

Maharishi University of Management, CS472

Lab : Decorate My Text

Except where otherwise noted, the contents of this document are Copyright 2012 Marty Stepp, Jessica Miller, and Victoria Kirst. All rights reserved. Any redistribution, reproduction, transmission, or storage of part or all of the contents in any form is prohibited without the author's expressed written permission.

original lab idea and code by Victoria Kirst and Jeff Prouty; revised by Brian Le, Katlyn Edwards, Roy McElmurry IV, and Marty Stepp



Today's lab

Today you'll write a page where the user can type text into a box, and by clicking on UI controls, the user can decorate the text by giving it funny styling.

The HTML page `decoratemytext.html` contains a basic HTML shell and page header. This skeleton already links to a CSS file `decoratemytext.css` that defines all the styles you need. You do not have to write any CSS code today.

You will write a JavaScript file `decoratemytext.js` that will manipulate the text.

Download the HTML file below (right-click, Save Target As...) to get started:

-  [decoratemytext.html](#)

Exercise : Create UI Elements (~15 min)

(See example screenshot on next slide.)

The first task is to expand `decoratemytext.html` by adding UI controls. Add HTML code for the following:

- A field for users to enter large (multi-line) amounts of text.
 - It should be 4 rows by 30 columns in size.
 - Wrap it in a bordered field set box with the label "Text".
- Add a second bordered field set box labeled "Decoration" that contains the following controls:
 - A button labeled: Bigger Decorations!
 - A checkbox labeled "Bling"

- NOTE: Controls are sometimes used in forms; but you must not use the `form` tag on your page.

Exercise , output

Your page should look like this:

Text Decorator



Text

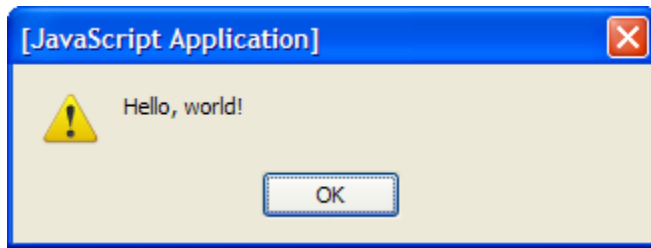
```
Sample Item 1  
Sample Item 2
```

Decoration

☐ Bling

Exercise : JavaScript `alert` (~5 min)

Now you'll write a bit of JavaScript testing code that pops up an `alert` box. This is just a test to make sure that your browser is running your JavaScript file, before we move on to tougher exercises.



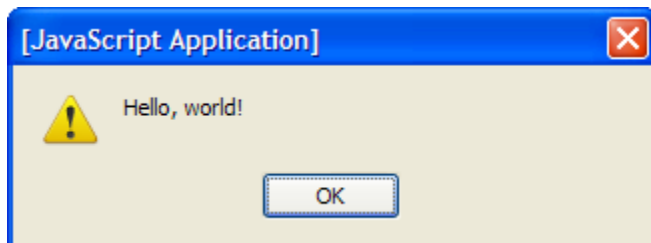
- Create a new file and save it as `decoratemytext.js`.
- Put the following line of code into the file:

```
alert("Hello, world!");
```

- Link your HTML page to your JavaScript file using a `script` tag.
- Refresh your page in the browser. Do you see the `alert` message?
 - If so, move on.
 - If not, double-check your `script` tag syntax or ask a TA for help.

Exercise : Hello World Button (~10 min)

Now let's set up a very basic JS event handler. Modify your JS code and HTML so that the "Hello, world!" `alert` message won't pop up until the user clicks the "Bigger Decorations!" button.



- Modify your JS file to wrap the `alert` into a function.
- Add an `onclick` event handler for the "Bigger Decorations!" button that calls your new function.
- Refresh your page in the browser. Click the button. Do you see the `alert`?
 - If so, move on.
 - If not, remember that the DOM may not be loaded yet, add an `onload` handler to the global `window` object.
 - Still not, double-check your `onclick` tag syntax and function, or ask a TA for help.

Exercise : Bigger Decorations Button (~15 min)

(See example screenshot on next slide.)

Modify your JS code so that when the user clicks "Bigger Decorations!", the text in the text area will get larger.

- Make sure your text area has an `id` attribute so your JS code can talk to it.
- Modify your JS function so that it now changes the text area's font size to **24pt**.
- Use proper units, such as "4em", and proper naming conventions.

Exercise , output

The text should look like this after the button is clicked:



Text

Sample Item 1
Sample Item 2

Decoration

☐ Bling

Exercise : Bling Checkbox (~15 min)

(See example screenshot on next slide.)

Add an event handler so that when the user checks "Bling", the text area will receive some styles.

- Add an `onchange` on the checkbox that calls a function that pops up an `alert`.
- Modify your new function to set the text area's font weight to **bold**.
 - You can see if a checkbox is checked by examining its `checked` attribute. (Give it an `id`.)
 - When the box is unchecked, the font weight should go back to normal.

- Once the bold part works, add the following styles to the text when the box is checked:
 - change its color to green
 - underline the text (this is the CSS `text-decoration` property)

Exercise , output

Your page should look like this when the box is checked:

Text Decorator



Text

Sample Item 1
Sample Item 2

Decoration

Bigger Decorations!

☒ Bling

Exercise : Font Timer (~10-15 min)

- Make it so that when the "Bigger Decorations!" button is clicked, rather than setting the font size to 24pt, you'll make it **2pt larger** than its current size.
 - Read the font size such as "12pt" and change it to a larger one like "14pt".
 - You may want to use the `parseInt` function to help you solve this.
- Once that works, make it use a **timer**.
 - Now a single click should continually increase the font size by +2pt every 500 ms.
 - Use the `setInterval` function. The timer should call the function you just wrote.