| Sprint 8, Assignment 7.12 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” | “Write primitive Data types” |
| 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. | An autoparts supply company needs you to update their website’s ‘Welcome’ page to improve business, as it currently just has the company logo (and no text). Since you now know how to do things like writing strings, using methods, and performing math, let’s see if you can improve this page!  Of note, they have asked that the welcome page celebrate how long each visitor they’ve had an account for, or note the day they first signed-up (so try to address this mid-way through the assignment). |
| The Assignment  A short specific description of the assignment and tasks using bullet points that the student will need to do. | * Write three strings: A basic string, a string that contains a literary quotation from an author/quote from a public speaker (perhaps using interpolation), and a single string that spans multiple lines of code * Use three methods * Perform a basic math calculation, and then verify the data type of your result (that it’s a number, and not a string) * Perform a boolean check * Use JavaScript’s coercion to combine mixed data types (strings and numbers) |

| 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 | (Using your app.js file) Write and display a string with a text greeting | Console.log using double-quotes (“) | console.log(“Hello valued customer!”) |
| 2 | Write and display a string that uses interpolation | Change your double-quotes to backticks (`), and add double-quotation marks around quoted text | console.log('Hi Neighbor! You know what we always say: "We have better prices than our competition - \*every\* time!"'); |
| 3 | Write and display a string that spans multiple lines of code | Separate your string into individual sentences - each one on a different line - for readability | console.log(`Hi Neighbor!  You know what we always say: "We have better prices than our competition - \*every\* time!"  And make sure to check our “latest discounts” page!`); |
|  |  |  |  |
| 4 | Try punching up our first line (“Hi Neighbor! …) by making the entire line capitalized | Use the .toUpperCase method  console.log(`Hi Neighbor!`.toUpperCase()) | “HI NEIGHBOR!” |
| 5 | Check to see how many elements/characters are in the second line of your greeting | If your second line of the string is “You know what we always say: "We have better prices than our competition - \*every\* time!"” … how many characters is that? Try using the .length method  console.log(`You know what we always say: "We have better prices than our competition - \*every\* time!"`.length) | It should be about 90 characters |
| 6 | Split the second line into two, at the “:” character | Use the .split method  console.log(`You know what we always say: "We have better prices than our competition - \*every\* time!"`.split(':')) | You know what we always say:  “We have better prices than our competition - \*every\* time!" |
|  |  |  |  |
| 7 | Perform a basic math calculation  Note: this is a great opportunity to meet the client’s celebratory needs | If you need to find out how long someone has been a customer, you could subtract the number of days since they first created their account from the current date:   * Assume it’s the 27th of September, and subtract 27 from 30 (using console.log) in order to find out when this hypothetical customer signed up   console.log(30-27); | This particular account was created on the 3rd |
| 8 | Check to see that your result is a number and not a string, using the typeof operator | Make sure to wrap your math in a set of parentheses  console.log(typeof (31-27)); | number |
|  | Perform a boolean check  Note: this may come in handy later, if the client wants to perform checks (such as whether the customer has done business for more than a year, lives in Europe, or is a “value plus” member) | Is 27 greater than 30?  console.log(27 > 30); | false |
|  | Use strict equality to confirm that your math is returning a number (and not a string) | Is the result of 30-27 the same as “3”?  console.log((30-27) == "3");  Is the result of 30-27 strictly equal to 3?  console.log((30-27) === 3); | True  False |
|  | Test Javascript’s coercion ability by adding our result to a string in order to get   * A string, and then * A number | console.log((30-27) + ` is the first day you joined!`);  console.log(++"3"); | `3 is the first day you joined!`  4 |
|  |  |  |  |