the initial state, and after names have been typed and "Blend my name!" has been clicked.

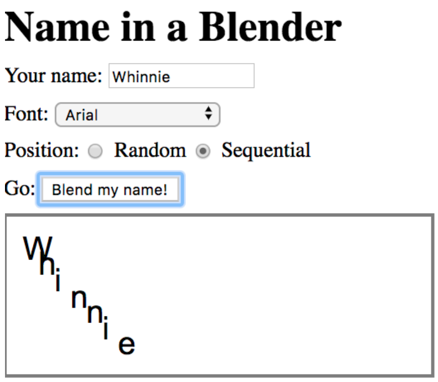


Name in a Blender

Your name: Bei Xiao

Font: Comic Sans MS

Position: ● Random ○ Sequential



Name in a Blender

Your name: Whinnie

Font: Arial

Position: ○ Random ● Sequential

5. (15pts Extra credit) JS/DOM

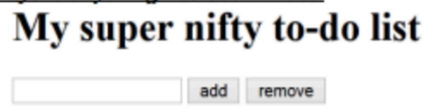
Write JavaScript code to add behavior to the following page for keeping track of a to-do-list. The page UI allows the user to type an item into a text box. The user can click the "add" button to add the item to the bottom of the list. Each word in the phrase should be inserted as a li, inside a ul with the id of list. If the user wishes to remove an item he or she can type the text of the item he or she wishes to remove in the text box and click the "remove" button. This should be case insensitive. For example, if the list only contains "foo" and the user tries to remove "FoO", it should be removed. If the user tries to remove an item that is in the list multiple times only the first occurrence should be removed. The items should have background colors that alternate between white and yellow (first white, then yellow, then white, yellow, etc.). This should still be the case no matter how many items are removed or added and no matter what order these operations are done in. You may not use the CSS3 nth-child pseudo selector to do this. The code should work for multiple clicks of the buttons. On each click it should clear any previous information you typed in the input boxes. Do not use any JavaScript libraries such as jQuery or Prototype.

Here is the relevant HTML code for the page:

```
<h1> My super nifty to-do list </h1>
<ul id=”list”></ul>
<div>
  <input type = ‘text’ id=’item’/>
 <button id=’add’>add</button>
</div>
```

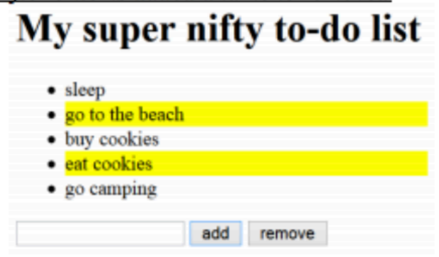
The screen shots shows the state after items have been added, and the state after items have been removed.

1. *Before anything has been added*



My super nifty to-do list

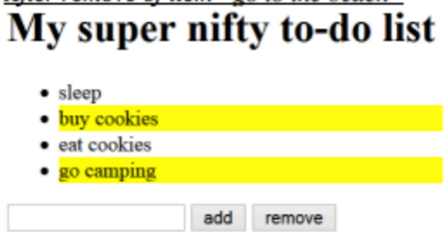
2. *After 5 items added and none removed*



My super nifty to-do list

- sleep
- go to the beach
- buy cookies
- eat cookies
- go camping

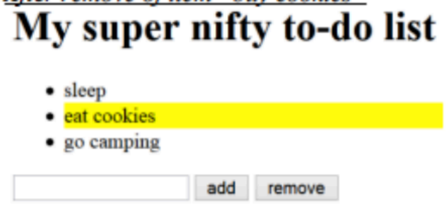
3. *After remove of item “go to the beach”*



My super nifty to-do list

- sleep
- buy cookies
- eat cookies
- go camping

4. *After remove of item “buy cookies”*



My super nifty to-do list

- sleep
- eat cookies
- go camping