

## Unit 2: Pagination & Content Filter Project

### How to approach the pagination project

There are many ways you could create this project. If you need some help thinking about how to structure your JavaScript programming, here's an approach we suggest:

Create several functions that handle small tasks. Use those small functions inside new, larger functions to accomplish more complex tasks. In software design this is sometimes called "composition" or "aggregation".

### Functions for a "Meets Expectations" Grade:

There are two main functions you'll need for this project:

- ❑ **showPage:** This smaller function displays a "page" or a list of ten students based which "page number" the user selected.
- ❑ **appendPageLinks:** This function creates the links to the different "pages" or lists of students. It will call the showPage function to display the proper list of students based on which link the user clicks. For example, clicking the link to page 1, tells the showPage function to display the first 10 students.

### The showPage Function

This function builds a list of ten students and displays it on the page. The students displayed depends on the page number passed to this function. The function should loop through all the students in the list and determine if each student is on this page. It will show all the students on this page and hide the rest. Here are some ideas for how this could work in code:

```
function showPage(/* arguments here for page number and student list */) {  
  // first hide all students on the page  
  // Then loop through all students in our student list argument  
  // if student should be on this page number  
    // show the student  
}
```

## The appendPageLinks Function

This function creates all the page links based on a list of students. It will determine how many pages we need based on the list's length, create a list of links for each page, and append that list to the page. When each link is clicked, we'll use the showPage function to display the corresponding page, and mark the active link.

```
function appendPageLinks(/* take a student list as an argument */) {  
  // determine how many pages for this student list  
  // create a page link section  
  // “for” every page  
    // add a page link to the page link section  
  // remove the old page link section from the site  
  // append our new page link section to the site  
  // define what happens when you click a link  
    // Use the showPage function to display the page for the link clicked  
    // mark that link as “active”  
}
```

## Exceeds Grade: The Search Function

The function takes a value from the input field, and compares it to each student in the list. If that value is found inside the name or email of a student, that student is added to a new "matched" list. If the "matched" list is empty, then display a message that no matching students were found. Otherwise, call the appendPageLinks function to create new pagination for the search results. Then call the showPage function to display the first page of matched results.

```
function searchList() {  
  // Obtain the value of the search input  
  // remove the previous page link section  
  // Loop over the student list, and for each student...  
    // ...obtain the student's name...  
    // ...and the student's email...  
    // ...if the search value is found inside either email or name...  
      // ...add this student to list of “matched” student  
  // If there's no “matched” students...  
    // ...display a “no student's found” message  
  // If over ten students were found...  
    // ...call appendPageLinks with the matched students  
  // Call showPage to show first ten students of matched list  
}
```