| Sprint 10, Assignment 1.7 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 phrased in the active/imperative voice (depending on the task), e.g. “Create your own X” | “Selecting Elements in the DOM” |
| Description  The essential tasks a student must perform in order to complete the assignment  ?+ a general, colorful introduction to the assignment? | * Using Element selectors, select …   + All instances of a tag   + An element with a unique ID   + All instances of a class * Using CSS selectors, select …   + A unique ID   + The first instance of a class   + All instances of a class   + A nested instance of that class |
| Link to full code file (Answer)  Push a full code file to our [GitHub repo](https://github.com/bitdegree-foundation/academy-assignments-code-files) for this sprint’s assignment and paste a link to that file here. Please write the file in a concise and clear format, according to the “[example-js-code-file](https://github.com/bitdegree-foundation/academy-assignments-code-files/blob/master/example-js-code-file.js)” found in our repo. | (Upload) |

| # | Step  Write each small step of the task. These assignments should be doable by students on their own (without BitDegree-instructor help). It can be any number of steps, but keep them pretty clear and separate (don’t combine 2 steps into 1 step). | 1x hint  Write brief text or partial code that will help the student figure out how to either move forward or completely accomplish the given step. | Output Expected code that our platform will be able to run & check against student submissions |
| --- | --- | --- | --- |
| 1 | Create an .html document, and paste the following text into it:    <!DOCTYPE *html*>  <html *lang*="en">  <head>  <meta *charset*="UTF-8" />  <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />  <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />  <title>"Selecting DOM Elements" (Sprint 10, Assignment 1.7)</title>  </head>  <body>  <script *src*="./1\_7\_script.js"></script>  </body>  </html> |  |  |
| 2 | Add two div elements with IDs - such as "container\_1" and "container\_2" |  | <body>  <div *id*="container\_1"></div>  *<!-- STEP 2 -->*  <div *id*="container\_2"></div>  *<!-- STEP 2 -->*  <script *src*="./1\_7\_script.js"></script>  </body> |
| 3 | Within "container\_1's" div, nest a div with the class of "text"; do the same for "container\_2" |  | <div *id*="container\_1">  <div *class*="text"></div>  *<!-- STEP 3 -->*  </div>  <div *id*="container\_2">  <div *class*="text"></div>  *<!-- STEP 3 -->*  </div> |
| 4 | Give each of these text-class divs something to display on the web page - such as "This is the FIRST section" and "This is the SECOND section" |  | <div *id*="container\_1">  <div *class*="text">  This is the FIRST section  *<!-- STEP 4 -->*  </div>  </div>  <div *id*="container\_2">  <div *class*="text">  This is the SECOND section  *<!-- STEP 4 -->*  </div> |
| 5 | Create a .js file named " 1\_7\_script.js " |  | "use strict"; |
| 6 | Store and log all instances of a given **tag** using Element selectors |  | const elementsByTag = document.getElementsByTagName("div");  console.log(elementsByTag); |
| 7 | Store and log the sole instance of a given **ID** using Element selectors |  | const elementByID = document.getElementById("container\_1");  console.log(elementByID); |
| 8 | Store and log all instances of a **class** using Element selectors |  | const elementsByClass = document.getElementsByClassName("text");  console.log(elementsByClass); |
| 9 | Store and log the sole instance of another **ID** using CSS selectors |  | let queryElementID = document.querySelector("#container\_2");  console.log(`This is the ONLY element that has the \*ID\* of "container\_2":`);  console.log(queryElementID); |
| 10 | Store and log the **first** instance of a **class** using CSS selectors |  | let queryElementsFirstClass = document.querySelector(".text");  console.log(`This is the FIRST element that has the \*class\* of "text":`);  console.log(queryElementsFirstClass); |
| 11 | Store and log **all** instances of a **class** using CSS selectors |  | let queryElementsAllClass = document.querySelectorAll(".text");  console.log(`This is EVERY element that has the \*class\* of "text":`);  console.log(queryElementsAllClass); |
| 12 | Store and log a **nested** instance of a **class** using CSS selectors |  | let queryElementNestedClass = document.querySelector("#container\_2 > .text");  console.log(`This is the \*first\* element INSIDE the #ID of "container\_2" element ... that has the \*class\* of "text":`);  console.log(queryElementNestedClass); |
| ... |  |  |  |