## Overview

The final activity for today will be to build a working To-do App. You will build three different features into your app:

1. The ability to add tasks to your HTML
2. The ability to mark tasks completed and alter the look of your element
3. The ability to delete tasks off your list

## Instructions

1. Open index.html and style.css from your .zip file.
2. Take some time to go through the HTML/CSS. In this activity, you will be manipulating those elements with jQuery.

**Step 1: Create Tasks**

1. Open the index.js file from your .zip file.
2. Create a click function that targets the id of enter.
   * If you need a reminder of how to create a click function, see the skeleton of our event listener below:   
     <https://www.w3schools.com/jquery/jquery_events.asp>

| $("your selector here").on("event listener type", function(){  // Write code to run here  }); |
| --- |

1. Inside your newly created event listener, create a var named task.
   * Create a jQuery selector that targets the id of todoItem: $("your selector here").
   * Add the jQuery function .val(); at the end of your jQuery selector.
2. Beneath your newly created var, create another jQuery selector that targets the id of todoList.
   * At the end of your newly created jQuery selector, add the jQuery function .append();
     + Inside your brackets of .append, type out the following:
       - "<div class='task'>"
       - +
       - Task
         * **Note:** Task is the variable we set earlier that we are concatenating in this string.
       - +
       - "<div class='x fas fa-times'></div></div>"



1. Now, open the index.html page and try adding a task to your list using the form.

**Step 2: Delete Tasks**

1. Create a jQuery selector that targets the document.
2. Attach a click event listener to it with three parameters: the type of function, the element we are selecting, and an anonymous function.
   * **Note:** This is something we have not done yet in this course—see the code example and documentation below this note for how to accomplish this.
   * The first parameter is your “click” function, followed by the element you are selecting, “x,” and followed by an anonymous function.  
     <https://api.jquery.com/on/#direct-and-delegated-events>

| $(document).on("click", ".x", function(){ }); |
| --- |

* + Appended elements need this special type of event listener. Event listeners bind on page load. If an element is appended or deleted, it is unbound from its event listener, forcing you to target a parent container with your event listener.

1. Inside your newly created event listener, add a jQuery selector that targets this.
   * Add two jquery functions to the end of your jquery selector:
     + .parent()  
       <https://api.jquery.com/parent>/
     + .remove() with a semicolon at the end  
       <https://api.jquery.com/remove/>
2. That’s it! Reload your page, add two or three tasks to your list, and try clicking the X button on any of them.

**Step 3: Complete Tasks**

1. Create another jQuery selector that targets the document.
2. Attach a click event listener to it with three parameters—a click, the class of task, and an anonymous function.
   * **Hint:** This is the same syntax as the last event listener, except we are targeting a different class.
3. Inside your newly created event listener, create a jQuery selector that targets this.
   * Attach the jQuery function toggleClass() with a value of done  
     <https://api.jquery.com/toggleClass/>.
     + Reminder: Don’t forget your semicolon!
4. Under our new selector, create an if-else statement.

| if () {   }  else {  } |
| --- |

1. Inside your if statement’s brackets, create a jQuery selector that targets this.
   * Add the jQuery function .hasClass(); with a value of done .
2. Inside your if statement’s curly brackets, create a jQuery selector that targets this.
   * Add the jQuery function .find(); with a value of div to the end of our jQuery selector.  
     <https://api.jquery.com/find/>
   * Add the jQuery function removeClass(); with a value of fa-times to the end of .find.  
     <https://api.jquery.com/removeClass/>
3. Under your newly created selector, create another jQuery selector that targets this.
   * Add the jQuery function .find(); with a value of div to the end of your jQuery selector.
   * Add the jQuery function addClass(); with a value of fa-check to the end of .find().
     + Note: These classes toggle Font Awesome’s free icons. These icons are created with HTML elements and classes. Check it out! <https://fontawesome.com/start>
4. Inside your else statement curly brackets, create a jQuery selector that targets this.
   * Add the jQuery function .find(); with a value of div to the end of our jQuery selector.
   * Add the jQuery function addClass(); with a value of fa-times to the end of .find().  
     <https://api.jquery.com/addClass/>
5. That’s it! Reload your page, append an element to your to-do list, and click on one of them to mark it as done!