| Sprint 8, Assignment 13.3 Please also update the doc name with correct numbers. | |
| --- | --- |
| Assignment type: JS Interactive | |
| Assignment name  Can remain the same as the assignment topic name, or…  can be created depending on the task in the active form, e.g. “Create your own X” | “Using Array Methods” |
| BDG Description\*  What is the task and why is it important?  In this part, it’s encouraged to think about storytelling and future job-specific context e.g. “You’ve been asked to help out X with Y. They want Z on their website, yet aren’t too sure on how to achieve it”.  Drawing on practical examples and adding context can increase a student's motivation and increase long term learning according to Instructional Design principles, because this helps to relate some familiar or existing knowledge to new bits of information.  This will appear in the course as text before a button, leading to the interactive platform assignment. | Your author client would like to add some features to your program:   * She can’t remember where she began to talk about a specific topic, so try using the .indexOf method to find where our poem first mentions “piggy” * She also can’t remember the last place she mentioned a place, so try to find the last instance of a word that appears multiple times in our poem using .lastIndexOf * She would like to find a quotation in her work, so use the .slice method to grab what the last piggy said |
| The Assignment  A short specific description of the assignment and tasks using bullet points that the student will need to do. | Use the .split, .indexOf, .lastIndexOf, .slice, .splice, .join, and .concat methods |

| Steps  Step-by-step instructions on what the student should do. | | |  |
| --- | --- | --- | --- |
| Step # | Step  Write each small step of the task | At least 1x hint(s)  Write some text (not necessarily, but can also be a part of code if relevant) which would hint the student to figure out the correct step forward. | The correct output should be…  (if relevant to the task) |
|  | Convert the original, single-lined poem from before to an array | Use the .split method:  let originalPoem = `This little piggy went to market this little piggy stayed home this little piggy had roast beef and this little piggy had none and this little piggy cried "Wee! Wee! Wee!" all the way home!`;  let arrayedPoem = originalPoem.split(" "); | arrayedPoem = ['This', 'little', 'piggy', 'went', 'to', 'market', 'this', 'little', 'piggy', 'stayed', 'home', 'this', 'little', 'piggy', 'had', 'roast', 'beef', 'and', 'this', 'little', 'piggy', 'had', 'none', 'and', 'this', 'little', 'piggy', 'cried', '"Wee!', 'Wee!', 'Wee!"', 'all', 'the', 'way', 'home!'] |
| 1 | We need to find where a specific line starts (using the .indexOf method) | We need to know at what index number the word “piggy” starts | console.log(arrayedPoem.indexOf("piggy")); |
| 2 | We need to find something later in the manuscript (using the .lastIndexOf method) | We need to know at what index number we can find the word “cried”  Remember that this method will begin at the end of the array and work *backwards* | console.log(arrayedPoem.lastIndexOf("cried")); |
| 3 | We need to grab a quote and store it in a new variable (using the .slice) method | We need to grab an entire quote, so use our last two methods to find the beginning and end of the quote. | console.log(arrayedPoem.indexOf(`cried`));  console.log(arrayedPoem.lastIndexOf(`all`));  console.log(arrayedPoem.slice(28,31)); |
| 4 | We need to delete a phrase (using the .splice) method | Use the .indexOf method to find out where the first “and” is, and then delete it | console.log(arrayedPoem.indexOf(`and`));    console.log(arrayedPoem.splice(17,1)); |
| 5 | We need to combine several pieces of information into a sentence/string (using the .join method) | Re-combine the array’s elements into a single string, with each word separated by a space | console.log(arrayedPoem.join(" ")); |
| 6 | We need to add that sentence at the end of the manuscript (using the .concat method) | Add an ending to the poem, like “The End” but stored in the array as separate elements | console.log(arrayedPoem.concat("The","End")); |
| ... |  |  |  |