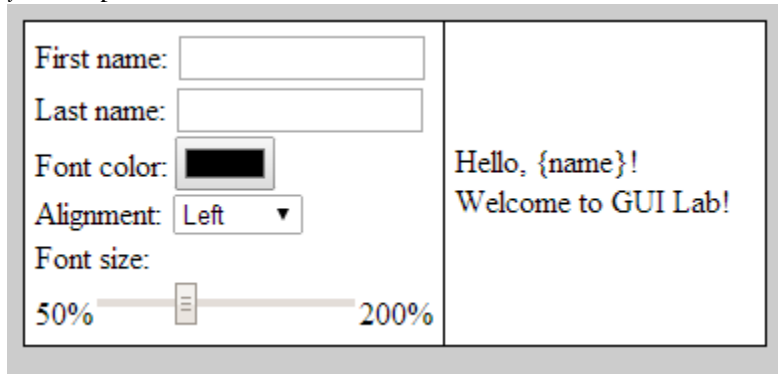


## Project 1A: Hello {name}

due 9:00am Tuesday, September 9, 2014.

### Goal

The goal of this first project is to get an HTML5 workflow setup and do some basic input handling with javascript.



First name: <input type="text"/>	Hello, {name}! Welcome to GUI Lab!
Last name: <input type="text"/>	
Font color: <input type="color"/>	
Alignment: <input type="text" value="Left"/>	
Font size:	
50% <input type="range"/> 200%	

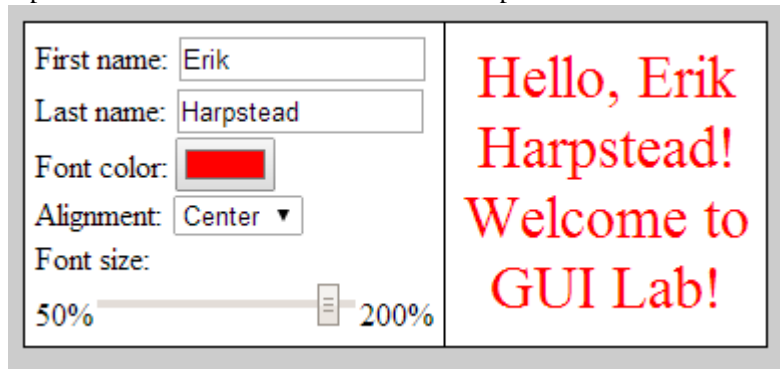
Your task is to replicate the interface shown above. When the user first starts the program, both of the first and last name fields are empty, the font color should be set to black, the text alignment should be to the left, and the font size should start at 100%. The two sections of the interface are contained in 1 pixel black borders with no space between cells and set on a grey (#cccccc) background. All content within each cell has 5 pixels of padding from the edge.

When a user inputs either their first or last name the “{name}” in “Hello, {name}!” should change to match. A space should appear between the first and last names only when both of them are not blank. If the user clears both name fields the string should return to “Hello, {name}!”

When the user selects a new color or alignment option the text in the box should change to reflect the new user choices. The standard color selector is sufficient for color choices. The alignment should include options for left, right, centered, and justified.

When the user moves the font size slider the text in the box should change to reflect the user choice. The slider should range from 50% to 200% and default to 100%. The slider should provide a reasonable resolution so users have a degree of fine control but can also notice the effect of a change.

Here is an example of the interface after some user input:



The screenshot shows a web form with the following elements:

- First name: Erik
- Last name: Harpstead
- Font color: Red
- Alignment: Center
- Font size: 50% to 200% (slider)

The output text on the right is:

Hello, Erik  
Harpstead!  
Welcome to  
GUI Lab!

For every project, you will be required to leave your name in a comment. To create a comment in your Hello {name} file, below the first line (the one that reads `<!DOCTYPE HTML>`), type in: `<!-- (your name) -->`. If you are using the default template a comment is added in this location by default just replace the default text with your name.

## Provided Files

The files provided include an `index.html` file, a `main.css` file and a `main.js` file.

The provided HTML file contains both a `<head>` and a `<body>`. Inside the `<head>` there are 2 `<meta>` tags, a `<title>` and a `<link>` tag. The `<meta>` tags should be left as is, they define the character set for the file and tell the browser to use a normal full screen scaling. The `<title>` should be changed to “Project 1 - {your name}”. The `<link>` tag includes a reference to the `main.css` file and should be left as is.

The `main.css` file defines the styling of the interface. The provided file already contains styling to draw the border around the “target” cell. You will have to add further code to this file to change the background and styling for both cells.

The `<body>` contains a `<table>` with a single row `<tr>` and 2 cells `<td>`. The first cell (`id=“controls”`) should be where you place your input fields. The second cell (`id=“target”`) should contain the text being edited. The “target” cell already contains 2 `div`s with the initial text for you to use. The `<body>` also contains a `<script>` reference to the `main.js` file, which should be left as is.

The `main.js` file is where you should write your javascript code to respond to user input. Note that this script starts with the line `“use strict”`; This line tells the javascript compiler to take a strict interpretation of your code. This means a number of things that are normally allowed in javascript will be considered errors (for a full list of what is outlawed see here: [http://www.w3schools.com/js/js\\_strict.asp](http://www.w3schools.com/js/js_strict.asp)). All javascript in this class will be run with strict mode enabled. If the `“use strict”`; line is missing from a submission it will be added back in before testing and grades will be assigned based on the resulting behavior.

## Hints

- You may want to look at the different kinds of form input fields available in HTML5 ([http://www.w3schools.com/html/html\\_forms.asp](http://www.w3schools.com/html/html_forms.asp) & [http://www.w3schools.com/html/html5\\_form\\_input\\_types.asp](http://www.w3schools.com/html/html5_form_input_types.asp))
- In CSS style names are usually written with a dash like (e.g. `text-align:left`), when accessing a style through javascript it is normally written with camelCase (e.g. `style.textAlign = "left"`).

## Tentative Rubric

Turned in & compiles:	5 points
All elements are present:	5 points
{name} text updates correctly with first and last name:	5 points
Text color updates correctly:	5 points
Text alignment updates correctly:	5 points
Text size updates correctly:	5 points
Cell borders and backgrounds are correct:	5 points
Cell spacing is correct:	5 points
Background color is grey (#cccccc):	5 points
Your name is in a comment:	5 points
<hr/>	
Total	50 points

## Critical Thinking

You are not required to turn in answers to any of the questions in this section, but we recommend that you explore and think about some of the questions.

1. You may have noticed in the W3schools documentation that some of the inputs in this project are not compatible with all browsers (e.g. the color chooser). How could you accomplish a similar functionality using different input elements?
2. There are a number of different events that could be used to accomplish the interactions in this project. Why did you choose the ones that you did, which ones are most appropriate for maximizing the responsiveness of the interface?
3. Using a `<table>` for layout like this is considered to be bad style by some web developers. We mainly use it because it is simple to implement without much effort. How else could you accomplish the desired design without using a `<table>`?

## Turning it in

Project 1a is due by 10:30am September 9th, 2014 as a zipped file. Email your file to Erik at [eharpste@cs.cmu.edu](mailto:eharpste@cs.cmu.edu). Late entries will be penalized -5% for every late day.