

TO-DO LIST

Objectives: Data in Angular components, Arrays, Events, Forms

Task: Create an Angular application that displays to-dos and allows the user to add new ones.

Phase 1: Display the data

For phase one, display the list of todos, including the buttons. But the buttons will not do anything yet. User actions will be added in phase two.

- Include a Todo Interface, which consists of a "task" (string), a "completed" (boolean), and a "duration" (number).
- The component must add an array of sample todo objects to the view. Each todo object should match the Todo interface. Include a mix of complete and incomplete items. (Pre-populate the array in the TypeScript)
- Loop through the array to display the Todos.
- Display a "complete" button only if the task is NOT completed.
- Display an "X" button for removing todos.
- Apply a "completed" CSS class to a todo if completed is true. Use a CSS rule to apply **text-decoration: line-through** to these elements.
- At least some CSS is required, but not all. If you have time, try to match the suggested styling. See visual design and style guide below.

Extended Challenges

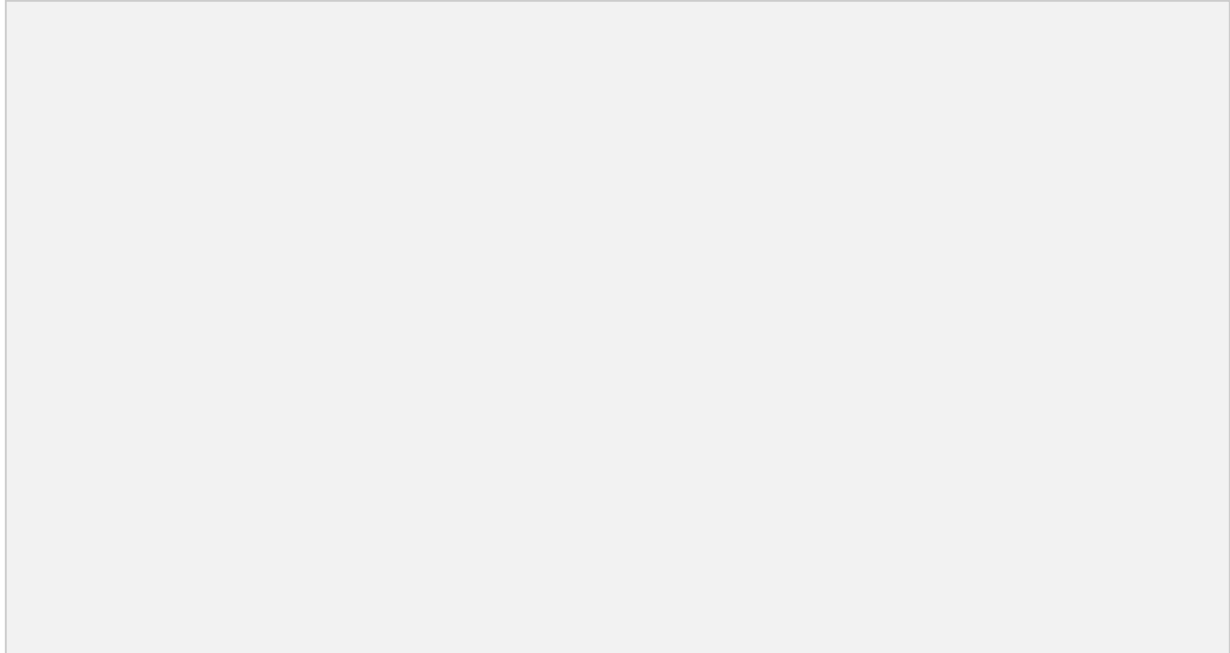
1. Add a "priority" property to the interface. It can be "NORMAL", "HIGH", or "LOW". Display the priority of each item unless it is "NORMAL".
2. Use the [DecimalPipe](#) (`| number`) to control number formatting of duration.
3. See if you can get the "Complete" and "X" buttons working.






Visual Design (next page):

If you have time, try to match these styles, but if not just get as close as you can.

NOTE: The high priority and low priority are for an extended challenge. Use the [↑](#) and [↓](#) [HTML entities](#) for the up and down arrows.





- Font-family: Open Sans;
- Font-weight: 300;
- Body background: linear-gradient(to right, #f857a6, #ff5858)
- Box-shadow: box-shadow: 0px 0px 5px 0px rgba(0,0,0,0.75);
- Colors:
 - Complete button text and border: #ff69b4 
 - Submit button and input borders: #eee 
 - Submit button background: #ff1493 
 - Submit button text: white
 - Other text is black
 - Box background is white
 - "High Priority": Crimson 
 - "Low Priority": DarkGoldenrod 



Phase 2: Add, complete, and remove todos

For phase two, implement the buttons and create a form for adding todos.

- Clicking the “complete” button executes a function named `completeTask` that sets the task’s `completed` property to `true`.
- Allow the user to remove items from the list by clicking the “X”.
- Submitting the form adds a new todo to the array with the task set to the contents of the input fields and the `complete` property set to `false`. Include an input field for the task and for the duration.

Extended Challenges

1. Include radio buttons to select priority.
2. Display a congratulations message if the list is empty or all items are complete.
3. Allow the user to filter the list by typing in the top input. (Only items which include the text typed will be shown.)
4. Allow the user to edit a to-do.

Visual Design (next page):

If you have time, try to match these styles, but if not just get as close as you can.

NOTE: The priority level radio buttons are an extended challenge.

NOTE: The text input at the top of the screenshot is for an extended challenge.



