Given: `var snow = new Array(6, 8, 8, 12, 21, 15, 7, 8, 10, 15);`

24. Add code to Exercise 23 to find and display the years with the highest and lowest snowfall amounts.
25. Add code to Exercise 24 to find the average snowfall over the 10 years.
26. Add code to Exercise 25 to find the number of years that snowfall was above the average.
27. Add code to Exercise 26 to find the number of years that snowfall was within two inches (above or below) the average.
28. Add code to the `addRings()` function of Example 8.14 in the chapter to check if a value Jackie wants to add to her inventory is already part of the inventory.
29. Add code to Example 8.14 in the chapter that will allow Jackie to enter her inventory of pendants and to add or subtract from that inventory.
30. Given the following JavaScript code, what are the values of the eight elements in the `numbers[]` array?

```
var numbers = new Array;
for (var i = 0; i < 8; i++)
        numbers[i] = (i * i) + 1;
```

Programming Challenges

On Your Own

1. Create a web page that will display statistics and calculate the standard deviation on a list of numbers input by the user. The display on the web page should include the following:

   - the entire array of numbers, as entered by the user
   - the mean (average) of the numbers
   - the standard deviation

   Helpful information: Assuming you name your variables as shown, the formulas you need are listed below the variables:

   array of numbers: **nums**
   standard deviation: **stdDev**
   number of elements in the array: **N**

   two sums: **sum1** and **sum2**
   mean: **mean**
   counters as needed: **count**, **i**, **j**, **k**, etc.

   Formulas:

   The formula for the standard deviation of a list of **N** numbers in an array named **nums** with a mean of **mean** is the square root of

   $((nums[0] - mean)^2 + (nums[1] - mean)^2 +...+ (nums[N-1] - mean)^2)/(N-2)$

- to sum the numbers to calculate the mean: **`sum1 = sum1 + nums[count]`**
- to calculate the average: **`mean = sum1/(N)`**
- to calculate the first part of the standard deviation, find the differences between each number, **`nums[count]`** and the **`mean`** and then square the result of each of these differences. First, find the square of each difference, using a loop:

  `(nums[count] - mean)`^{2}

  then sum all the differences so, in a loop, use

  `sum2 = sum2 + (nums[count] - mean)`^{2}

  and finally, the standard deviation is calculated using the `Math.sqrt()` method:

  `stdDev = Math.sqrt(sum2/(N - 1)`

Save the page with the filename `std_dev.html` and submit your work as instructed by your teacher.

2. Add code to Example 8.14 to allow Jackie to create an inventory of pendants and to add or subtract from that inventory using Example 8.14 as a model. Save the page with the filename `pendant.html` and submit your work as instructed by your teacher.
3. Create a web page that will allow the user to customize the page. The user should be prompted to enter the following information which will be displayed on a new page, along with the customizations the user has chosen.

   User information: name, nickname, email address, favorite movie, favorite book, favorite type of music

   User-selected customizations: page background color, text color, text size

   You can add any features you wish and allow the user to customize each item differently if you want. Save the page with the filename `customize.html` and submit your work as instructed by your teacher.
4. Create a web page for a business. The page should include parallel arrays which hold names of the items sold and the corresponding price for each item. The user should be prompted to select an item and a quantity and the page will then display the total cost for that quantity of the item chosen. You can create your own inventory or use the following suggestions. Save the page with the filename `supplies.html` and submit your work as instructed by your teacher.

Item	**Price**	**Item**	**Price**
notebook	$ 5.95	laptop case	$ 29.99
pen	4.95	cell phone case	18.99
mechanical pencil	2.95	3-ring binder	6.95
lead refill pack	.98	3-hole paper refill	2.00

5. Create a web page that will allow a teacher to enter his or her students' names and test scores into parallel arrays. The page should prompt the teacher for the number of students in the class and the number of tests administered. Use this information to create the appropriate number of parallel arrays. Then create the

arrays and prompt the teacher for data to populate the arrays. Display the data on the web page in a table. Also display the high and low score for each test as well as the average for each test. Save the page with the filename `class_parallel.html` and submit your work as instructed by your teacher.

6. Redo Programming Challenge 5 but use a two-dimensional array instead of parallel arrays for the data. Save the page with the filename `class_two_d.html` and submit your work as instructed by your teacher.

7. A magic square is a two-dimensional array of positive integers in which the following is true:

ote
Two-Dimensional Array
ır_Own_7_MagicSquare

- The number of rows equals the number of columns.
- Every row, column, and the two diagonals add up to the same number.

Create a web page with a table that has four rows and four columns. Allow the user to enter values for each cell. Store these values in a two-dimensional array named **`magic`** and determine if it is a magic square.

Note: if we call the array **`magic`** then the sums of the two diagonals are as follows:

```
diagonal1 = magic[0][0] + magic[1][1] + magic[2][2] + magic[3][3];
diagonal2 = magic[0][3] + magic[1][2] + magic[2][1] + magic[3][0];
```

Save the page with the filename `magic.html` and submit your work as instructed by your teacher.

Case Studies

Greg's Gambits

Now you will add a new game to the `Greg's Gambits` site. The game will be a maze that a player must navigate, while avoiding pitfalls, to save a victim. It will be helpful if you work through the `Game of 15` as discussed in this chapter as well as the section on image swapping.

Create a web page using the `Greg's Gambits` template. Put a table into the `"content"` area. This table will hold the maze and each cell will be a step that the player can take. You will fill each cell with an image file; you can use the images provided in the Student Data Files or find your own. We will continue the instructions for this game assuming that you are using the files provided: `myHero.jpg`, `saved.jpg`, and `blue.jpg`.

First, create a small 4 × 4 table. After the program is coded and works properly, you can add more rows and columns to make a more challenging game. Next, write the JavaScript code.

Create a two-dimensional array of images and, initially, store the `myHero.jpg` image in element `[0][0]` and the `saved.jpg` image in element `[3][3]`. All other cells should contain the `blue.jpg` image.

Next, create the pitfalls. Use a random number generator to identify several cells. At first, start with three pitfalls; you can add more later. You should generate a random value for a row and a column to identify a pitfall cell. You will also need to check to be sure you have not identified the first cell, where the player begins, or the last cell, where the victim awaits, and that the cells you identify do not create an impasse (for example, the three identified pitfalls cannot block the player from moving).

Allow the player to move to any cell above, below, to the left, or to the right. Diagonal moves are not allowed unless you want to get fancy.

Each time the player clicks on a cell (attempts to make a move) you need to check the following:

- Is the desired cell a pitfall? If it is, an alert should tell the player about the pitfall and tell him to make a different move. You may, if you want, create "killer" pitfalls which can end the game.
- Is the desired cell the last cell in the table? If it contains the `saved.jpg` image, the player has won and should be informed and the game ends.
- If the selected cell is not a pitfall and not a winner, swap the cells and the game continues.

Be sure to give your player the option to replay the game.

Note: You can "hard code" the pitfalls and put them in whatever cells you choose. Using randomly generated pitfalls makes the game more interesting and also makes it easier to enhance the game by increasing the size of the maze.

Save your page as `greg_maze.html`. Open the `index.html` page for `Greg's Gambits` and add a link, under the `Play A Game` link, that links to this page named `Greg's Maze`. Submit your work as instructed by your teacher.

Carla's Classroom

In this exercise, you will polish the `Images and Imagination` page created in the chapter. Therefore, you will need to create the slide show page as instructed in the text. Now add code to this page to extract the story a student has written about one of the images in the slide show. A new page should be created so the student can save a copy of the image selected and his or her story. Allow the student to select a background color for the new page and a text color. Prompt the student for his or her name and for color choices. Use arrays for the color choices and be sure to include a check to ensure that a student does not pick the same color for the page background and text. The page should have the same filename as in the chapter, `carla_slideshow.html`. Submit your work as instructed by your teacher.

Lee's Landscape

Add a page to the `Lee's Landscape` website that will allow the customer to select a service and see the options available for that service. Use a two-dimensional array with six rows and five columns to store the information given in the table below. The page should display a list of services. When the customer selects a service, the options should be displayed on the web page.

Services	Options			
lawn mowing	weekly	twice a month	monthly	by call
hedge trimming	weekly	twice a month	monthly	by call
mulch	buy per square yard	delivery	spreading	full yard discounts
pest control	monthly	twice a year	yearly	by call
weed control	weekly	twice a month	monthly	by call
extra services	sod installation	irrigation systems	yard cleanup	ponds, pools, & streams

Be sure to give this web page an appropriate page title such as `Lee's Landscape || Services`. Save this file with the filename `lee_services.html`. Add a link to the `Lee's Landscape` home page to this new page. Submit your work as instructed by your teacher.

Jackie's Jewelry

Add a slide show page to the `Jackie's Jewelry` website. The page should highlight Jackie's creations. Each image should have a caption that describes the piece. You can use the images in the Student Data Files or you can find your own images online. Remember that if you use an image from a website, you must credit the owner of the image on your web page. Be sure that you size all the images before creating the slide show. If the images are not the same size, your slide show will not display properly.

You may find it helpful to review the `Images and Imagination` page discussed in this chapter before you begin to create the slideshow.

Be sure to give this web page an appropriate page title such as `Jackie's Jewelry || Gallery`. Save this file with the filename `jackie_gallery.html`. Add a link to the `Jackie's Jewelry` home page to this new page. Submit your work as instructed by your teacher.