Task checklist for JavaScript functions

1. Validate the form on your contact page. Ensure that all text has content, and that the content conforms to requirements (you decide and document what those requirements are)

This JavaScript has been inserted in the contact.html page at the bottom of the body. I used 3 regexes for validation, one of them was found here: https://emailregex.com/.

2. When the form has been validated, show a confirm dialog box that displays the data entered and asks the user if that data entered is correct. If yes, thank the visitor and submit the form. If no, prevent the form from submitting so users can go back and make adjustments where necessary. Ensure the form fields remain populated with data.

This code is in the same place as above, I used the confirm() method.

4. Add functionality that allows the user to enter the name of an item into a search field. As they type, a list of possible matches should appear. The user can select an item on the list to place it into the search textbox.

This code is on the autocomplete.js script (scripts/autocomplete.js) which I found here: https://www.w3schools.com/howto/howto_js_autocomplete.asp - the autocomplete.css file was found in the same link. The search bar itself is on the rentals.html page (also where the CSS is called from). Some code (the array "properties" and function which calls the rest of the code) is executed from the top of the rentals.js script (scripts/rentals.js)

6. As an alternative to above, dynamically build a table that contains a list of products/services that a user enters into the search textbox. This would be done when the user clicks the search button. If there are no products matching the entered term, a message informing the user that no matches have been found should be displayed in the page (not in an alert box).

For this code I created a jsonCall.js script (scripts/jsonCall.js). I used vanilla JS to create an xmlHttpRequest to a JSON file stored on myjson.com. There is also a file in the root folder called resource.json which can be used if the myjson server is down. Again the method is called from the rentals.html page (where the search bar is).

7.Create an image gallery that allows the user to move through a selection of recommended products/services. The user should be able to move backwards and forward through the selection.

This code is in carousel.js (scripts/carousel.js). It's a combination of my own code and some found here: https://www.w3schools.com/howto/howto_js_slideshow.asp. The CSS file carousel.css was also found from the same site. The CSS and JavaScript is called from the gallery.html file.

The website is also hosted on my GitHub account here: https://gamblepants.github.io/