Web Apps Assignment 6

The purpose of this assignment is to explore basic java script and interact with the DOM. Make sure that your commit messages are good. You can either make your commits and pushes from git bash or from GitHub desktop.

Part 1: Create remote and local repos

1) Click on the https://classroom.github.com/a/b9NC0g7h classroom where you will have your own repo for the assignment.
2) You should have a bare bones README.md and sample code for input and checked.
3) Clone down the repo to a local repo.
4) Open the repo in VS Code.
5) **Screen shot 1 of repo on VS Code for submission**

Part 2: Create basic HTML structure

1) Use VS Code to make a copy of sample-input.html with the name author.html.
2) Pick a descriptive title and for the header use " Author entry" followed by your name.
3) Save your work.
4) Open the file with a browser (like Chrome) and verify that you have the title on the tab and your header is displayed.
5) Inspect the code
6) **Screen shot 2 on your browser of rendered page and the inspector**.

Part 3: Create the input form and build the response code. This part is pretty close to the sample-input.html. You can modify it, or for more of a challenge, build up the elements on your own and refer back if you need help. Note that any error messages will show up in the console.

1) First, start with the form.
2) You will need two input elements each with an id
    a. The first element will hold a secret string value. Make it of type password. Give it an initial value of "Dickens". Label it with "The hidden author name"
    b. The second element will hold the input to check. Make it of type text. Label it with "Guess the author".
3) Add the button. You can use either a regular button or an input of type button. It should have the text "Check the guess". It will have an onclick that is checkAuthor).
4) We need an element with an id to put our results. It can be a paragraph or a div.
5) We need some java script in a script element.
6) Create two global variables ***good*** and ***oops*** both initialized to 0.
7) Create a function checkAuthor().
    a. It will get the value from the hidden author input element.
    b. It will get the value from the guess input element.

    c. Compare the values and if they are the same add one to good and do an alert with message “You got it right”, otherwise add one to oops and alert with message “Sorry”
    d. Replace the innerHTML for the tagged element from step 4 with the good/oops counts.
8) Verify that the code works. You should be able to change both inputs and pressing the button should result in an alert and an updated score displayed.
9) Commit and Push. You can either use git bash or github desktop to do your commit and push operations.
10) **Screen shot 3 on VSCode with form and right/wrong holder visible.**

Part 4: Change an image

1) Get the three images for your cities from the last assignment and add them to your directory.
2) Add a div container and add one of the images to the container. Make the size of the image 100 by 100.
3) Add an onclick to the image with value selectImage().
4) Add a script. Make a global variable useImage with value 1
5) Create a function selectImage().
    a. Do a switch on the useImage with three cases (1, 2, 3)
    b. For each case set the image src and then update the useImage value to the next one. Wrap back around to the first image from the last one.
    c. Don’t forget the break
6) Test it out. Each click should cause the next image to display.
7) Commit and Push.
8) **Screen shot 4 on browser** (should see both the guess and image.)
9) **Screen shot 5 on VSCode** showing the java script.

Part 5: Get values and do a looping computation.

1) Use VS Code to make a copy of sample-input.html with the name tips.html.
2) Pick a descriptive title and for the header use your name followed by “ shows tip amounts”.
3) You will need two input elements each with an id
    a. The base element holds the amount we are going to compute the tip for. Make it of type number. Give it an initial value of 50. Label it with “Enter base amount”
    b. The conversion element will hold the second value. Give it an initial value of 15. Make it of type number. Label it with “Enter the tip percent.
4) Add the button. You can use either a regular button or an input of type button. It should have the text “Show tip ”. It will have an onclick that is tips().
5) We need a tagged element to put our results. It can be a paragraph or a div.
6) We need some java script in a script element.
7) Create a function tips().

a. It will get the base value from input amount.
b. It will get the tip from input tip.
c. It will initialize a local variable ***myOutput*** to an empty string.
d. Create a loop that will iterate 5 times. It will compute a tip value on the base value for five tip levels: tip, tip +4, tip + 8, tip + 12, tip + 16
e. Append each of the values and the calculated tip value to toShow using a format of "***tip%*** is ***calculated tip***; "
f. Replace the innerHTML for the tagged element from step 5 with "Tips are:" concatenated with toShow

8) Check your results. An example:
    a. **Tip are: 12% is 2.4; 16% is 3.2; 20% is 4; 24% is 4.8; 28% is 5.6;** (base amount was 20, tip is 12)
    b. Commit and Push.

Part 6: Create a radio button group that switches between calculations.

1) Create a form holding four labeled radio input elements. You can use sample-checked.html as a reference. Each of the inputs will have onClick set to "chooseOp()"
    a. The first radio button will be labeled with "sum". Give it the value "sum"
    b. The second radio button will be labeled with "diff". Give it the value "difference"
    c. The third radio button will be labeled with "mult". Give it the value "product"
2) We need a tagged element to put our results. It can be a paragraph or a div.
3) We need some java script in a script element.
4) Create a function chooseOp().
    a. It will get the value from input base from the previous part
    b. It will get the value from input tip from the previous part
    c. It will set a value based on the checked property of the radio buttons and will use base and tip as the operands to the corresponding operation. (I.E. if sum is checked, then the value will be the sum of base and tip).
    d. Replace the innerHTML for the tagged element from step 3 with appropriate text (E.g. product of 5 and 10 is 50)
5) **Screen shot 6 on browser** (should see both parts.)
6) **Screen shot 7 on VSCode** showing the java script.
7) Commit and Push.

Part 7: Use check boxes

1) Use VS Code to make a copy of sample-input.html with the name cruise.html.
2) Pick a descriptive title and for the header use your name followed by "goes on a cruise".
3) At the top of the body after the header add in a form group holding three labeled checkbox input elements. Each needs an id. (Note: you can keep the basic form, we will modify it in the next part.)

4) You will need three input elements each with an id each of checkbox.
    a. The first checkbox will be labeled with “Shares Cabin”. Give it the value “True”
    b. The second checkbox will be labeled with “Breakfast”. Give it the value “True”
    c. The third checkbox will be labeled with “Pool Pass”. Give it the value “False”
5) Add a button. You can use either a regular button or an input of type button. It should have the text “Select a cruise ”. It will have an onclick that is selectCruise().
6) We need a tagged element to put our results. It can be a paragraph or a div.
7) We need some java script in a script element.
8) Declare a global variable my_cruise and set it to an empty object.
9) Create a function selectCruise().
    a. It will reset the my_cruise to an empty object.
    b. It will set the properties shared, breakfast, and pool for my_cruise with true or false depending on whether each of the checkboxs is checked or not.
    c. Replace the innerHTML for the tagged element from step 6 with my_cruise.
10) Commit and Push.

Part 8: Work with the object from the previous part

1) Below the checkbox form, we will need a form (you can modify the form from the starter code) the with two text elements and a button.
2) You will need two input elements each with an id
    a. The first element holds the name of a property. Make it of type text. Give it an initial value of “days”. Label it with “Enter the number of days”
    b. The second element will hold the price of the cruise. Make it of type text. Label it with “Enter price”.
3) Add the button. You can use either a regular button or an input of type button. It should have the text “Update the cruise”. It will have an onclick that is updateCruise().
4) Add in a function updateCruise() to the script from the previous part.
    a. It will get the value from days input element.
    b. It will get the value from the price input element.
    c. It will set the properties of my_cruise with the value we just got from the input elements. (Use the form my_cruise[…] = …)
    d. Replace the innerHTML for the tagged element with my_cruise.
5) Commit and Push.
6) **Screen shot 8 on browser**.
7) **Screen shot 9 on VSCode** showing the java script.

Part 9: Host on GitHub pages

1) Go to setting on GitHub and select the main branch.
2) Copy the URL.

3) Edit README and add three lines “Hosted at “ followed by a link to the URL from the previous step with the files author.html, tips.html, and cruise.html. (Commit and Push if you did it on the local repo.)
4) Verify that if you click on the links in the README, that you get your pages.

Bonus: On the first part, use a div to display incorrect guesses. After 7 incorrect guesses, replace the onclick property value with a new function call that displays an alert with the text “Sorry, ran out of guesses.” You may also pull in the bootstrap framework and use it to style the elements in these apps. Use your sense of style to improve the look of these apps. Do small chunks and commit each time your modification succeeds. I would recommend that you tag the commit for your working app. (Read about creating a Git tag for a commit here https://git-scm.com/book/en/v2/Git-Basics-Tagging)

Provide screen shots on the browser showing each of the modified apps.