BROWSER API EXERCISES

Using the supplied Starter Files, perform the following tasks:

1. Create a variable called container. For the value, get the div with an ID of “app” from the page. Log it to the console to test.
2. Create a variable called **firstHeader**. Get the value of the first h1 on the page. Log it to the console to test.
3. Create a variable called **firstDiv**. Get the value of the first div on the page. Log it to the console to test.
4. Log out the HTML inside of container to the console.
5. Log out the HTML inside of **firstHeader** to the console.
6. Log out the text inside of **firstHeader** to the console.
7. Log out the HTML inside of **firstDiv** to the console.
8. Log out the text inside of **firstDiv** to the console.
9. Create a variable called links that includes all of the links on the page. Log it to the console to test.
10. Create a variable called **linksArray** that converts link to an array using **Array.from(link)**.
11. Map over the **linksArray** and log out the text of each link.
12. Create a variable called **menuLinks** that selects all of the links inside of the menu and converts them to an array using **Array.from()**. Then logout all of the menu links to the console.
13. Create a function called **staticHeader()** that returns the following string of HTML <h2>Static Header</h2>. Log out **staticHeader()** in the console to test.
14. Add the static header inside of the ***<div id=”app”></div>*** right before the closing of the div. Use **insertAdjacentHTML()**.
15. Create a function called header(title) that takes a title as a parameter and then returns the following string with the title: <***h2>TITLE</h2>***. Log out header(`My Header`) to test.
16. Add the dynamic header inside of the ***<div id=”app”></div>*** right before the closing of the div. Use **insertAdjacentHTML()**.
17. Create a new function called **menuLink(text, link)** that takes two parameters: text and link. Have the function return the following markup: ***<li><a href=”LINK”>TEXT</a></li>***. Log **menuLink(`Home`, `/#home`)** in the browser to test.
18. Create an event handler function called **logTitle(e)**. Have the function prevent the default event behavior. Then have it log out the **innerText** of whatever was just interacted with. This does not need to do anything; we are just setting up the next step in exercise 19.
19. Map over the **linksArray** from before and attach **logTitle** as an event listener to each link when it is clicked on. Try clicking and checking the console to test.
20. Create an event handler function called **logLink(e)**. Have the function prevent the default behavior and then log out the “**href**” value of whatever link was just interacted with.