|  |  |
| --- | --- |
| Sprint 9, Assignment 4.9 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 Scope Variables” |
| 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. | With your code becoming ever-more impressive, now is the time to start concealing the important parts for purely internal use. Using scope variables, convert the three functions you’ve just created into function and arrow expressions. |
| The Assignment  A short specific description of the assignment and tasks using bullet points that the student will need to do. | * Declare one or more function expressions * Declare an arrow expression |

|  |  |  |  |
| --- | --- | --- | --- |
| 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) |
| 1 | Convert your addressFunction function into a function expression |  | const addressFunction = function (input) {  *// console.log(input);*              return {              type : input[0],              line1 : input[1],              line2 : input[2],              city : input[3],              state : input[4],              zip : input[5]}          };          this.address = addressFunction(addressInfo); |
| 2 | Convert your parentCombine function into a function expression |  | const parentCombine = function () {              let result = ``;              for (const element of parents) {  *// console.log(element);*                  result = result.concat(element + ` & `);              }              return result.slice(0, -3)          };          this.parents = parentCombine(parents); |
| 3 | Convert your greeting function into an arrow expression |  | const greeting = () => `Ho Ho HOooo!`;          this.salutation = greeting(); |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| ... |  |  |  |