

JS Advanced Regular Exam – 19 Feb 2022

Problem 1. Ski Lift

Environment Specifics

Please be aware that every JS environment may **behave differently** when executing code. Certain things that work in the browser are not supported in **Node.js**, which is the environment used by **Judge**.

The following actions are **NOT** supported:

- `.forEach()` with **NodeList** (returned by `querySelector()` and `querySelectorAll()`)
- `.forEach()` with **HTMLCollection** (returned by `getElementsByClassName()` and `element.children`)
- Using the **spread-operator** (`...`) to convert a **NodeList** into an array
- `append()` in Judge (use only `appendChild()`)
- `prepend()`
- `replaceWith()`
- `replaceAll()`
- `closest()`
- `replaceChildren()`
- Always turn the collection into a **JS array** (`forEach`, `forOf`, et.)

If you want to perform these operations, you may use `Array.from()` to first convert the collection into an array.

Use the provided skeleton to solve this problem.

Write the missing functionality of this user interface. The functionality is divided in the following steps:



The image shows a web interface for buying ski lift tickets. The background is a snowy mountain landscape with ski lift chairs. The interface is divided into three main sections: 'Buy ticket', 'Ticket Preview', and 'Confirm ticket'. The 'Buy ticket' section contains form fields for 'First name:', 'Last name:', 'Number of people:', 'From Date:' (with a date picker), and 'Days:'. Below these fields is a 'Next step =>' button. The 'Ticket Preview' and 'Confirm ticket' sections are currently empty.

Your Task

Write the missing JavaScript code to make the **Ski Lift** work as expected:

All fields (**First Name**, **Last Name**, **Number of people**, **From date** and **Number of days**) are filled with the correct input.

- **First Name, Last Name, Number of people, From date and Number of days** are **non-empty strings**. If any of them is empty, the program should not do anything.

1. Getting the information from the form

- On clicking the ["**Next step**"] button the information from the input fields is listed in the "**info-ticket**" section. For the input fields a **list item** is **added** to the "**ticket-info-list**" unordered list.
- The text format and order for the list item should be the same as on the picture below.
- When the button is clicked, the input fields must be cleared and the ["**Next step**"] button must be **disabled**. At the same time the "**Edit**" and the "**Continue**" buttons must be **enabled**.

The HTML structure looks like this:

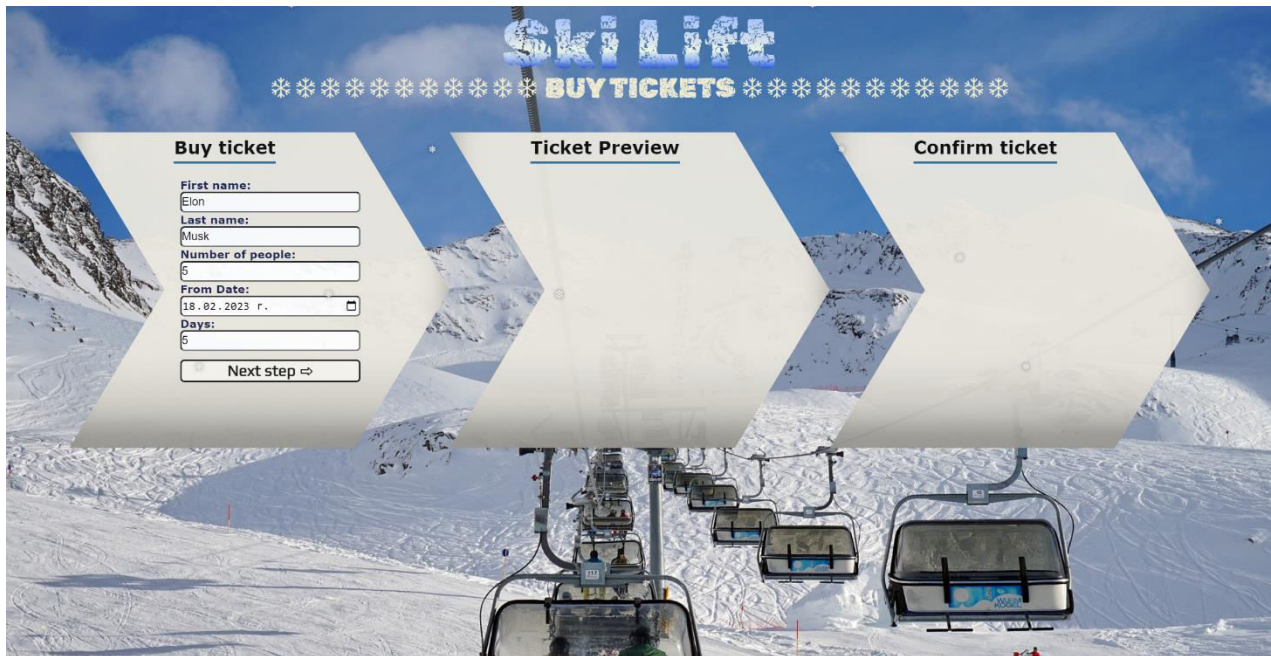
```

▼ <div class="first-container"> flex
  <h1>Ticket Preview</h1>
  ▼ <ul class="ticket-info-list">
    ▼ <li class="ticket">
      ▼ <article> == $0
        <h3>Name: Elon Musk</h3>
        <p>From date: 2023-02-18</p>
        <p>For 5 days</p>
        <p>For 5 people</p>
      </article>
      <button class="edit-btn">Edit</button>
      <button class="continue-btn">Continue</button>
    </li>
  </ul>
</div>

```

The functionality here is the following:

- When the "**Edit**" button is clicked, all of the information is loaded in the input fields from step 1 and the ["**Next step**"] button is enabled again.



- The list item must be removed from the " ticket-info-list" and all of the information must go back to the input fields again .

```
<div class="first-container"> flex == $0
  <h1>Ticket Preview</h1>
  <ul class="ticket-info-list"> </ul>
</div>
</div>
```

- When the "Continue" button is clicked, the information from "ticket-info-list" unordered list must be transferred to "confirm-ticket" in the same HTML structure. For you, this means removing everything inside of the ul with class "ticket-content " and adding in "confirm-ticket", the list item with same information and class "ticket-info-list" , delete "Edit" and "Continue" buttons and add new buttons "Confirm" and "Cancel" with class "confirm-btn" and "cancel-btn" .



- This is HTML structure of "confirm-ticket" unordered list:

```

<div class="first-container"> flex
  <h1>Confirm ticket</h1>
  <ul class="confirm-ticket">
    <li class="ticket-content">
      <article>
        <h3>Name: Elon Musk</h3>
        <p>From date: 2023-02-18</p>
        <p>For 5 days</p>
        <p>For 5 people</p>
      </article>
      <button class="confirm-btn">Confirm</button>
      <button class="cancel-btn">Cancel</button>
    </li>
  </ul>
</div>

```

- When the "Cancel" button is clicked, the list item must be removed, from the "confirm-ticket", the ["Next step"] button is enabled again.



- When the "Confirm" button is clicked, the <div> element with id="main" must be removed, from the body and you must add <h1> element with id="thank-you" and text "Thank you, have a nice day! " and <button> with id="back-btn" and text "Back ".



The HTML structure looks like this:

```
<body id="body" data-new-gr-c-s-check-loaded="14.1091.0" data-gr-ext-installed> == $0
  <div class="snowflakes" aria-hidden="true">...</div>
  <script src="app.js"></script>
  <!-- Code injected by live-server -->
  <script>...</script>
  <h1 id="thank-you">Thank you, have a nice day!</h1>
  <button id="back-btn">Back</button>
</body>
```

- On clicking the [**"Back"**] button you must **reload** the page.

Submission

Submit only yours **solution()** function.

GOOD LUCK... 😊