In class exercises – JavaScript

Array Iteration:

Iterate through an array of string values top print products.

Let's printing out product names and prices on an invoice, then total the prices and print them at the bottom. In your JavaScript file add the code to satisfy the requirements below — each of these marks a place where you have to add something to the code. They are as follows:

1. Below the `// number 1` comment are a number of strings, each one containing a product name and price separated by a colon. We'd like you to turn this into an array and store it in an array called `products`.
2. On the same line as the `// number 2` comment is the beginning of a for loop. In this line we currently have `i <= 0`, which is a conditional test that causes the for loop to only run once, because it is saying "stop when `i` is no longer less than or equal to 0", and `i` starts at 0. We'd like you to replace this with a conditional test that stops the loop when `i` is no longer less than the `products` array's length.
3. Just below the `// number 3` comment we want you to write a line of code that splits the current array item (`name:price`) into two separate items, one containing just the name and one containing just the price. If you are not sure how to do this, consult the Useful string methods article for some help, or even better, look at the Converting between strings and arrays section of this article.
4. As part of the above line of code, you'll also want to convert the price from a string to a number. If you can't remember how to do this, check out the first strings article.
5. There is a variable called `total` that is created and given a value of 0 at the top of the code. Inside the loop (below `// number 4`) we want you to add a line that adds the current item price to that total in each iteration of the loop, so that at the end of the code the correct total is printed onto the invoice. You might need an assignment operator to do this.
6. We want you to change the line just below `// number 5` so that the `itemText` variable is made equal to "current item name — $current item price", for example "Shoes — $23.99" in each case, so the correct information for each item is printed on the invoice. This is just simple string concatenation, which should be familiar to you.

Array Methods

To complete the app, we need you to:

1. Add a line below the `// number 1` comment that adds the current value entered into the search input to the start of the array. This can be retrieved using `searchInput.value`.
2. Add a line below the `// number 2` comment that removes the value currently at the end of the array.

Selection

In this example, you are going to finish a simple calendar application. In the code you've got:

- A `<select>` element to allow the user to choose between different months.
- An `onchange` event handler to detect when the value selected in the `<select>` menu is changed.
- A function called `createCalendar()` that draws the calendar and displays the correct month in the `<h1>` element.
- We need you to write a conditional statement inside the `onchange` handler function, just below the `// ADD CONDITIONAL HERE` comment. It should:

    1. Look at the selected month (stored in the `choice` variable. This will be the `<select>` element value after the value changes, so "January" for example.)
    2. Set a variable called `days` to be equal to the number of days in the selected month. To do this you'll have to look up the number of days in each month of the year. You can ignore leap years for the purposes of this example.

Hints:

- You are advised to use logical OR to group multiple months together into a single condition; many of them share the same number of days.
- Think about which number of days is the most common, and use that as a default value.

Event Object Variables

The button in the box is a submit button for a form. This will automatically submit to the URL specified in the action attribute. Because this is a dummy form, we don't actually want it to submit to the URL provided, we want the form to do something else. So, first thing we want to do is to prevent the default behavior.

1. Using the event object variable, call the `preventDefault` function.
2. Refresh the page to see what happens.
3. Using the event object variable again, use the target property to change the background color to a random color.