WEEK 4 ASSESSMENT RUBRIC

Guidelines for Grading

Below is an answer sheet I use to grade the week 4 assessment. Please note, when it comes to syntax/spelling/grammar, the answer should be marked wrong if the code will not compile or execute properly.

JAVASCRIPT**/**JQUERY

1. To start, the answer must require what IIFE stands for: Immediately Invoked Function Expression. It also must include any reason(s) why we use them. Here is a list of acceptable answers:
   1. Self-containing function that creates a closure around the code
   2. Keeps the global scope from being polluted with variable names
   3. Executes itself
2. Requires three reasons why using jQuery is beneficial. Here is a brief list of acceptable answers:
   1. Lightweight
   2. Great documentation
   3. Cross-browser compatible
   4. Massive community
      1. Which also allows for a large number of plugins
   5. Easy DOM manipulation
   6. Reduces the amount of code needing to be written
3. Requires three separate lines of code to be written.
   1. $(‘#my-info’)
   2. $(‘.hero’)
   3. $(‘li’)
4. The answer to this must be functional. There cannot be another variation of this, simply due to it specifically asking for the .on() method to be used.

$(‘.btn’).on(‘click’, function() { }); -- rewards 1 pt

$(‘#main-form’).slideToggle(); -- rewards 1 pt

The final answer is:

$(‘.btn’).on(‘click’, function() {

$(‘#main-form’).slideToggle();

});

1. This answer is going to require an explanation of why the two are different. At the core, GET is a method that makes a request to only retrieve information. POST is a method that makes a request that has additional information being sent.
2. Three steps for this answer.
   1. $.get(‘<http://www.example.com/books.json>’).done(function(data) { }); -- rewards 1 pt
   2. var response = data -- rewards 1 pt
   3. console.log(response.books[1].title); -- rewards 1 pt

$.get(‘<http://www.example.com/books.json>’).done(function(data) {

var response = data;

console.log(response.books[1].title);

});

ANGULAR

1. Two part answer. The first part requires a snippet of HTML. The second part requires a snippet of JavaScript.
   1. <html ng-app=”myApp”> -- rewards 1 pt
   2. var app = angular.module(“myApp”, []); -- rewards 1 pt
2. Must use ng-model on an input.
   1. <input type=”text” ng-model=”username”>
3. This answer is going to be a brief explanation of what the filter is doing. The filter is going to list the items from the friendsList in order alphabetically.
4. This answer is going to be a brief explanation of what is wrong with the ng-repeat.
   1. The counter/parameter “user” is not being used within the Angular expression inside the <li> element. As long as they can get that idea across, it should be acceptable.
5. Two part answer. First, they must create the controller. Second, they must assign a property to the scope called greeting which is initialized to “Hello World”.
   1. app.controller(“myController”, function($scope) { }); -- rewards 1 pt
   2. $scope.greeting = “Hello World”; -- rewards 1 pt

app.controller(“myController”, function($scope) {

$scope.greeting = “Hello World”

});

1. Requires the student to write out the two service providers from Angular. Most people will/should answer with A and B
   1. Service -- rewards 1 pt
   2. Factory -- rewards 1 pt
   3. Provider -- rewards 1 pt
2. This should be a brief explanation that describes differences between template or templateUrl. A template is simply just a string of HTML written to be rendered within a directive. This can be any amount of HTML. A templateUrl is a path to a .html document that will be rendered within a directive.
3. Requires two parts to be finished.
   1. $http({

url: “<https://www.example.com/books.json>”,

method: “GET”

}) -- rewards 1 pt

* 1. .then(function(response) {

console.log(response);

}) -- rewards 1 pt

$http({

url: “<https://www.example.com/books.json>”,

method: “GET”

}).then(function success(response) {

console.log(response);

});

1. This will require two answers. Note, this is only worth 1 pt.
   1. templateUrl: “views/band.html”
   2. controller: “bandmembersController”
2. This is a straight-forward answer. A B or C can be acceptable answers
   1. Service
   2. Factory
   3. Provider