# JS Advanced Exam

# Problem 1. Hotel Reservation

**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:

**Your Task**

**Write the missing JavaScript code** to make the **Hotel Reservation** work as expected:

All fields **(First Name, Last Name, Check-in date, Check-out date** and **Number of guests)** are **filled with the correct input.**

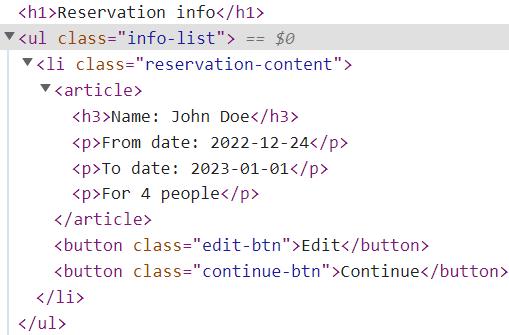
* **First Name, Last Name, Check-in date, Check-out date** and **Number of guests** are **non**-**empty** **strings** and **Check-in** dateis **before Check-out** date. If any of them is empty, the program should not do anything.

1. **Getting the information from the form**

## 

* On clicking the ["Next"] button the information from the input fields is listed in the **"**Reservation info" section. For the input fields a **list item** is **added** to the **"**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"] button must be **disabled**. At the same time the "Edit" and the "Continue" buttons must be **enabled**.

The HTML structure looks like this:



**The functionality here is the following:**

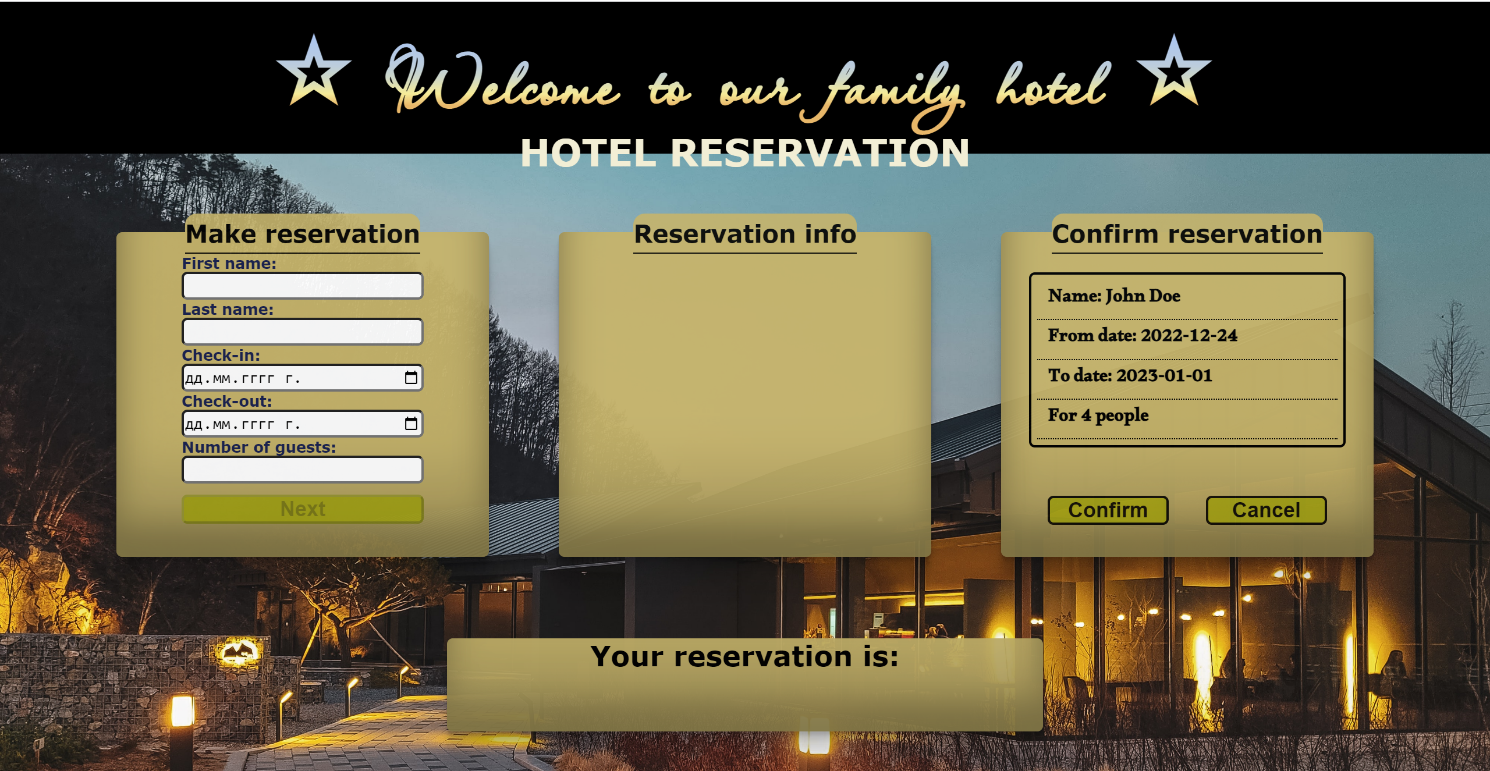
* **When the "**Edit" **button is clicked, all of the information is loaded in the input fields from step 1 and the** ["Next"] **button is enabled again.**

****

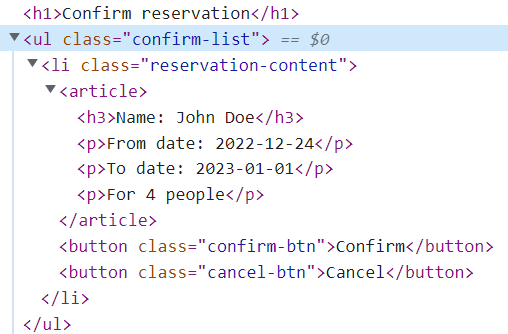
* **The list items must be removed from the** "info-list" and **all of the information must go back to the input fields again**.

****

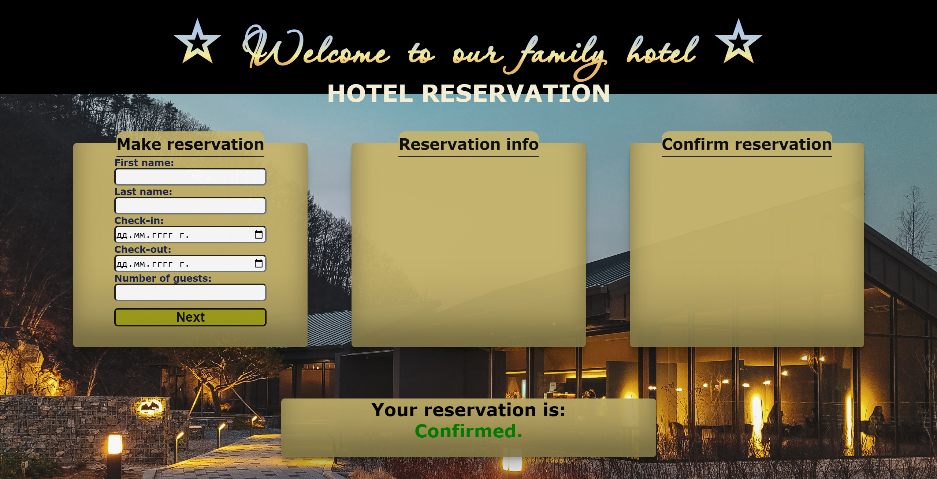
* **When the** "Continue" **button is clicked,** **the information from** "info-list" unordered list must be transferred to "confirm-list" in the same HTML structure**. For you, this means removing everything inside of the ul with class =** "info-list" **and adding in** "confirm-list", **the list item with same information and change** "Edit" and "Continue" buttons to"Confirm" and "Cancel" with changed **class**.

****

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



* **When the** "Confirm" **button is clicked,** **the list item must be removed, from the** "confirm-list", **the** ["Next"] **button is enabled again** you must add new class and add text to <h1> element with id="verification".If "Confirm" button is clicked class must be "reservation-confirmed"and text is "Confirmed.".
  + "Confirm" button:

****

* **When the** "Cancel" **button is clicked,** **the list item must be removed, from the** "confirm-list", **the** ["Next"] **button is enabled again** you must add new class and add text to <h1> element with id="verification". If "Cancel" button is clicked class must be "reservation-cancelled"and text is "Cancelled.".
* "Cancel" button:

****

# Submission

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

*GOOD LUCK… ☺*