| Sprint 10, Assignment 2.6 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” | “DOM Methods in Practice” |
| Description  The essential tasks a student must perform in order to complete the assignment  ?+ a general, colorful introduction to the assignment? | It's time to do some work on your personal website! You are going to need a navigation bar that can easily direct your visitors (hopefully many software companies) to things like your portfolio projects, a bio, and maybe even your own educational coding blog! You've found an HTML template, but it is far too basic and impersonal for what you need. Use your new DOM skills to update it via JavaScript:   * Delete & replace the stale "company blurb" text, * Delete the old sorted list, and * Use a loop to create a new one |
| 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 | Paste the following starter code into a new HTML document:  <!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>Sprint 10, Assignment 2.6 ("DOM Methods in Practice")</title>  </head>  <body>  <p *id*="company-blurb">  Thank you for using our business template -  <strong>please visit us</strong> for more great <code>code</code>!  </p>  <ul *id*="navBar">  <li>Home</li>  <li>Products</li>  <li>Customer Support</li>  <li>Careers</li>  <li>Investors</li>  <li>News</li>  <li>About Us</li>  </ul>  <script *src*="./2.6.Assignment.script.js"></script>  </body>  </html> |  | <!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>Sprint 10, Assignment 2.6 ("DOM Methods in Practice")</title>  </head>  <body>  <p *id*="company-blurb">  Thank you for using our business template -  <strong>please visit us</strong> for more great <code>code</code>!  </p>  <ul *id*="navBar">  <li>Home</li>  <li>Products</li>  <li>Customer Support</li>  <li>Careers</li>  <li>Investors</li>  <li>News</li>  <li>About Us</li>  </ul>  <script *src*="./2.6.Assignment.script.js"></script>  </body>  </html> |
| 2 | Create a new javascript file named "2.6.Assignment.script.js" |  |  |
| 3 | Store & remove or replace the company blurb | Hint: use querySelector to get the paragraph's unique ID  Hint: you could use .innerText to update what's displayed to something more personal, or .remove to get rid of it entirely. For now, replace the text with a reminder. | const junk = document.querySelector("#company-blurb");  console.log(junk);  junk.innerText = "\*\*\* REMEMBER to put something fun here once I get the navbar working ... \*\*\*"; |
| 4 | Grab & store the parent element of your navBar | Hint: it is an unsorted list, and has a unique ID - either of which you can target | const navMenu = document.querySelector("#navBar"); |
| 5 | Log to terminal both the navMenu and its children | Hint: use the "children" method | console.log(navMenu);  console.log(navMenu.children); |
| 6 | Grab & store the children in their own variable. Log both the array in this variable, and the array's length. | Hint: store what you have just logged | const childList = navMenu.children;  console.log(`Here are all of the children of the navMenu object (stored in 'childList'):`);  console.log(childList);  console.log(`The number of elements in your childList is ${childList.length}`); |
| 7 | Create a 'for' loop that will log and then delete each child element - each sorted list element | Hint: use the "remove" method  Hint: if you are only able to delete every other element, this may be because your 'for' loop is starting at the *beginning* of the array. However, in this admittedly-strange case, you will need to make your loop start at the \**end*\* of the array (and work backwards).  Note: this is because you are storing a "live NodeList". Try googling these and see how they automatically update (and move) their elements around in the array. Can you figure out why a loop would end up skipping every other element? What index place would it check in its second iteration, and what element is now stored in that index (thanks to live NodeList logic)? | for (let i = childList.length - 1; i >= 0; i--) {  console.log(`Here is the current element of the childList`);  console.log(childList[i]);  childList[i].remove();  } |
| 8 | Create a new array and store new names for your personal site's navBar (for instance... "Home", "About Me", "Portfolio", "Tech Blog", "YouTube Channel", "Other Social Media","Coding Meme of the Day" etc) |  | const list = [  "Home",  "About Me",  "Portfolio",  "Tech Blog / T.I.L.",  "YouTube Channel",  "Other Social Media",  "Coding Meme of the Day",  ]; |
| 9 | Create a 'for' loop that will 1. iterate through your list, 2. create a new sorted-list element every time, 3. set the innerText of that list element to the text stored in your list, 4. log the list element, and finally 5. append it as a child to your parent navBar element |  | for (let item of list) {  const newItem = document.createElement("li")  newItem.innerText = item;  console.log(newItem);  navBar.appendChild(newItem);  } |
| 10 | *Create a new sorted-list element every time* |  |  |
| 11 | *Set the innerText of that list element to the text stored in your list* |  |  |
| 12 | *Log the list element* |  |  |
| 13 | *Append it as a child to your parent navBar element* |  |  |
| ... |  |  |  |