Exercise Unit01 – Laying out a Page

No exercises for this Unit. So take a breath and look at this list:

http://blog.teamtreehouse.com/23-essential-html-5-resources

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 2 – Tools and Resources

a. Logo Hijack – The Logo Bandit

In this exercise, with your newfound DOM knowledge, you are going to hijack google.com and change it into a Yahoo search site instead.

1. Open up www.google.com in Chrome, and open up the elements.
2. Find the Google logo using the Chrome Development tools.
3. Modify the source of the logo IMG so that it's a *Yahoo* logo instead
4. Find the Google Search button using the same tools.
5. Modify the test of the button so that it says "Yahoo!" instead.

Note: Google Change their markup from time to time, so you may need to be creative!

b. Drawing out a DOM Tree

In this exercise, you are going to draw out a DOM tree from web page provided for you

1. Open **`Exercises/Unit02-ToolsAndResources/dom-exercise.html`** using Chrome.
2. Using the Chome Development Tools > Elements panel, identify the elements that make up the DOM Tree.
3. Write these element using paper and a pencil or in your text editor.

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 3 – Structural Tags, Sections & Articles

a. Replacing the meaningless divs

1. Open **`Exercises/Unit03-Sections/html5-layout-styled.html.`**
2. Replace all these divs with the new HTML5 structural tags you've just learned (**`nav`**, **`header`**, **`hgroup`**, etc.)
3. If you do it correctly it should look identical to what you started with.

b. From Outline to Web

In this exercise, you will create an HTML5 page based on a **`.txt`** file that includes the DOM Tree Outline.

1. Open **`Exercises/Unit03-Sections/dom-tree-outline.txt`**
2. Based on the **`.txt`** given to you, make an html5 document with all the tags shown to you.

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 4 – Audio and Video

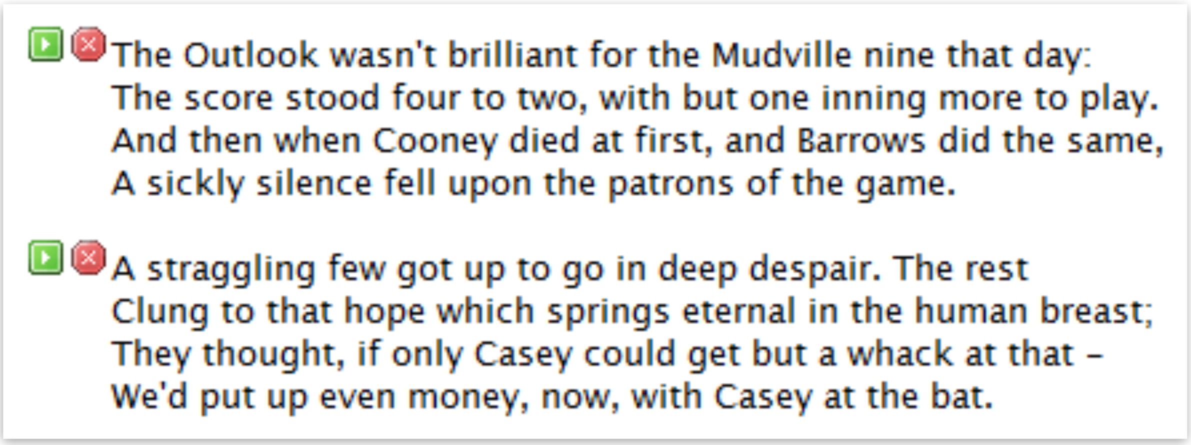
In this exercise, you will create an HTML5 file from scratch that plays video files.

a. Video Multiple Sources

1. Create a new HTML5 file called **`video-multiple-sources.html`** in the **`Exercises/Unit04-AudioAndVideo`** directory.
2. Write the code to include the justin.mp4 (mime type is video/mp4) and justin.ogv (mime type is video/ogg) files as source options. Both files are located in the media folder.
3. Play around with video attributes such as: controls, autoplay, and loop.

b. Audio Javascript

1. Open the **`audio-javascript.html`** file that's in **`Exercises/Unit04-AudioAndVideo`** directory and the **`audio-javascript.js`** file in the **`scripts`** folder. Notice that each **`<p>`** tag now has an **`id`** of the form **`pos-`** and a number. This number represents the time (in seconds) at which this stanza begins.
2. You will add JavaScript code to insert working play and pause images (found in the **`images`** folder) at the beginning of each stanza, like this:



The Outlook wasn't brilliant for the Mudville nine that day:
The score stood four to two, with but one inning more to play.
And then when Cooney died at first, and Barrows did the same,
A sickly silence fell upon the patrons of the game.

A straggling few got up to go in deep despair. The rest
Clung to that hope which springs eternal in the human breast;
They thought, if only Casey could get but a whack at that –
We'd put up even money, now, with Casey at the bat.

Copyright ©2016 DEV6 All Rights Reserved.

When the play image is clicked, the audio should jump to that stanza and play, when the pause image is clicked, the audio should pause

3. Challenge:
    a. Add code to the **`reportTime()`** function so that the stanza currently being played is given the “highlight” class (already defined in **`style.css`**
    b. Make sure to remove the class when the stanza is no longer being played.

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 5 – Forms

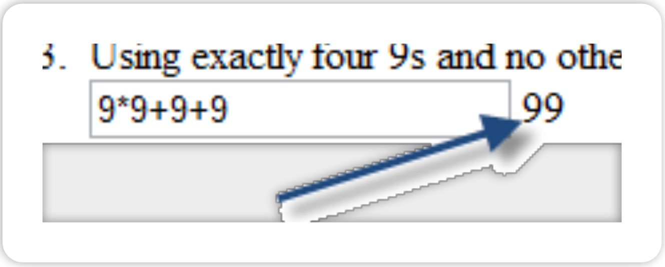
In this exercise, you will create an HTML5 quiz that validates form entries and reports the percentage of both the valid (but not necessarily correct) answers and the percentage of correct answers.

1. Open `Exercises/Unit05-Forms/quiz.html` in your editor.
2. Make the following changes to the form:
   a. Add placeholders to all questions.
   b. Make all questions **`required`**.
   c. Question 1 should only accept valid colors.
   d. Question 2 should only accept integers greater than or equal to 20.
   e. Question 3 should only accept the pattern shown in the footnote[1] (don't look if you want to figure out the pattern yourself).
   f. Question 4 should only accept valid dates.
   g. Question 5 should only accept valid URLs and should provide a list of common search engines to choose from, but should not limit the answer to those shown in the list.
3. At the bottom of the form:
   a. Add a bar showing the percentage of valid (but not necessarily correct) answers answered. Give it an **`id`** of **`"quiz-progress"`**.
   b. Add a bar showing the percentage of correct answers answered. Give it an **`id`** of **`"quiz-success".`**

[1] [9\+\-*/]{4,7}

Copyright ©2016 DEV6 All Rights Reserved.

4. Finish the `updateMeasures()` function so that it correctly updates the two bars added above on every form change.
   Hint: one way to do this is to loop through the input fields stored in the questions variables.
5. Challenge:
   a. Add code so that the result of the formula the user enters in question 3 is displayed next to the input field like this:



3. Using exactly four 9s and no othe

9*9+9+9

99

   b. Look into Bootstrap form components (http://www.w3schools.com/bootstrap/bootstrap_forms.asp) and give your form some bootstrap styling

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 6 – JavaScript

b. Hello World

To get started with writing JavaScript, open **`Exercises/Unit06-JavaScript/hello-world.html`**.

1. Add a script tag that calls to **`hello-world.js`** in **`hello-world.html`**
2. Add code into the function in **`hello-world.js`** add an call to alert a sentence when you open the document
3. Call the function so that when you open the page a window alerts you with "Hello World"
4. Add a line in the function so that "Hello World" is written in your console

c. Creating an object

Open **`Exercises/Unit06-JavaScript/film.html`**.

1. Add a script tag that calls to **`film.js`** in your html.
2. In **`film.js`** define an object called **`film`**
3. This object contains several properties, the first being **`director`** and give it the value of **`Alfred Hitchcock`** as a string
4. Assign another property named **`data`** with an empty array. This array has three objects with two properties, **`title`** and **`year.`** Give the first object the title of *North By Northwest*, as a string and the year of 1959 as a number.
5. Repeat the object with the title of *To Catch a Thief* and 1955 as the year. Add a third movie, if you'd like.

Copyright ©2016 DEV6 All Rights Reserved.

6. Add one more property to your object **`film`** and call it **`consoleFunction`**. This property is a function. In this function, state an object called **`theFilmObject`**. It should equal to **`this.`**
7. You're almost done! **`this`** calls to the film object itself, which means that you can call any property that you have defined and it will give it to you.
8. In the console, display a title like this for each movie object

   `title + " is a film directed by " + director`

Hint: **`director`** is not part of the data object …

9. Don't see anything in the console? In the last statement: call upon the **`film`** object and the **`consoleFunction()`** so that you can see what it is that you have written.

Challenge:

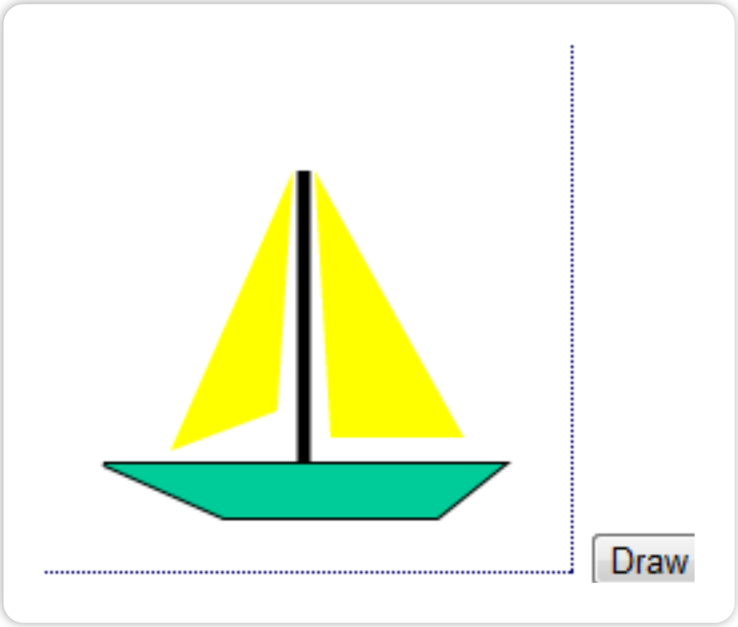
1. Create a empty **`div`** tag with the **`id="filmList"`** and have the list of films show on your web page using **`getElementById()`** and **`innerHTML`**

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 7 – Canvas

1. Sailboat

   Use HTML5 canvas to draw a simple sailboat like the one shown below:



   a. Open **`Exercises/Unit07-Canvas/sailboat.html`** in your editor.

   b. Add the JavaScript code necessary to draw the sailboat pictured above.

Challenge:

Have the left sale blink between different colors. You'll need to use some JavaScript skills to make this happen.

If JavaScript isn't your thing, try adding a little person on your sailboat.

Copyright ©2016 DEV6 All Rights Reserved.

2.Snowman

Use circles and squares to create a snowman:

a. Open **`Exercises/Unit07-Canvas/snowman.html`** in your editor.

b. Add the JavaScript code necessary to draw a snowman. Try adding:

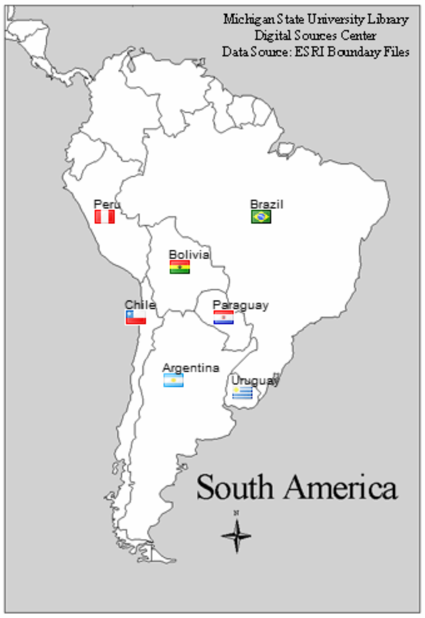
- A layer of snow on the ground.
- Three balls for the body and head.
- Eyes, mouth and nose.
- A hat.
- Arms.
- Buttons.
- A sun.

3. South America

a. Use two images found in the **`Exercises/Unit07-Canvas/images`** folder:

- **`south-america.gif`** - a map of South America.
- **`flags.png`** - a picture containing small graphics of country flags.

b. Open **`Exercises/Unit07-Canvas/south-america.html`** in your editor. You will be recreating the following drawing.



c. Add the JavaScript code necessary to:

- Create the image objects and set

Copyright ©2016 DEV6 All Rights Reserved.

their source values.

- Draw the backdrop (the map).
- Place the flags using the sprite method shown earlier. Each flag is 18 pixels wide and 13 pixels high. The source and destination positions are shown in the table below.
- Add the country names.

Country	**Source X**	**Source Y**	**Destination X**	**Destination Y**
Chile	283	88	100	250
Argentina	255	4	130	300
Brazil	171	60	200	170
Paraguay	59	452	170	250
Uruguay	59	564	185	310
Bolivia	59	200	135	210
Peru	31	424	75	170

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 8 – APIs

No exercises for this Unit. Here's some suggested reading for the Offline API and Drag and drop!

http://html5doctor.com/go-offline-with-application-cache/

https://developer.mozilla.org/en-US/docs/Web/API/HTML_Drag_and_Drop_API

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 9 – CSS3

a. The logo bandit is back! And he's making Google into a Pizza haven.

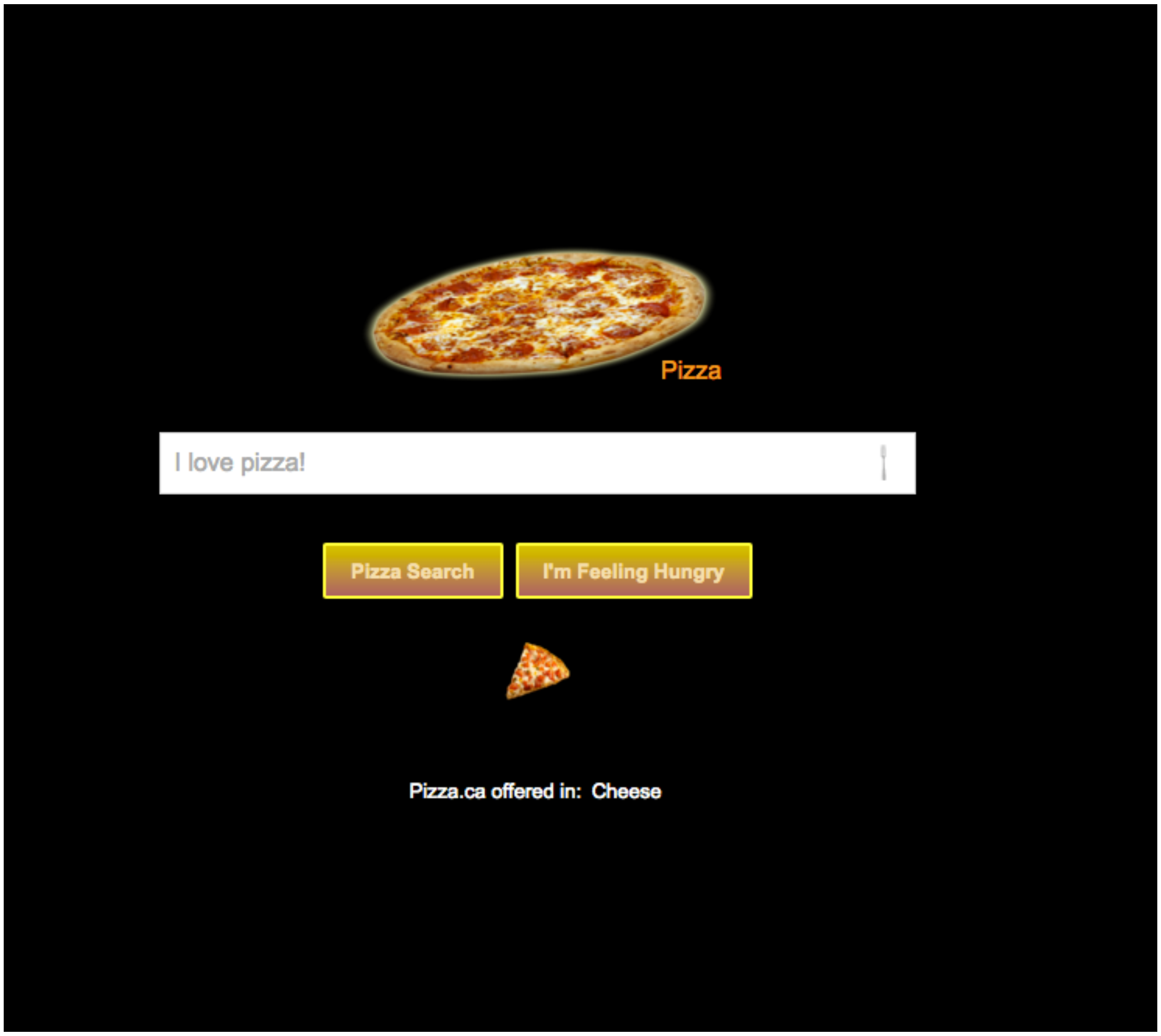
1. Open up www.google.com in Chrome, and open up the elements panel.
2. Using the HTML and CSS panels, you will be changing everything into pizza themed.
3. Change the entire background to black.
4. Find the Google logo using the elements cursor and change it to a picture of a pizza.
5. Next to the logo is a google's country at that moment. Find that word and change it to "Pizza". Change the color of the word to a good cheesy orange.
6. Give the input space a placeholder saying "I love Pizza…"
7. Next we'll be styling the buttons. First change the wording to say "Pizza Search" and "I'm Feeling Hungry". Make sure to change both the **`text`** and the **`aria-label`**. Aria Labels are an attribute designed to help assistive technology or screen readers.
8. Give the buttons a border of solid yellow. Make sure to use the Hex of yellow and not the word. Change the background color and font colorof the button too to whatever gradients you choose.

   Hint: the color for the buttons may not be under background-color.
9. Change the font at the bottom of the page to "Pizza.ca offered in Cheese". Make sure you don't accidently click any buttons, your changes won't be remembered. Change the writing to white

Copyright ©2016 DEV6 All Rights Reserved.

10. The final part of this exercise is changing the “Search by Voice” icon at the end of the input into a fork. Find the image using the element’s cursor and change the image url into a fork.

In the end, your google page should look something like this:



b. CSS3 Exercise

1. Open **`Exercises/Unit09-CSS3/html5-css3.html`**
2. Using **`style-css3.css`** set the container gradient with the width and height of **`400px`**. Make sure that you include all browser prefixes.

Copyright ©2016 DEV6 All Rights Reserved.

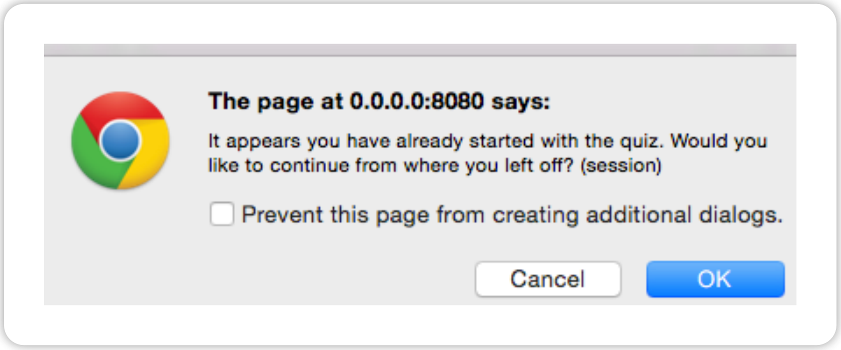
3. Make it so that the gradient direction is reversed when in portrait mode, and back in landscape.
4. Using the 2D transforms change the perspective of the container object.
5. Create a CSS feature so that when you hover the container, your object transforms with a 3D feature.
6. Add a transition so that the 3D feature ease's in.
7. Create three `<h3>` tags and choose a different font for each.
8. Without giving an id or class to the header tags, reference them using new selectors that you've just learned.
9. Give them each a different font, one from the font-folder provided to you and the two others from the Google font of your choice.
10. Add some margin and padding to center the container.
11. Make sure to use the Mozilla Developer Network Tools to help you design your exercise.

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 10 – Storage

In this exercise, you will create a quiz application that allows the user to save and resume later. It also protects the user from losing data if he/she accidentally refreshes.

1. Open Solutions > 6.b. Saving Quiz Challenge in your browser and play with the application:
   a. Answer one or more questions and then refresh the browser. You will get a dialog giving you the chance to use refill the form with values stored in **`sessionStorage`**:



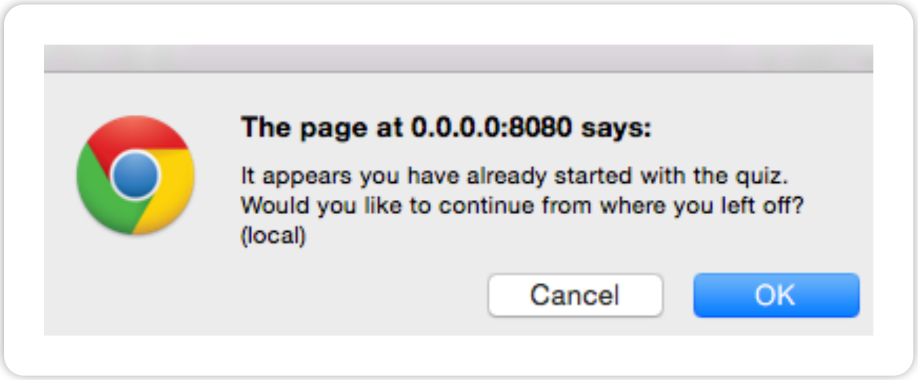
The page at 0.0.0.0:8080 says:

It appears you have already started with the quiz. Would you like to continue from where you left off? (session)

   b. Press OK and your values get added back.
   c. Refresh again and press Cancel on the dialog. Your values do not get added back.
   d. Refresh again. You don't get the JavaScript dialog because the **`sessionStorage`** key/value pairs were removed when you pressed Cancel in the previous step.
   e. Answer one or more questions again and click the Save My Answers for Later button.

Copyright ©2016 DEV6 All Rights Reserved.

f. Close and reopen the browser. You will get a dialog giving you the chance to use refill the form with values stored in **`localStorage`**:



The page at 0.0.0.0:8080 says:

It appears you have already started with the quiz.
Would you like to continue from where you left off?
(local)

g. If you click Cancel, the **`localStorage`** key/value pairs will be removed and you'll have to start the quiz over.

h. If you click OK, the form will be refilled with your previous answers and they will be saved into **`sessionStorage`**.

i. Also notice that the footer contains the time and date the quiz answers were last saved

2. Now open **`Exercises/Unit06-Storage/saving-quiz.html`** in your editor.
3. In the **`addLoadEvents()`** function:
   a. Loop through the inputs and add event listeners that capture change events to save the associated key/value pair in **`sessionStorage`**. Don't forget to use the prefix.
   b. Add an event listener to the Save button to capture a click event and call **`saveAnswers`**.
   c. Call the **`refill()`** function at the end.
4. Write the code in the **`saveAnswers()`** function to save all the answers in **`localStorage`**.

Copyright ©2016 DEV6 All Rights Reserved.

5. The **`refill()`** function:
    a. calls **`hasAnswers()`**, which returns "session", "local", or false, depending on if and where it finds saved answers. If there are no saved answers, it returns without doing anything.
    b. declares some variables:
        i. **`confirmed`** - we'll change it to true if the user wishes to refill the form.
        ii. **`msg`** - the message to ask the user if he/she wants to refill the form.
        iii. **`questions`** - the question inputs
    c. loops through the inputs. On the first iteration, it prompts the user with the message. If the user clicks Cancel, the key/value pairs are deleted (via the **`deleteAnswers()`** function) and the function returns/ends. Otherwise, we iterate through the questions. This is where you come in.
    d. Add code to populate the question inputs from the appropriate storage location (based on the value of **`fillFrom`**).

Challenge:

Copyright ©2016 DEV6 All Rights Reserved.

1. Notice that an external script called **`dateFormat.js`** is included. That extends the Date object prototype with a **`format()`** method, which you use as follows:

```
var now = new Date();
var dateMask = "yyyy-mm-dd H:MM:ss";
var formattedNow = now.format(dateMask);
```

2. Use this to write out the date last saved to the output element below the form.
3. You will need to store the date in **`sessionStorage`** and **`localStorage`** as appropriate. Don't forget the prefix.
4. Note that the **`dateMask`** variable is already set in the code.

Copyright ©2016 DEV6 All Rights Reserved.

Exercise Unit 11 – Bootstrap

You will be creating a small form implementing the Bootstrap components that you have just learned.

1. Use the `Exercises/Unit10-Storage/saving-quiz.html` that you just completed and edit it to include bootstrap styling. If you have not finished the exercise use `Solutions/Unit10-Storage/saving-quiz-challenge.html`
2. First include the link to Bootstrap in the head of the document.
3. Wrap the contents of the body in a container.
4. Look into Bootstrap Forms to model the contents of the form tag and input and implement these into your document.
5. Add styling to your "Save Answers For Later" button.
6. Add a Nav Bar on the top of the Quiz and add a name to the quiz
7. In the Nav Bar implement a Google search input that search's Google and opens the results on a separate page.
8. Let's add a bootstrap sticky footer. Wrap the "Answers last saved:.." text into a footer tag and add a bootstrap footer to your document.
9. Next thing we'll do is wrap each quiz question into a Bootsrap panel. Add a title for each question. Make each panel a different color. Let's also add a panel for the progress bar a meter bar called "Meters". Add a Bootstrap table with two rows and two columns: one with the title "Successful Answers", and one for "Your Progress". Add the bars in their respected column.

**In Bootstrap 4 panels no longer exist and have been replaced by "Cards".

Copyright ©2016 DEV6 All Rights Reserved.

10. Next, lets create a responsive grid layout that's more interesting. Place the first three panels in their own row. Place the last three panels in their own row. Create columns that place the three panels in three columns when in landscape view and in three separate rows in portrait or small views.
11. Add media queries so that all of your panels are the same height when in landscape view.
12. If you have some extra time, add any extra CSS or bootstrap components you want (maybe a well for an introduction to the Quiz or some icons..). For full list visit: http://getbootstrap.com/components/

Copyright ©2016 DEV6 All Rights Reserved.