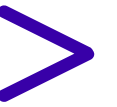


<InfPALS/>

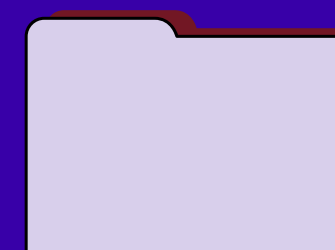
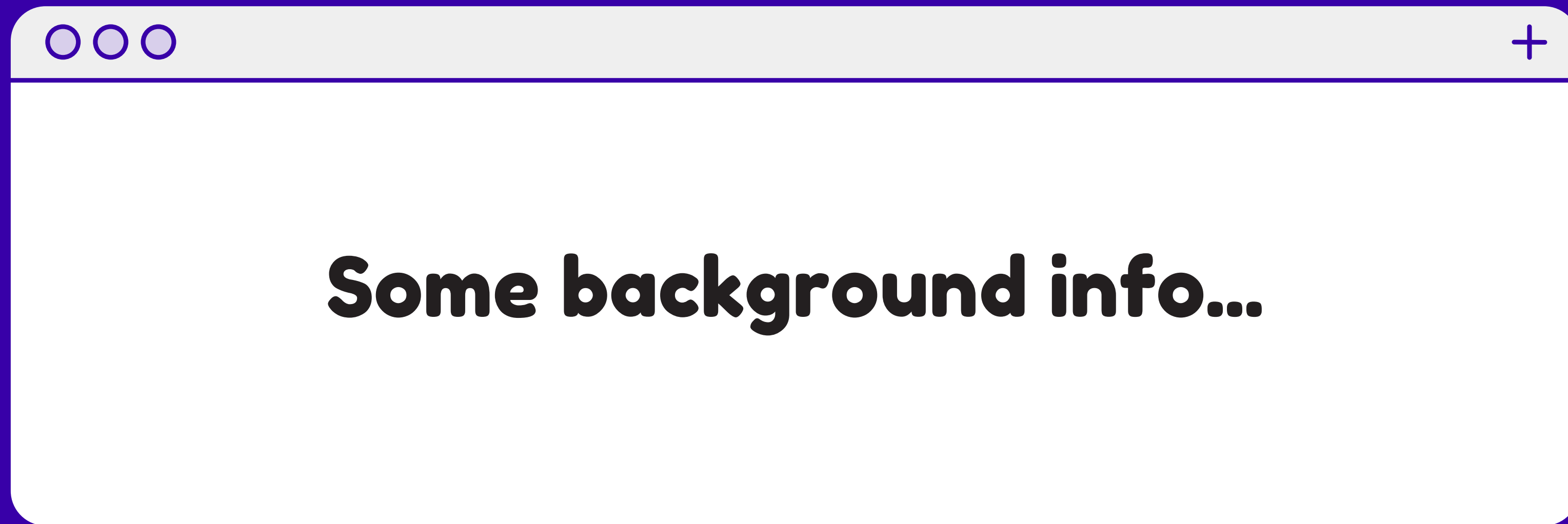


# What are we going to do today?



## Rebuilding Arrays

## Adding Items



# JavaScript

- Programming language made for developing websites
- Most know for building frontend web applications
- Used in combination with HTML and CSS



{.js}

**JavaScript**

# JavaScript Structures

## Function structure

- function function\_name  
    (parameter\_name){  
        body\_of\_the\_function  
    }  
)

{.jjs}

# JavaScript

# JavaScript Structures

## For Loop Structure

- `for ( let i = 0; i < number;  
i++){  
body_of_the_function  
}`

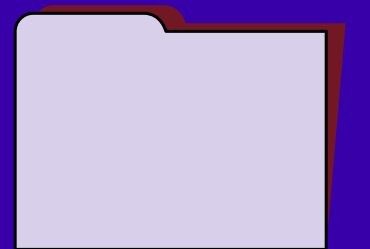
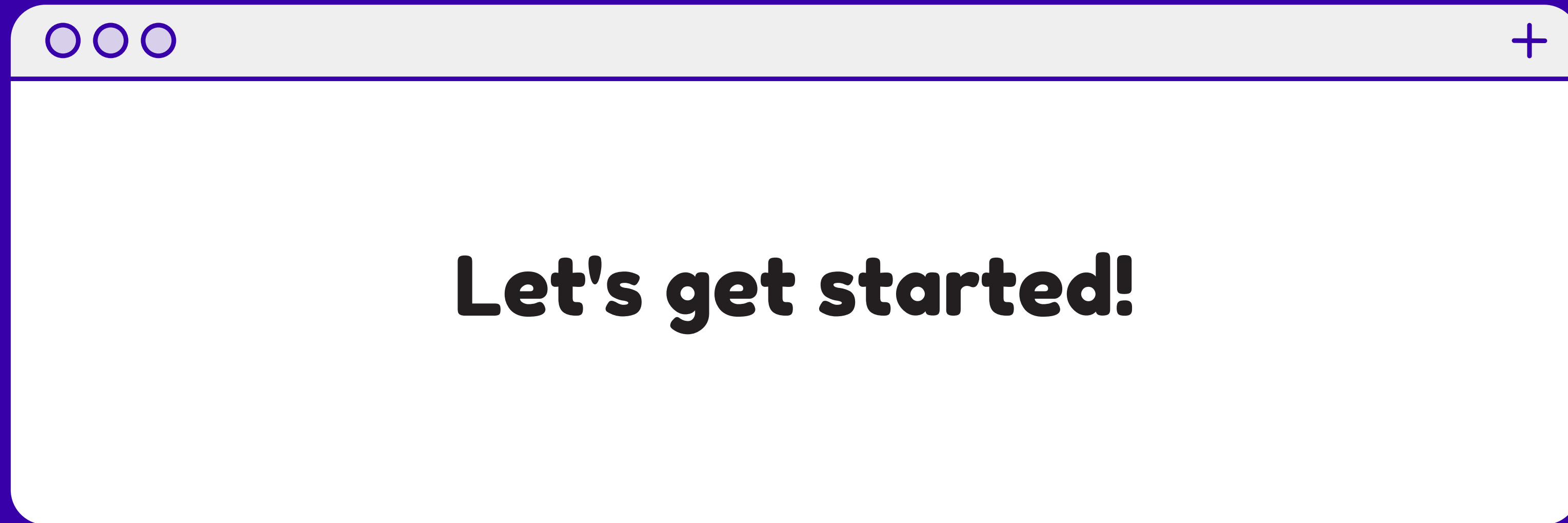
## Appending elems to the end of Array/List

- `name_of_array.push(elem)`

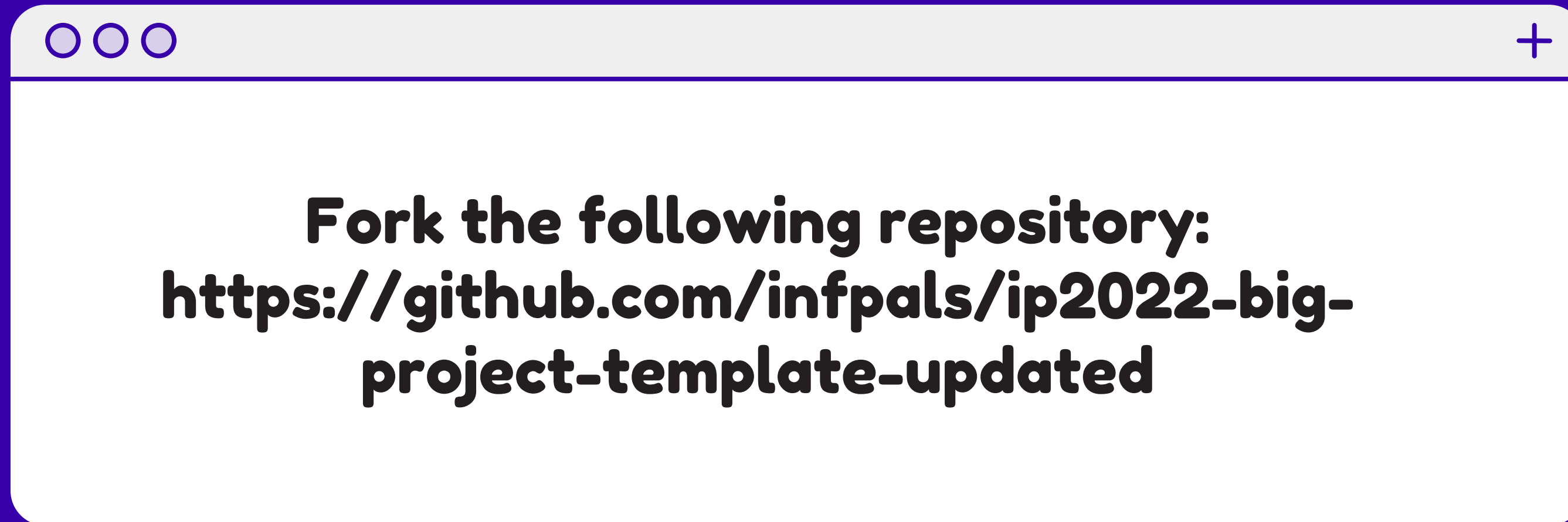
{.jst}

JavaScript

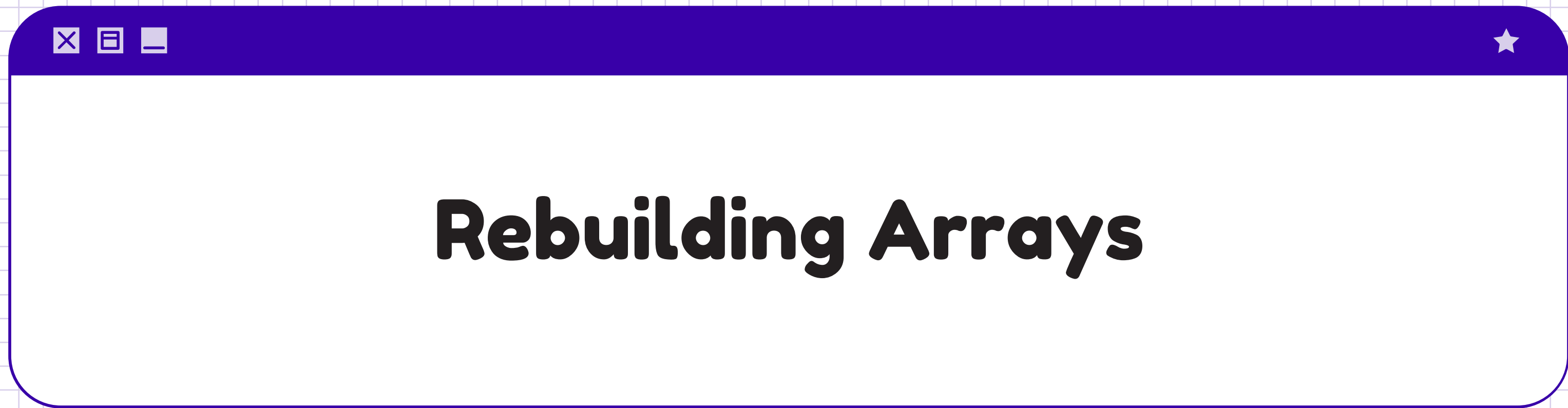
We hope you are having fun! ;)



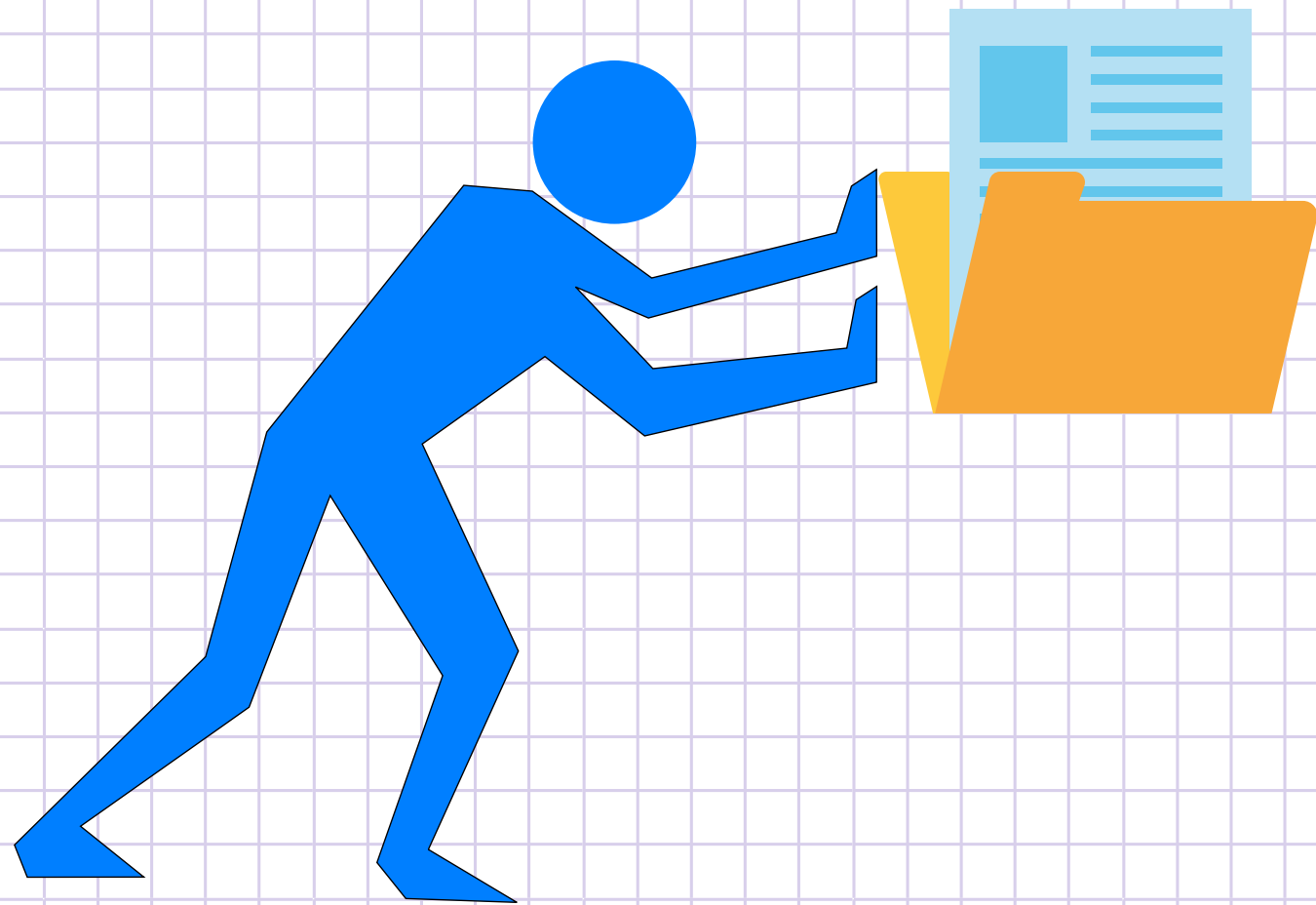
If you were not here last week ...







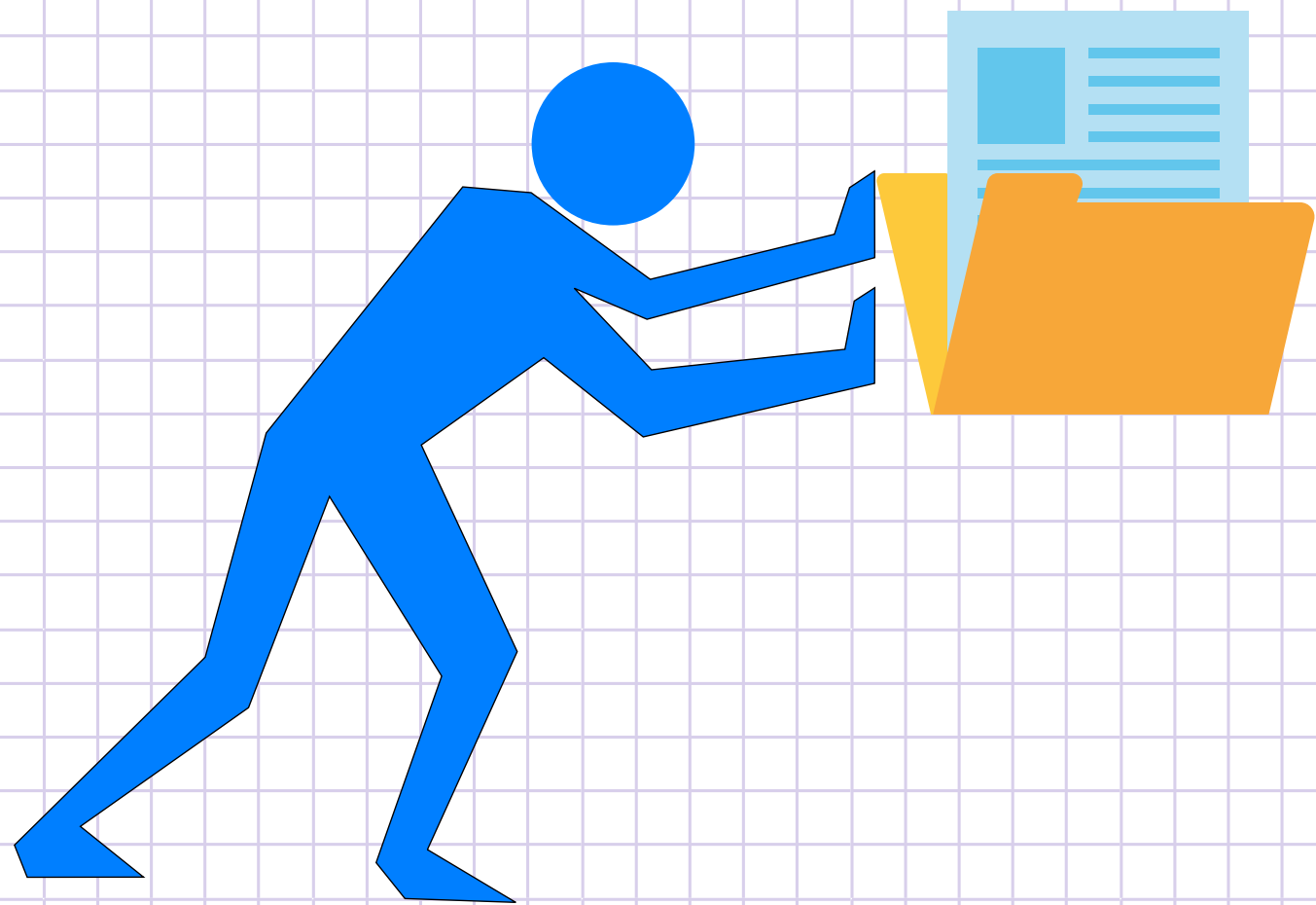
# Rebuilding Arrays



1. We need a new function that allows our arrays and local storage to reflect dragged and dropped items in new columns!

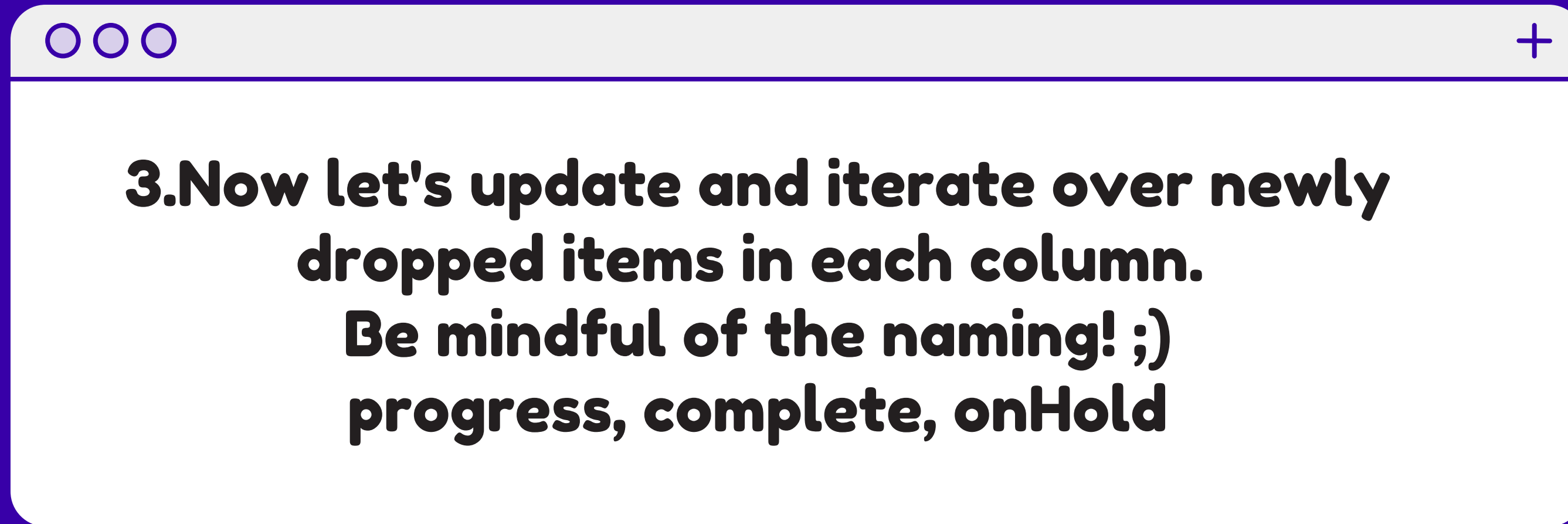
- Create a new function below our `updateDOM()` called `rebuildArrays()` without passing it any parameter
- Call this new function in our drop function

# Rebuilding Arrays



2. Let's iterate over each item in our backlog list (we got this from html) and update our local storage in rebuildArrays:
- First, set `backlogListArray = []`;
  - Create a for loop that loops while `i < backlogList.children.length`
  - Inside this loop, we want to add the newly dropped item to our array through `backlogListArray.push(backlogList.children[i].textContent)`;

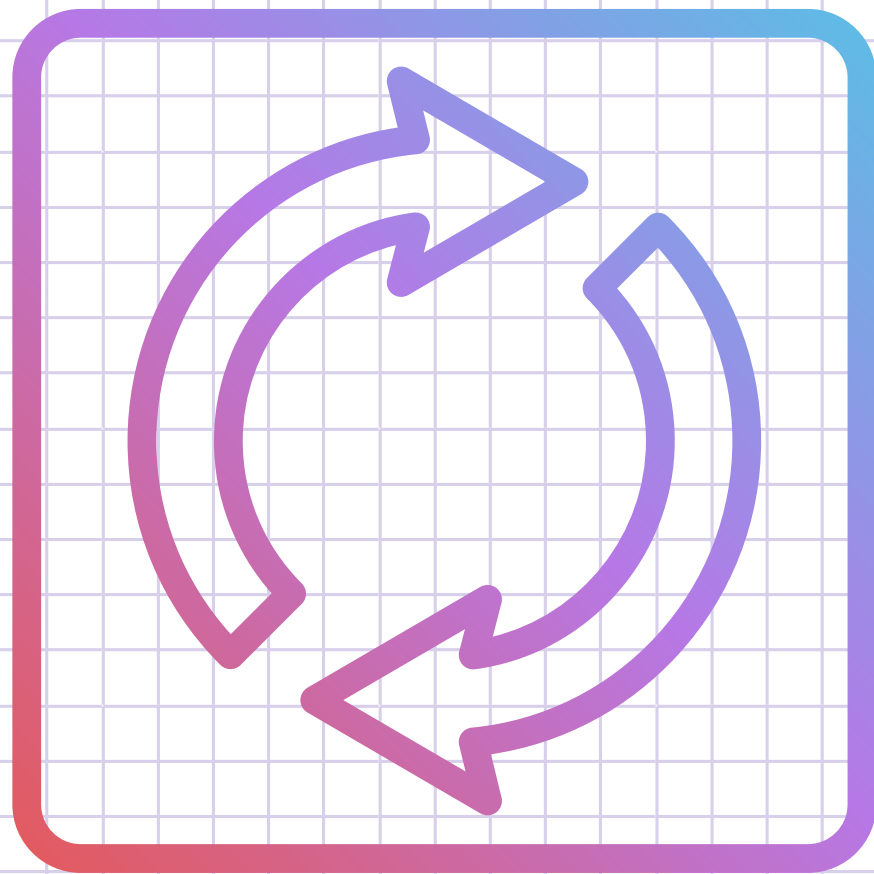
If you were not here last week ...



**Open index.html in your browser. What happens if you  
reload the browser?**



# Rebuilding Arrays

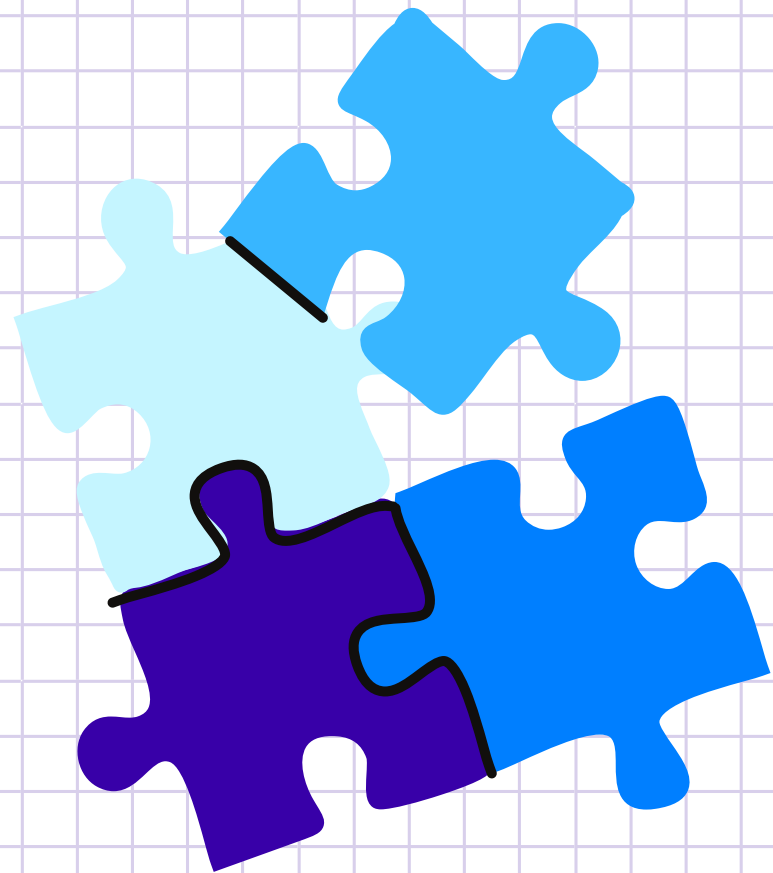


4. Let's update our DOM with the updated Arrays

- In our function `rebuildArrays` call the function `updateDOM()`;

# Rebuilding Arrays

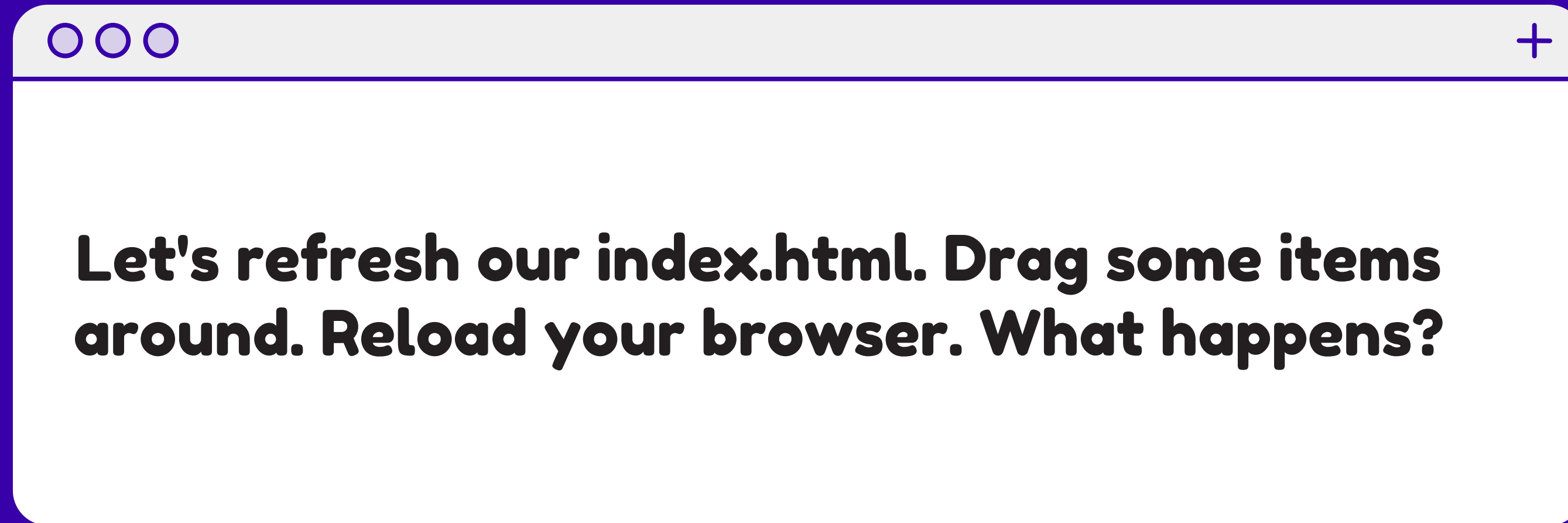
Put it together!

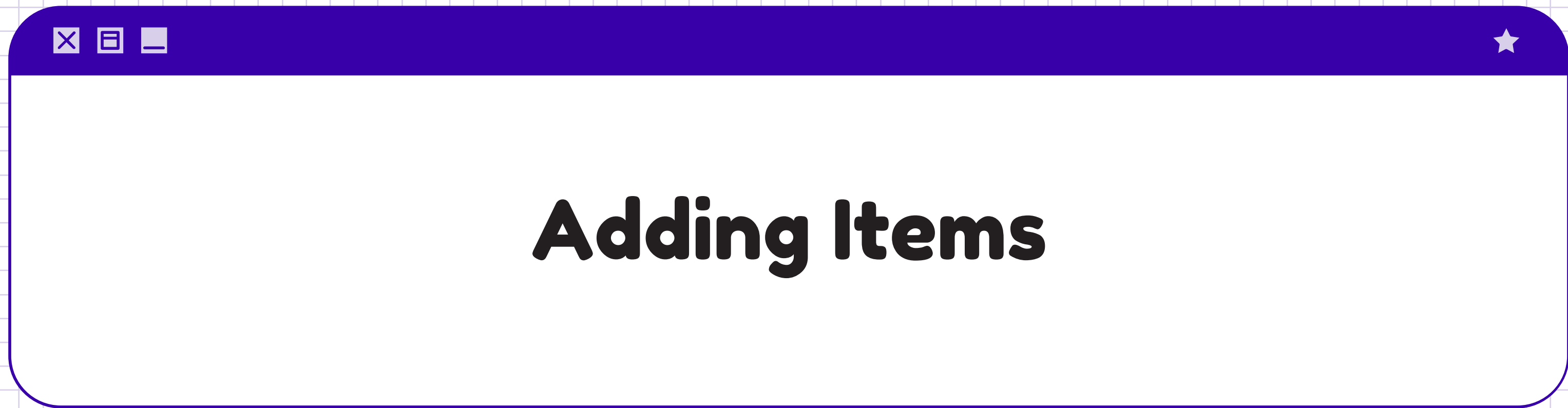


5. Lastly, let's save the updated columns

- At the end of UpdateDOM function
  - call `updateSavedColumns()` function
  - set `updateOnLoad` to true

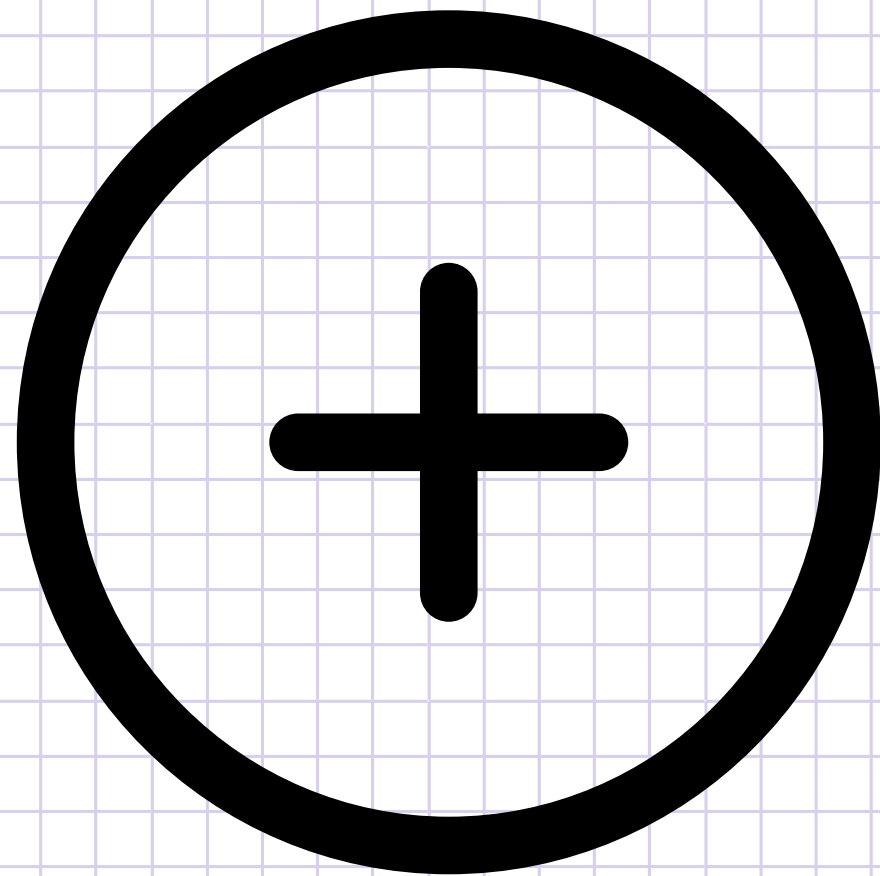
If you were not here last week ...





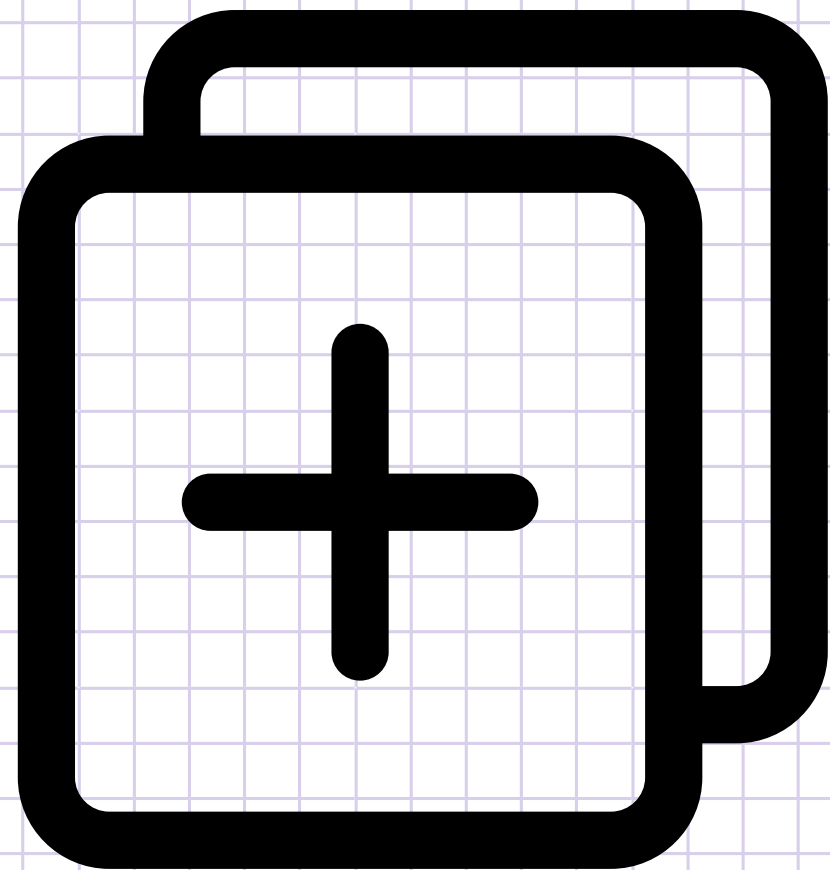


# Adding Items



1. We want to add event listeners to our HTML. This ensures what happens when we click on + add item and save item button.
  - In backlog column / section find section for the add button.
    - In the opening tag of section with class 'add-btn' add  
`onclick="showInputBox(0)"`
    - In the opening tag of the section with class 'add-btn solid' add  
`onclick="hideInputBox(0)"`

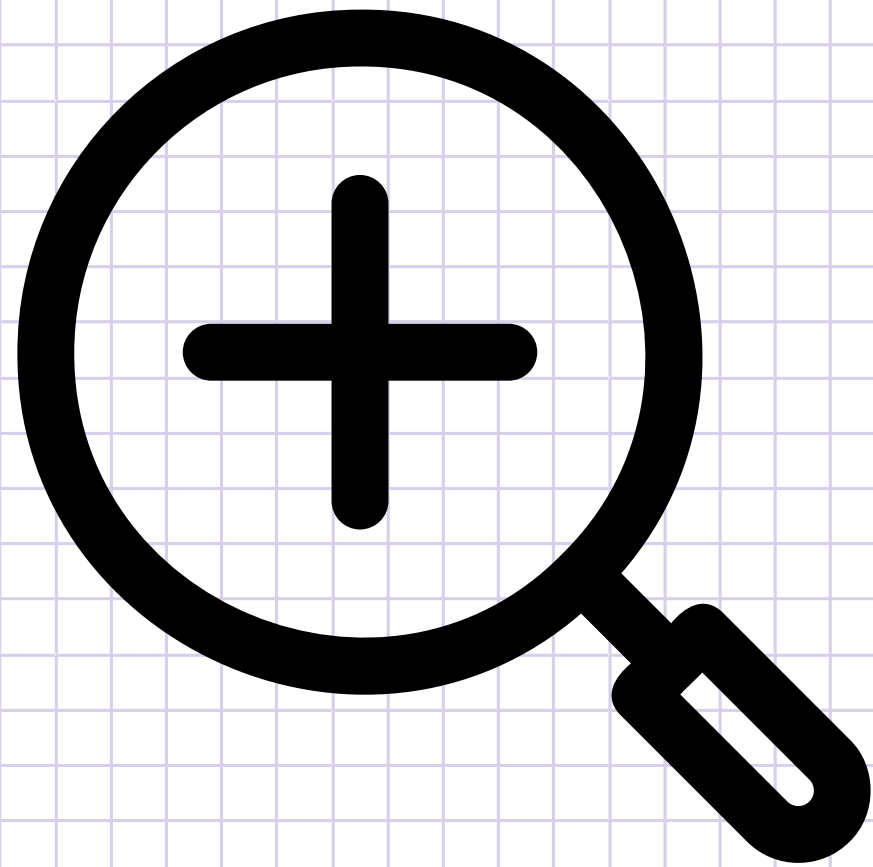
# Adding Items



2. Now add  
`onclick="showInputBox(0)` and  
`onclick="hideInputBox(0)"` to each  
column

PS: Don't forget to change the 0s into  
the appropriate column numbers as  
we did in the function `dragEnter`

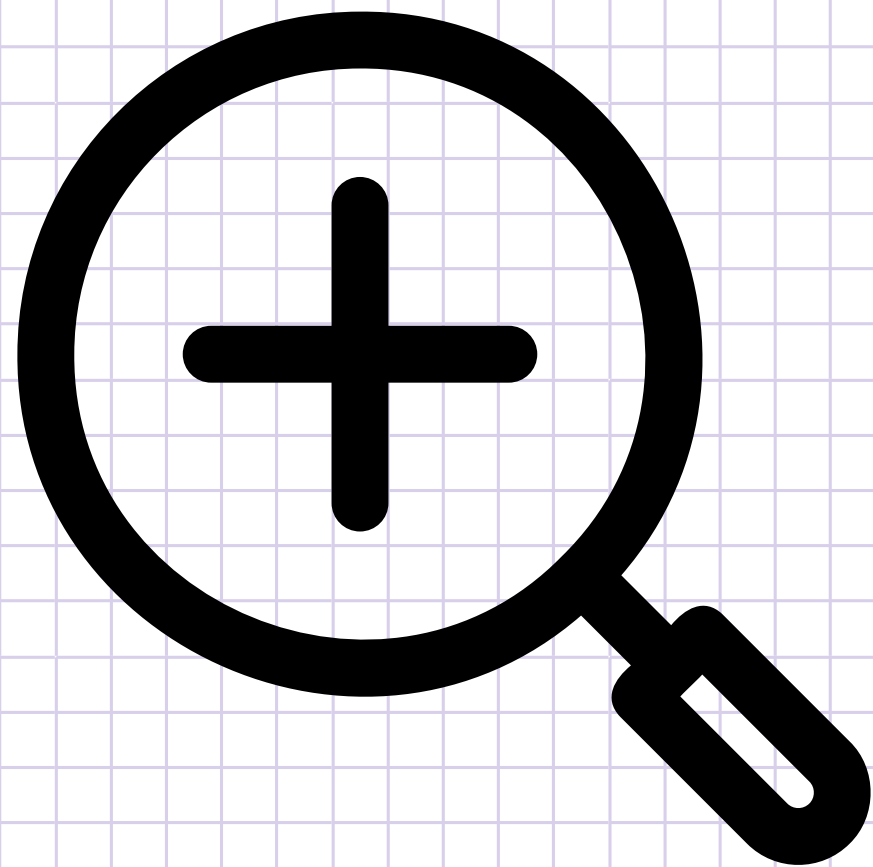
# Adding Items



3. Now, let's write our function `showInputBox` in JS.

- write function `showInputBox` passing it the parameter `column`
- Inside our function we want to say that once we click on + add item btn we want to hide it and show the save button and container
  - Inside `showInputBox` add `addBtns[column].style.visibility = 'hidden';` and `saveItemBtns[column].style.display = 'flex';` and `addItemContainers[column].style.display = 'flex';`

# Adding Items



4. Now, let's write our function `hideInputBox` in JS.

- write function `hideInputBox` passing it the parameter `column`
- Inside our function we want to say that once we click on save item btn we want to hide it and the container as well and we want to show the + add button
  - Inside `showInputBox` add `addBtns[column].style.visibility = 'visible';` and `saveItemBtns[column].style.display = 'none';` and `addItemContainers[column].style.display = 'none';`



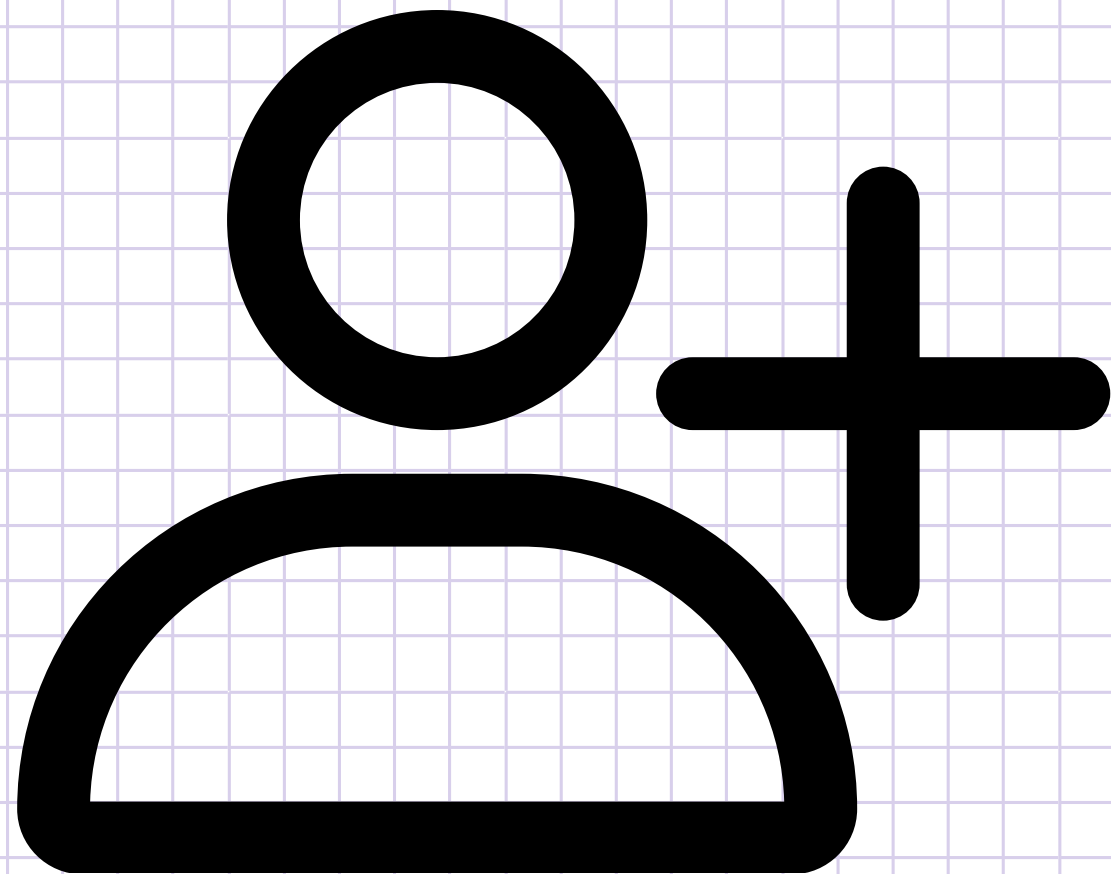
**Try your new add item button. What happens?**  
**Hint: We made a mistake in our html**



**In order to resolve our problems we need to shift our closing tags. Find tag with class add-btn-group and move its' closing tag above the tag with class add-container.  
Try your + add item btn now :)**

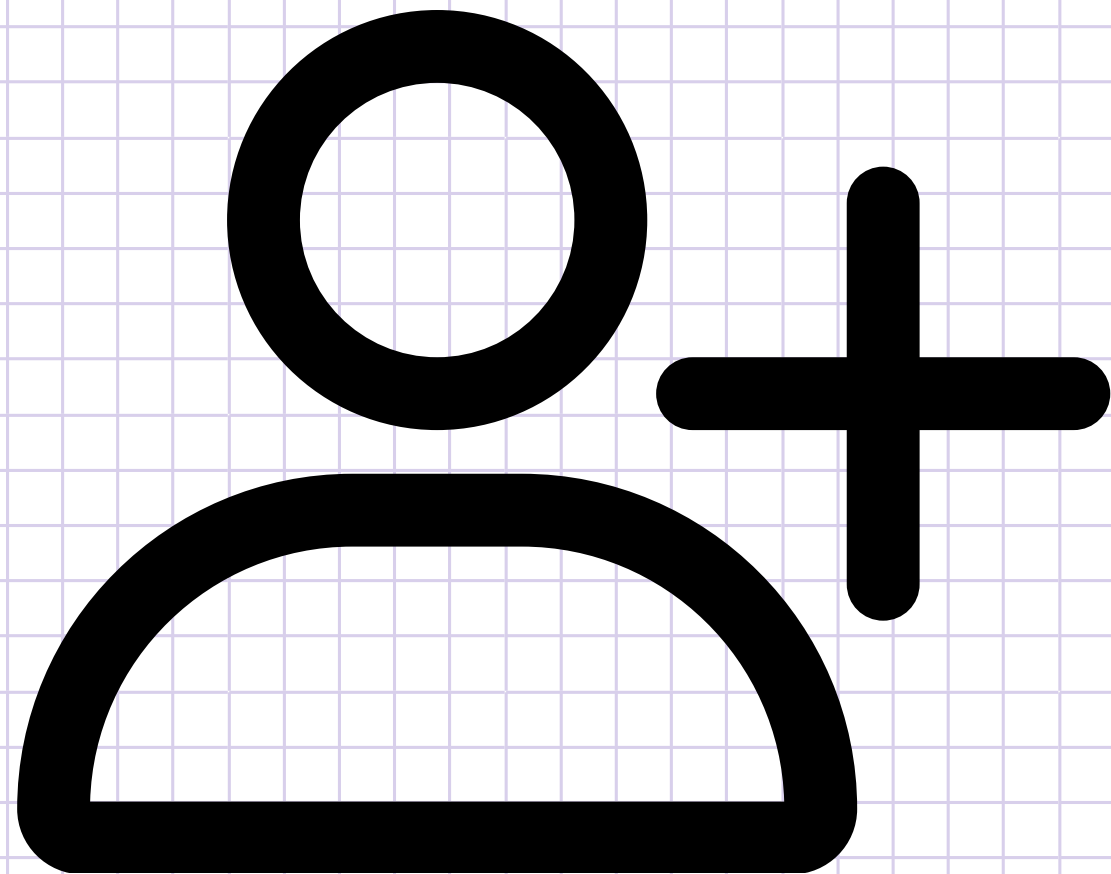


# Adding Items



5. Now we want to be able to write something into our content box
- We are able to do this if you add `contenteditable="true"` into each `columns` tag with class `add-item`
    - This is in our HTML
    - In the section `add-container`

# Adding Items

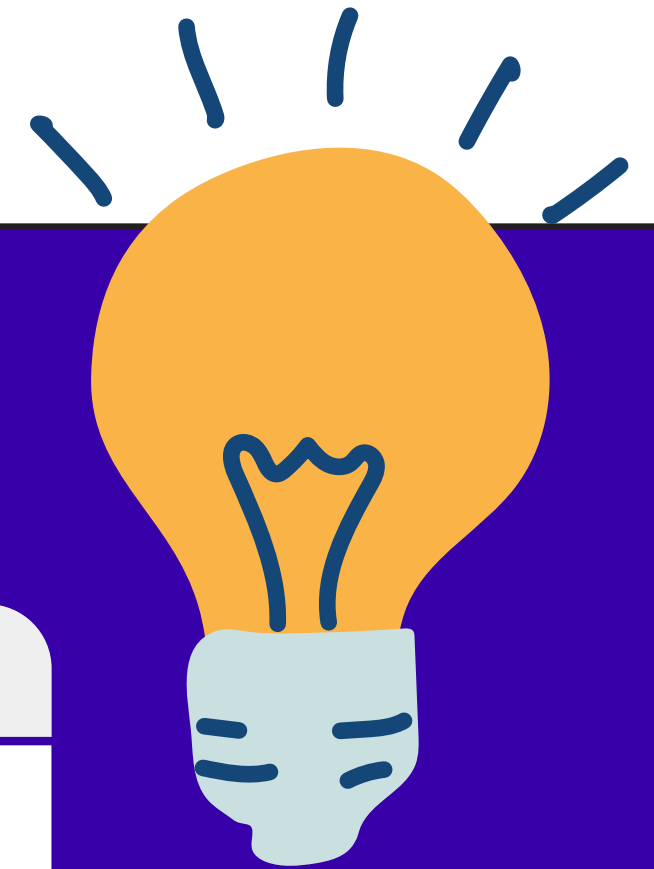


6. We also want to add the content of the new item to that given column
- We need to write new function in our js called `addToColumn()` and pass it parameter column
  - Inside we need to get the text content of the container, the column which we want to add it to , append that item to the array of items and update DOM
    - `const itemText = addItems[column].textContent;`
    - `const selectedArray = listArrays[column];`
    - `selectedArray.push(itemText);`
    - `updateDOM();`





**Try to add your new item to the Column you want! What happens? Does it work?**  
**Try to look at the console and figure out where is the problem.**

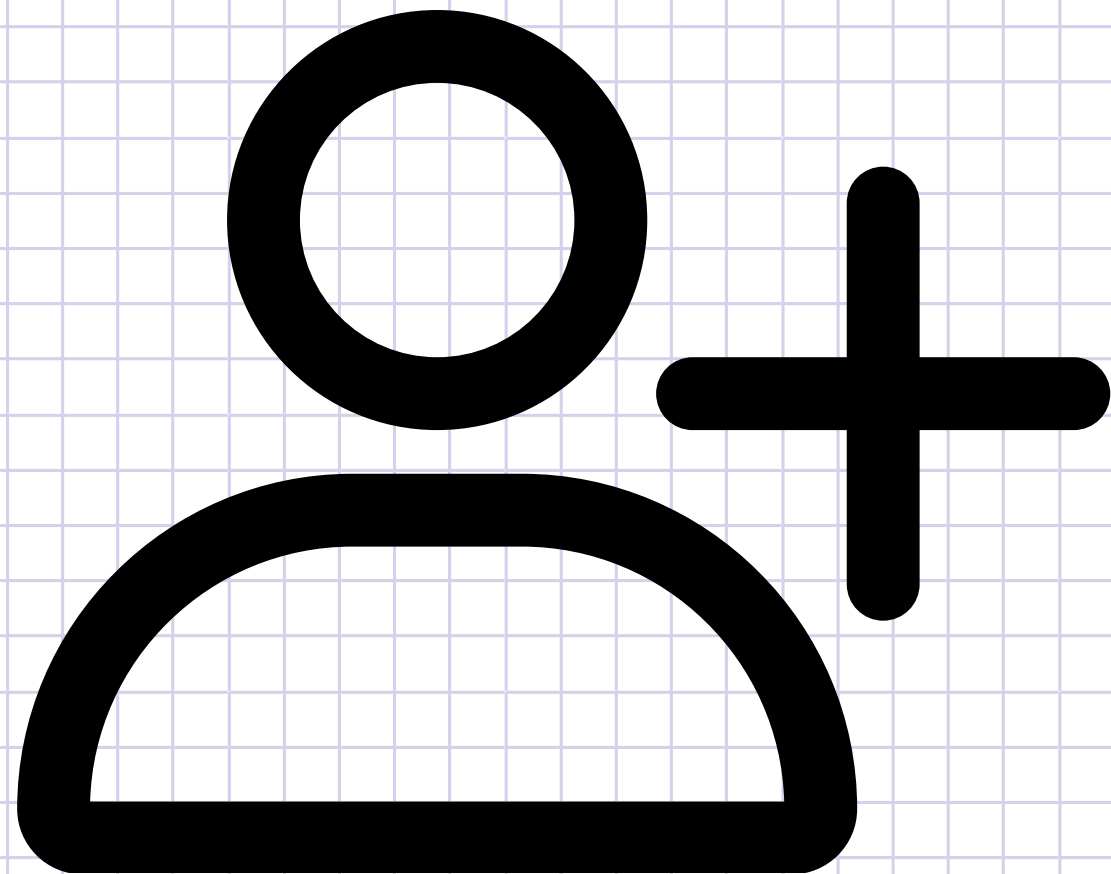


**Add addToColumn(column) inside our  
hideInputBox(column) function  
Under the comment //Initialize Arrays  
add let listArrays = [];  
and in the function updateSavedColumns you  
need to add  
listArrays = [backlogListArray, progressListArray,  
completeListArray, onHoldListArray];**

**Try it now!**



# Adding Items



6. Lastly, what happens if you try to add two items to one column of your choice?

- We want a blank container every time we add a new item and for that we add in the function `addToColumn`
  - `addItem[column].textContent = "";`

**Try it now!**  
**Everything should work!**



**Apply to be  
InfPals Leader  
in 2022/2023**

