CP1295
Advanced JavaScript
Assignment 05 v1

CP1295 Assignment 05

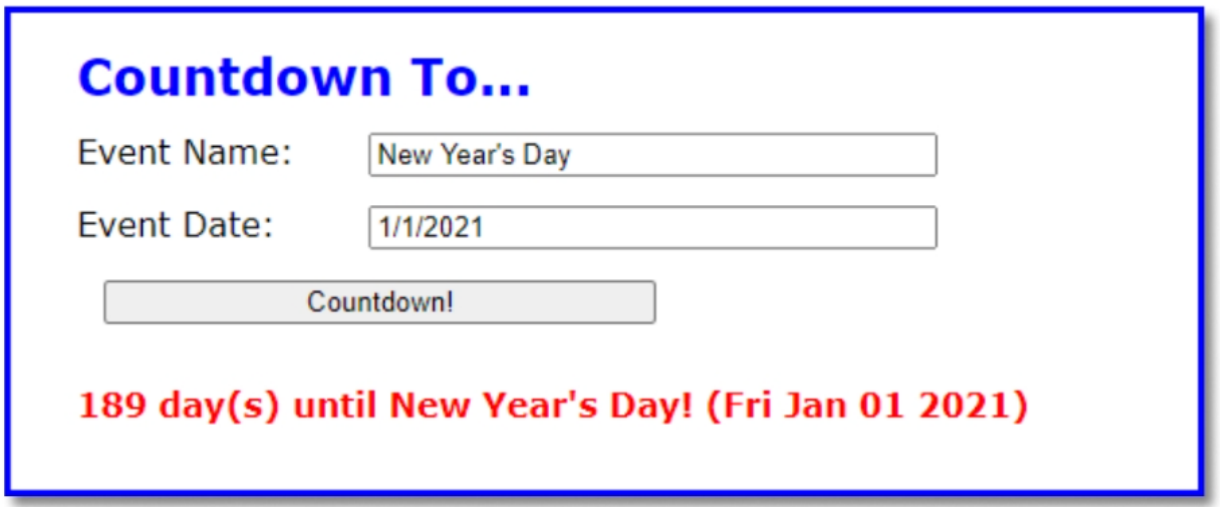
Chapter 16

P558

Exercise 1601

Enhance the Countdown application to use classes and libraries.

This exercise guides you through the process of changing a procedural Countdown application that uses functions to organize its code to an object-oriented application that uses classes and objects to organize its code.



Open, test, and review the application

1. Use your text editor or IDE to open the application in this folder:
   `\javascript_jquery\exercises\ch16\countdown\`
2. Run the application in Chrome and test it. To do that, enter an event name, enter an event date, and click the Countdown button.
3. Review the code in the click() event handler for the Countdown button. Note that the click() event handler contains most of the code for this application. As a result, the only way to reuse this code is to copy it, which isn't a good practice.

Add two JavaScript library files

4. Add two JavaScript files to the application. Name them lib_event.js and lib_validation.js.
5. In the index.html file, add script elements to include these new files. Since the main JavaScript file will use these files, make sure to code the script elements for the library files before the script element for the main file.

Use an Event class to define an Event object

6. In the event library file, code a class named Event that has a constructor that accepts two parameters: name and dateString. Store the values of these parameters in properties of the same name.
7. Add another property named date. Set this property to a new Date object that's created from the dateString parameter.
8. Switch to the click() event handler for the Countdown button. Then, find the code that gets the event name and date from the user and modify this code so it creates an Event object from the Event class. To do that, pass the event name and date string entered by the user to the constructor of the Event class.
9. Replace the code in the click() event handler that uses the name, dateString, or date variables with code that uses the name, dateString, or date properties of the Event object.
10. Test the application to make sure it still works correctly.
11. In the Event class, add a read-only accessor property named days. Then, move the code that calculates the number of days from the click() event handler to this property, and make sure the property returns the calculated number of days.

12. In the click() event handler, modify the code so it uses the days property of the Event object instead of the days variable.
13. Test the application to make sure it still works correctly.
14. In the Event class, add a method named getCountdownMessage(). Then, move the code that displays the countdown message from the click() event handler into this method. Make sure to return a string that contains the countdown message.
15. In the click() event handler, modify the code so it uses the getCoundownMessage() property of the Event object to get the countdown message. Then, display the countdown message on the web page.
16. Test the application to make sure these changes work correctly.

Move the validation code to the validation library file

17. In the validation library file, code an object literal named validation with four methods named isEmpty(), hasNoSlashes(), isInvalidYear(), and isInvalidDate(). Each method should accept one parameter named val.

18. Code the isEmpty() method so it returns true if the string argument it receives is empty. Otherwise, return false.
19. In the click() event handler, find the code that checks the length of the event name and date string, and replace that code with calls to the isEmpty() method of the validation object.
20. Test the application to make sure it still works correctly.
21. Repeat the previous three steps for the hasNoSlashes(), isInvalidYear(), and isInvalidDate() methods of the validation object. When you're done, the click() event handler should use these methods to check for slashes, a 4-digit year, and an invalid date.

Grading Rubric
CP1295 A05

Check Points			100
4	Add two JavaScript files. Name them lib_event.js and lib_validation.js	2	
5	In the index.html file, add script elements to include these new files	4	
6	In the event library file, code a class named Event that has a constructor that accepts two parameters: name and dataString	4	
7	Add another property named data. Set this property to a new data object that is created from the dateString parameter	4	
8	Switch to the click() handler for the Countdown button. Find the code that gets the event name and date from the user and modify it's code so it creates an Event object from the Event class	4	
9	Replace the code in the click() Event handler that uses the name, dateString, or date variables with code that uses the name, dateString, or date properties of the Event object	4	
10	Test the application to make sure it still works correctly	10	
11	In the Event class, add a read-only accessor property named days. Then, move the code that calculates the number of days from the click() handler to this property, and make sure the property returns the calculated number of days.	4	
12	In the click() event handler, modify the code so it uses the days property of the Event object instead of the days variable.	4	
13	Test the application to make sure it still works correctly	10	
14	In the Event class, add a method named getCountdownMessage(). Then move the code that displays th e countdown message from the click() event handler into this method. Make sure to return a string that contians the countdown message	4	

15	In the click() event handler, modify the code so it uses the getCountdownMessage() of the Event object to get the countdown message. Then display the countdown message on the web page	4
16	Test the application to make sure it still works correctly	10
17	In the validation library file, code an object literal names validation with four methods named isEmpty(), hasNoSlashes(), isInvalidYear(), and isInvalidDate(). Each method should accept one parameter named val	4
18	Code the isEmpty() method so it returns true if the string argument it receives is empty. Otherwise, return false	4
19	In the click() event handler, find the code that checks the length of the event name and data string, and replace that code with calls to the isEmpty() method of the validation object	4
20	Test the application to make sure it still works correctly	6
21	Repeat the previous three steps for the hasNoSlashes(), isInvalidYear(), and isInvalidDate() methods of the validation object. When you are done, the click() event handler should use these methods to check for slashes, a 4-digit year, and an invalid date	4
END TEST	Test the application to make sure it still works correctly	10