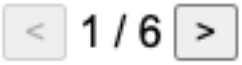
HTML CODE

```
<body>
    <section>
        <div id="controls"> <!-- Pager controls -->
            <button id="back"> < </button> <!-- Previous page button -->
            <span id="page-info"></span> <!-- Info on current page -->
            <button id="next"> > </button> <!-- Next page button -->
        </div>
        <div id="movie-list"></div> <!-- Movies per page -->
    </section>
    <template id="movie-template"> <!-- Movie template -->
        <div id="movie">
            <h2 id="title"></h2>
            <h3 id="year"></h3>
            <p id="director"></p>
            <ul id="genre"></ul>
        </div>
    </template>
</body>
```



ARRAY OF OBJECTS (JSON)

```
const data = [
    {
        id: 3,
        title: "The Dark Knight",
        year: 2008,
        director: "Christopher Nolan",
        genre: "Action, Crime, Drama"
    },
    {...}
]
```

PROBLEM BREAKDOWN

- Knowing how many pages there are ➔ calculation = roundUp(numMovies/moviesPerPage)
- Knowing how many movies there are in total (numMovies = length of the movie array)
- Knowing how many movies will be per page (constant moviesPerPage)
- Knowing which page we are on (currentPage = variable that we will initialize to 1)
- Knowing which movies we need to display ➔ subset of the movie array
    - Start of the subset = (currentPage - 1) * moviesPerPage
    - End of the subset = start + moviesPerPage
    - Get a subarray from start to end
- Function to draw the subset of movies
    - Write the current page number
    - Set an area containing all the movies
    - Clear the area to make room for the new subset

        - Draw the current subset of movies with a loop
            - Create a clone of the template
            - Fill in each label with the content of each movie
            - For the genre, convert it to an array and iterate through it to create a list
            - Add the movie to the movie area
            - Move on to the next movie
    - Add functionality to the buttons
        - When a button is pressed, increase or decrease the page number
        - Call the drawing function
    - Disable the buttons at the ends
        - If we are on the first page, disable the back button
        - If we are on the last page, disable the forward button
    - By default, the page...
        - Set the current page to 1
        - Paint that page, as well as the buttons

```
// --
// Variables
// --
const movieListDOM = document.querySelector("#movie-list");
const backButtonDOM = document.querySelector("#back");
const pageInfoDOM = document.querySelector("#page-info");
const nextButtonDOM = document.querySelector("#next");
const movieTemplate = document.querySelector("#movie-template").content;
const elementsPerPage = 5;
let currentPage = 1;
const data = [
    {
        id: 3,
        title: "The Dark Knight",
        year: 2008,
        director: "Christopher Nolan",
        genre: "Action, Crime, Drama"
    },
    {
        ...
    },
];
// --
// Functions
// --

/**
 * Function to load next page
 * @return void
 */
function nextPage() {
    // Increase "currentPage"
    currentPage = currentPage + 1;
    // Redraw
    render();
}

/**
```

```
 * Function to load previous page
 * @return void
 */
function backPage() {
    // Decrease "currentPage"
    currentPage = currentPage - 1;
    // Redraw
    render();
}

/**
 * Function to return data from the wanted page
 * @param {Int) page - page number
 * @return {Array<JSON>}
 */
function getDataSlice(page) {
    const startSlice = (page - 1) * elementsPerPage;
    const endSlice = startSlice + elementsPerPage;
    return data.slice(startSlice, endSlice);
}

/**
 * Function to return total number of available pages
 * @return {Int}
 */
function getTotalPages() {
    return Math.ceil(data.length / elementsPerPage);
}

/**
 * Function to manage the paging buttons. When first or last pages are
 * active, buttons are disabled. Otherwise, they're enabled.
 * @return void
 */
function manageButtons() {
    // Check if back button is active or not
    if (currentPage === 1) {
        backButtonDOM.setAttribute("disabled", true);
    } else {
        backButtonDOM.removeAttribute("disabled");
    }
    // Check if next button is active or not
    if (currentPage === getTotalPages()) {
        nextButtonDOM.setAttribute("disabled", true);
    } else {
        nextButtonDOM.removeAttribute("disabled");
    }
}

/**
 * Function to draw the new DOM from the variables
 * @return void
 */
```

```
function render() {
    // Clean previous DOM movies
    movieListDOM.innerHTML = "";
    // Get paged movies
    const dataSlice = getDataSlice(currentPage);
    //// Draw
    // Disable appropriate buttons (previous or next page)
    manageButtons();
    // Report current page and available pages
    pageInfoDOM.innerHTML = ` ${currentPage} / ${getTotalPages()} `;
    // Create an article for each element in the current page
    dataSlice.forEach(function (movieData) {
        // Clone the movie template
        const myMovie = movieTemplate.cloneNode(true);
        // Fill data of new movie
        const myTitle = myMovie.querySelector("#title");
        myTitle.textContent = movieData.title;
        const myYear = myMovie.querySelector("#year");
        myYear.textContent = movieData.year;
        const myDirector = myMovie.querySelector("#director");
        myDirector.textContent = "Directed by " + movieData.director;
        const myGenre = myMovie.querySelector("#genre");
        const genreArray = movieData.genre.split(', ');
        for (let g of genreArray) {
            let newItem = document.createElement("li");
            newItem.textContent = g;
            myGenre.appendChild(newItem);
        }
        // Append it in "movieListDOM"
        movieListDOM.appendChild(myMovie);
    });
}

// --
// Events
// --
backButtonDOM.addEventListener("click", backPage);
nextButtonDOM.addEventListener("click", nextPage);

// --
// Start
// --
render(); // Display the first page as soon as the page is loaded
```