

Module 1 Coding Assignment

[Click here to SIGN UP for the *Single Page Web Applications with AngularJS* course on Coursera.](#) It's FREE!

CODING TIME!!! You are able to code up your very first AngularJS application! How exciting is that?!

Time to Complete

It should take about an 1 hour or less.

Ask questions in [Week 1 Discussion Forum](#) if you get stuck! We are all learning, and going through getting stuck and then unstuck (even with someone's help) can be a very valuable learning experience!

DO NOT be scared by the length of this assignment! It's really not so much at all. I just wanted to explain everything as clearly as I could and break it down into smaller steps for your benefit.

Assignment Instructions

General Idea

The idea of this assignment is to create a front-end application that presents the user with a textbox where they can list comma-separated items they usually eat for lunch. Once that's entered, the user has to click the "Check If Too Much" button.

If the number of items in the textbox is less than or equal to 3 (e.g., 1, 2, or 3), a message should show up under to the textbox saying "Enjoy!". If the number of items is greater than 3 (4, 5, and above), the message "Too much!" should show up under the textbox. (*Hint:* To implement this behavior you can utilize the `split` method. [See documentation for that method](#))

If the textbox is empty and the user clicks the "Check If Too Much" button, the message "Please enter data first" should show up. 'Empty' here means either `" "` (empty string) or a string with just spaces in it. (Hint: AngularJS `ng-model` already performs the trimming for you, so there shouldn't be anything you need to do.)

Only 1 message should be shown at any given time. In other words, if you have both messages "Enjoy!" and "Too much!" showing up at the same time, it's an error.

You are not required to handle a case where there is no item between some commas. For example, you are free to consider this case `item 1, item2,,item3` or this case `item 1, item2, ,item3` as 4 items in the list. However, you can implement checking for an 'empty' item as a bonus feature (ungraded). Please make sure to put a comment somewhere next to the input textbox stating that you do NOT consider and empty item, i.e., `, ,` as an item towards to the count.

Rules

Breaking one of these rules will cause you to fail the assignment: * You are not allowed to use regular HTML `onclick` attribute to bind behavior to the button. You **must** use the AngularJS way of binding behavior. * At no point should your Javascript code look up *anything* in the DOM of the browser.

Steps

Here is what you will need to do to complete the assignment:

1. (If you haven't already) Create a GitHub.com account and a repository that you will use for this class.
2. (If you haven't already) Follow the Development Setup Video (beginning of Module 1) instructions on how to create a repository and set it up such that you can host and view your finished web pages on GitHub Pages, i.e., GitHub.io domain name. You will need to provide that URL for your peer review.
3. Create a folder in your repository that will serve as a container folder for your solution to this assignment. You can call it whatever you want. For example, `module1-solution` or `mod1_solution`, etc.
 - You can do this by 'cloning' your repository with `git clone https://www.github.com/your_repo_url` to your local machine, creating `module1-solution` folder in the root of the repository directory along with a README.txt inside of the `module1-solution` directory. Then, you would do `git add .`, followed by `git commit -m 'New folder'`, followed by `git push` to upload the new folder with the README.txt to the GitHub repository.
4. HTML/CSS for the assignment
 - Option 1: Copy the **contents** of the folder `assignment1-starter-code` into the newly created folder from the previous step. If you cloned this repository, the assignment 1 folder is located in `root_dir_of_your_local_repo/assignments/assignment1/assignment1-starter-code`
 - Option 2: Create the HTML/CSS yourself. Make sure to name the HTML file `index.html`. The only requirement is that your HTML have a textbox, a button, and a region where you will output the message. The rest is up to you.
5. Import AngularJS into your project and place a `<script>` tag right before the `</body>` tag.
6. Declare `ng-app` either on the `html` or the `body` element. Name your app `LunchCheck`.
7. Create `app.js` in your project and declare an Angular module to match your `ng-app` declaration.
8. Go back to `index.html` and declare a controller for some portion of your page that contains the

textbox, the button, and the message placeholder.

9. Annotate the textbox, the button, and the placeholder such that you can hook in behavior to those elements from your controller.
10. Go back to `app.js`. Declare and define a `LunchCheckController`. Properly inject `$scope` into the controller using the `$inject` property (shown how in video lecture) to make sure to protect your code from minification.
11. Create and implement properties and method(s) in order to implement the functionality outlined in General Idea section above.
 - Make sure that none of your variables/objects/functions "leak to the global scope". (Hint: IIFE)
12. After you are done and satisfied with your solution, don't forget to add/commit/push your code to your repository.

IMPORTANT REMINDERS:

- Closely follow the submission guidelines for this assignment on Coursera.org
- Make **sure** you provide the correct URL in your submission (it should be `GitHub.io`, *not* `GitHub.com`)
- Make **sure** to TEST your assignment not just on your local machine, but ALSO once you deploy it on GitHub, using the URL you are providing as part of your submission.
- This assignment will be peer-reviewed (and graded). The guidance will be given such that if submission instructions are not followed, the assignment is to be failed. This includes providing the wrong URL for your deployment. Following instructions is very much part of software development. After all, that's what software requirements are - instructions to follow.

BONUS (OPTIONAL AND NOT GRADED)

The first 2 of these can be done in a more advanced way, but can also be done with just what we've learned so far! * If the message is "Enjoy!" or "Too much!", make the font color green. If the message is "Please enter data first", make the font color red. * If the message is "Enjoy!" or "Too much!", make the border color around the textbox green. If the message is "Please enter data first", make the border color around the textbox red. * Implement this case `item 1, item2,,item3` or this case `item 1, item2, ,item3` as not counting an 'empty' item towards the count of how many items there are in the list. Please *make sure* to put a comment somewhere next to the input textbox stating that you do NOT consider an empty item, i.e., `, ,` as an item towards the count, so whoever is grading your assignment doesn't erroneously mark that as an error.