Session 7 Homework

Code For Everyone JavaScript

|  |  |
| --- | --- |
|  | **Study** |

1. insertAdjacentHTML first parameter is the position where you put the new HTML into the container

Read this [documentation](https://developer.mozilla.org/en-US/docs/Web/API/Element/insertAdjacentHTML), then read the following code

HTML File:

<ul id="list"></ul>

JS File:

const container = document.getElementById(‘list’);

for(let i = 0; i < 3; i++) {

container.insertAdjacentHTML(‘<position>’, `<li>${i}</li>`)

}

Answer which is which:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Position | Result |  | Position | Result |
| 1. beforebegin |  | 4. afterend | d. |
| 2. afterbegin |  |  |  |
| 3. beforeend |  |  |  |

1. Instead of using setTimeOut to create a timer, we can use setInterval for shorter code

|  |  |  |
| --- | --- | --- |
|  |  |  |

1. Read about function scope in [this link](https://medium.com/better-programming/javascript-variables-the-difference-between-var-let-and-const-10efef5d2728) then answer the following question
   1. Can a variable declared by let, in a function still exists after the function finishes running?
   2. Can var variable declared by var, in a function still exists after the function finishes running?

Given the following HTML

1. You can get DOM (Read) by tag name using getElementsByTagName function, the result you get is NOT a single HTMLElement but a HTMLCollection, which can be treated as Array (you can use indexing [], for statement and length with it). Refer to this [link](https://www.w3schools.com/jsref/met_document_getelementsbytagname.asp).

<ul>

<li>News</li>

<li>Articles</li>

<li>Blog</li>

<ul>

Use getElementsByTagName function to get ALL of the <li> tag. Print/log out the first element (which is <li>Articles</li>) and then use for loop to print ALL of the DOM you get from the function.

1. You can get DOM (Read) by class name using getElementsByClassName function. The result you get, again, is NOT a single HTMLElement by a HTMLCollection. Refer to this [link](https://www.w3schools.com/jsref/met_document_getelementsbyclassname.asp).

Given the following HTML

<div>

<div class=”singer”>James Bay</div>

<div class=”singer”>Nate Ruess</div>

<div class=”singer”>Ed Sheeran</div>

<div class=”singer”>Steve Garrigan</div>

<div>

Use getElementsByTagName function to get ALL of the <div> tag. Print/log out the second element (which is <div class=”singer”>Nate Ruess</div>) and then use for loop to print ALL of the DOM you get from the function.

1. Learn how to remove (Delete) an HTMLElement using HTMLElement.remove() function, refer to [this](https://developer.mozilla.org/en-US/docs/Web/API/ChildNode/remove)
2. When using addEventListener, in the callback you can add a parameter / argument to get more information about the event, for example:

btn.addEventListener(‘click’, (e) => {

// Callback body

});

e is called [Event](https://developer.mozilla.org/en-US/docs/Web/API/Event) Object, which contains additional information about the events.

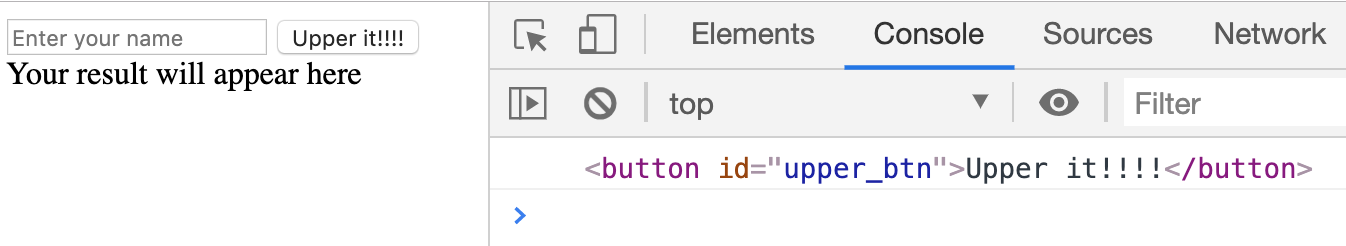
* 1. Catch ‘click’ event from 2 different buttons, id=”button1” and id=”button2”, in the callback body of the 2 events, print, or log out e.target. What is it?
  2. Catch a ‘keydown’ event from an input, in the callback body of the event, print / log out the e.key. What is it?

|  |  |
| --- | --- |
|  | **Review** |

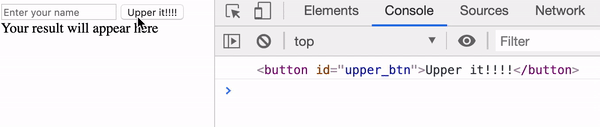
1. Write a function, using alert to show your name and one thing you wish to do this year
2. Write a function, using alert to show user’s name and one wish of the user this year, **name and wish are function parameters**
3. Write a function, using alert to show user’s name and one wish of the user in this year, name and wish are function parameters. **Handle the case of missing wish in function parameters**
4. Take a look at the following [UI](https://en.wikipedia.org/wiki/User_interface)

|  |  |  |
| --- | --- | --- |
|  |  |  |

* 1. Write an HTML for the UI above, create and link a JS file to it
  2. Get (Read) the ‘Upper it!!’ button



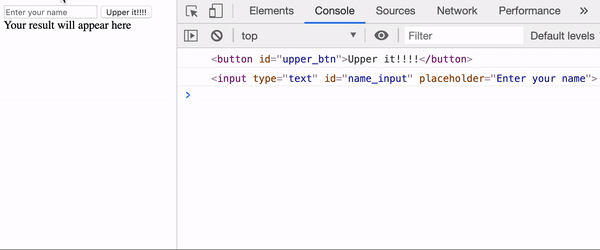
* 1. Catch event when ‘Upper it!!!!’ button is clicked



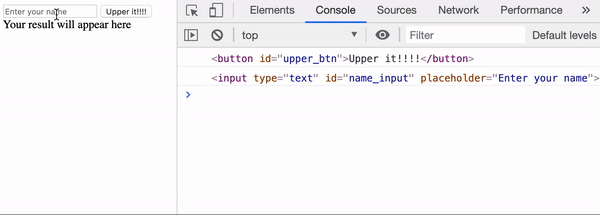
* 1. Get (Read) ‘Enter your name’ input



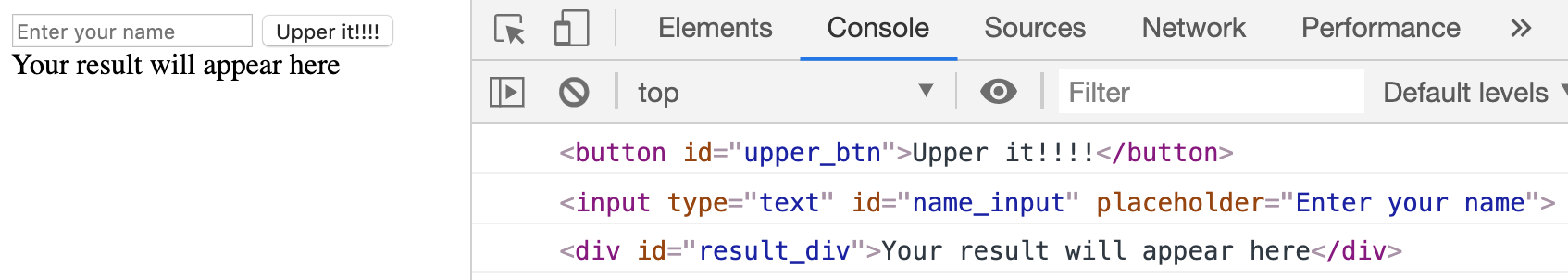
* 1. Get (Read) user name from the input



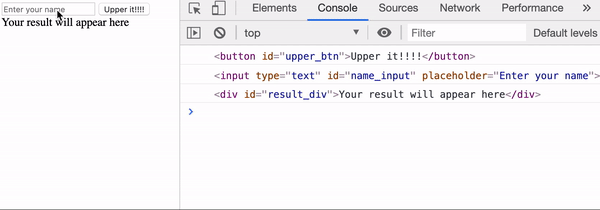
* 1. Convert user’s name to uppercase



* 1. Get (Read) the ‘Your result will appear here’ tag



* 1. Update the ‘Your result will appear here’ to the username uppercase version

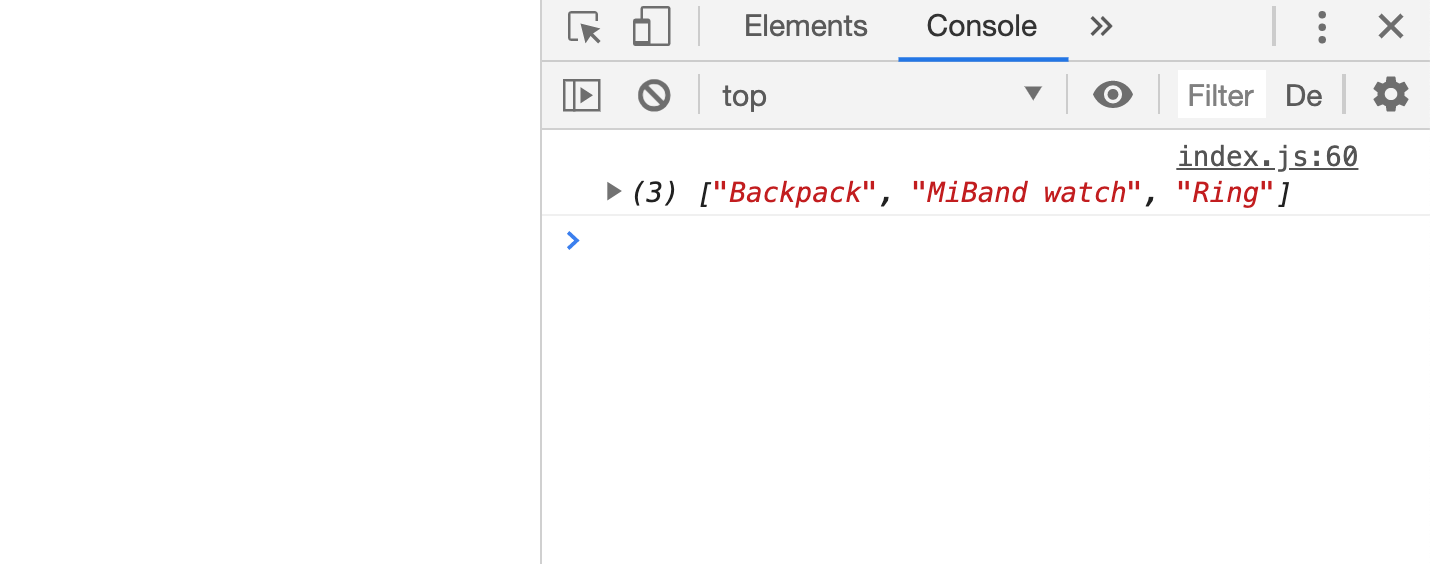


|  |  |
| --- | --- |
|  | **Serious exercices** |

1. Take a look at a list of belongings

|  |
| --- |
| Backpack |
| MiBand watch |
| Ring |

* 1. In JS, Initialize a variable named items to store the data above, log/print it out



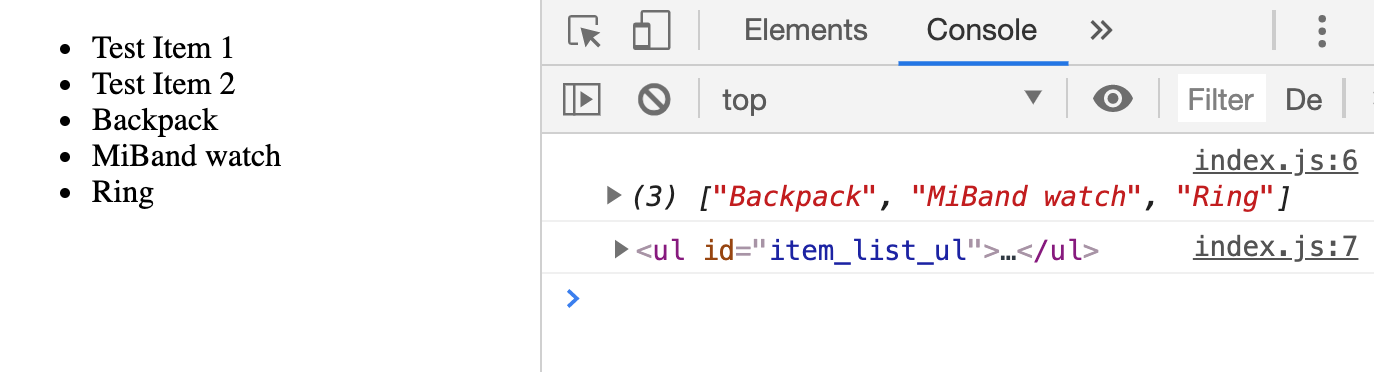
* 1. In HTML, create an <ul> with some test items. If you have not worked with unordered list before, watch the ‘HTML - List and Table’ video or see this [quick example](https://codepen.io/mindxvn/pen/vYYyNyW).

|  |  |
| --- | --- |
|  |  |

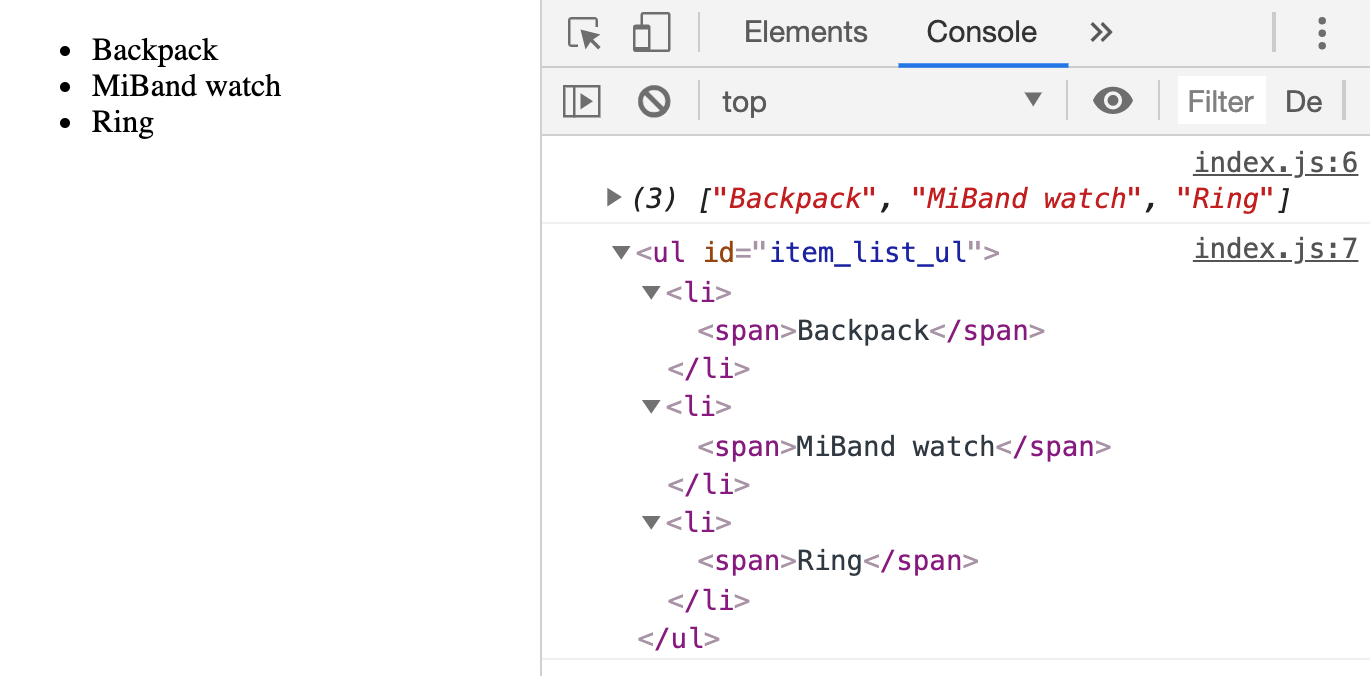
* 1. Get (Read) the list from DOM

|  |  |
| --- | --- |
|  |  |

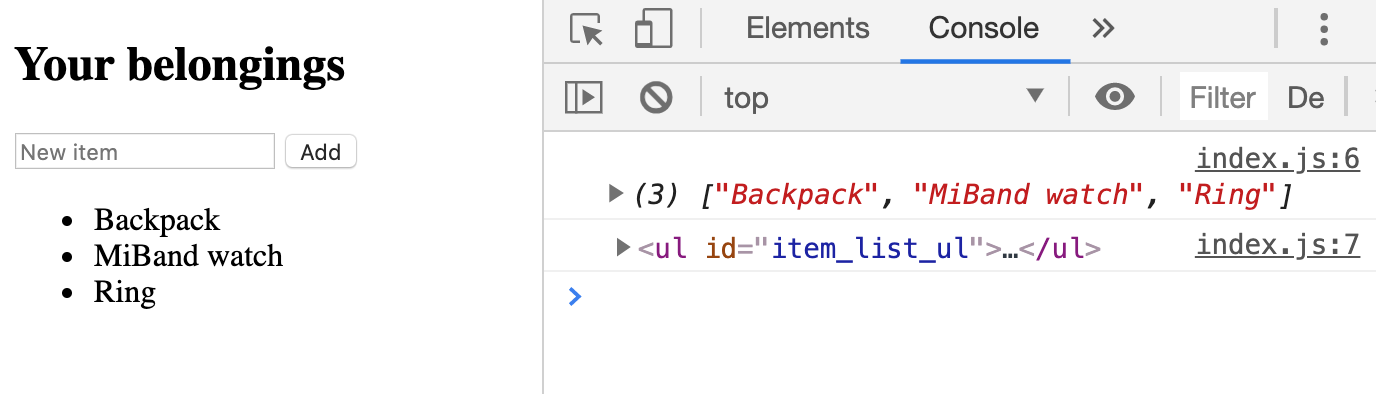
* 1. Show ALL of the data in items, each item is a <li> tag



The test items now can be removed



* 1. Update the UI (HTML) as follows:



* 1. Get the ‘New item’ input and ‘Add’ button

|  |  |
| --- | --- |
|  |  |

* 1. Register ‘Add‘ button click event

|  |  |
| --- | --- |
|  |  |

* 1. When ‘Add’ button is clicked, the value of ‘New item’ input

|  |  |
| --- | --- |
|  |  |

* 1. When ‘Add’ button is clicked, add the value of ‘New item’ input to items Array

|  |  |
| --- | --- |
|  |  |

* 1. Update UI (HTML) to reflect the update of items Array

|  |  |
| --- | --- |
|  |  |

* 1. Reset the input after adding new item

|  |  |
| --- | --- |
|  |  |

* 1. Add remove button for each item

|  |  |
| --- | --- |
|  |  |

* 1. Register to each ‘remove’ buttons’ event

|  |  |
| --- | --- |
|  |  |

* 1. Each time a ‘remove’ button clicked, identify the index of the clicked item

|  |  |
| --- | --- |
|  |  |

* 1. Remove from items based on clicked item index

|  |  |
| --- | --- |
|  |  |

* 1. Update UI (HTML) to reflect the update of items Array

|  |  |
| --- | --- |
|  |  |