

Hands-On Project 3-1

In this project you create an application that calculates the total cost of items selected from a lunch menu using a `for` loop and an `if` statement as part of the program code. The cost of each menu item is stored in the `value` attribute of an input control on a web form. Because attribute values are treated as text strings, you will have to convert the attribute value to a number using the following JavaScript function:

```
Number(object.value)
```

where *object* is a reference to an input box within the web page. Your application will automatically update the total order cost whenever the user clicks a menu item checkbox. **Figure 3-18** shows a preview of the completed project.



The screenshot shows a web application interface. At the top, there is a blue header with the text "Hands-on Project 3-1". Below the header, the page has a yellow background. On the left side, under the heading "Lunch selections", there is a list of five items, each with a checkbox and a price in parentheses: "Fried chicken (\$11.95)", "Fried halibut (\$13.95)", "Hamburger (\$10.95)", "Grilled salmon (\$17.95)", and "Side salad (\$8.95)". The checkboxes for "Fried halibut" and "Side salad" are checked. On the right side, the text "Total Order Cost: \$22.90" is displayed.

Lunch selections	Total Order Cost: \$22.90
<input type="checkbox"/> Fried chicken (\$11.95)	
<input checked="" type="checkbox"/> Fried halibut (\$13.95)	
<input type="checkbox"/> Hamburger (\$10.95)	
<input type="checkbox"/> Grilled salmon (\$17.95)	
<input checked="" type="checkbox"/> Side salad (\$8.95)	

Figure S-18 Completed Project 3-1

Do the following:

1. Use your code editor to open the **project03-01_txt.html** and **project03-01_txt.js** files from the js03 C project01 folder. Enter your name and the date in the comment section of each file and save them as **project03-01.html** and **project03-01.js**, respectively.

2. Go to the **project03-01.html** file in your code editor and in the head section add a `script` element to load the `project03-01.js` file. Include the `defer` attribute to defer loading the file until the entire page is loaded. Study the contents of the HTML file, noting that all checkboxes for the menu items belong to the `menuItems` class. Save your changes to the file.
3. Go to the **project03-01.js** file in your code editor. Below the initial comment section, declare a variable named **menuItems** containing the collection of HTML elements belonging to the `menuItem` class using the `getElementsByClassName()` method.
4. Create a `for` loop that loops through the contents of the `menuItems` collection with a counter variable that starts with an initial value of 0 up to a value less than the length of the `menuItems` collection. Increase the counter by 1 with each iteration. Within the `for` loop, add an event listener to the `menuItems[i]` element in the collection (where `i` is the value of the counter), running the `calcTotal()` function when that item is clicked.
5. Create the **calcTotal()** function to calculate the total cost of the customer order given the selected menu items. Add the following commands to the function:
 - a. Declare the **orderTotal** variable, setting its initial value to 0.
 - b. Create a `for` loop that loops through the contents of the `menuItems` collection. For `menuItems[i]` (where `i` is the counter), apply an `if` statement that tests whether the item has been checked. If true, increase the value of the `orderTotal` variable by the value of `menuItems[i]`. (Hint: Use the `Number()` function to convert the value of `menuItems[i]` to a number.)
 - c. After the `for` loop, insert a command to change the `innerHTML` property of the element with the id "billTotal" to the value returned by the `formatCurrency()` function using `orderTotal` as the parameter value.
6. Save your changes to the file and then open **project03-01.html** in your browser. Verify that the total cost of the order is automatically updated as you select and deselect menu items in the web form.

Hands-On Project 3-2

In this project you will generate an image gallery using images of the International Space Station. Each image will be placed within a figure box accompanied by a caption taken from an array of caption text. A preview of the completed project is shown in **Figure 3-19**.

Do the following:

1. Use your code editor to open the **project03-02_txt.html** and **project03-02_txt.js** files from the `js03 C project02` folder. Enter your name and the date in the comment section of each file and save them as **project03-02.html** and **project03-02.js**, respectively.
2. Go to the **project03-02.html** file in your code editor and in the head section add a `script` element to load the `project03-02.js` file, deferring the loading of the JavaScript source file until the entire HTML file is loaded. Study the contents of the HTML file and save your changes.
3. Go to the **project03-02.js** file in your code editor. Below the code that creates and populates the `captions` array, declare the **htmlCode** variable, setting its initial value to an empty text string.
4. Create a `for` loop with a counter that goes from 0 to less the length of the `captions` array in increments of 1. With each iteration, add the following text to the value of the `htmlCode` variable:

```
<figure>
<img alt='' src='slidei.jpg' />
<figcaption>caption[i]</figcaption>
</figure>
```

where `i` is the value of the counter for that iteration and `captions[i]` is the corresponding element from the `captions` array.

Hands-on Project 3-2

International Space Station Images

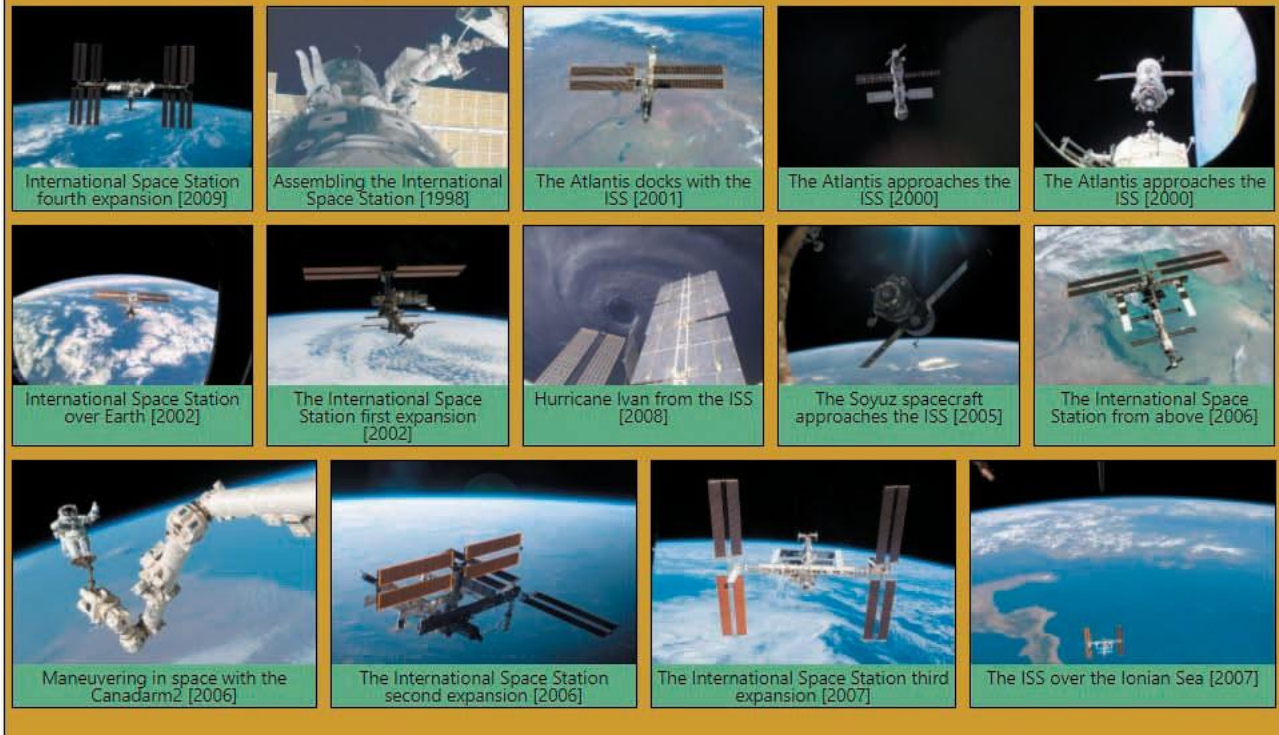


Figure S-19 Completed Project 3-2 NASA

5. After the `for` loop, change the inner HTML of the document element by the id "gallery" to the value of the `htmlCode` variable.
6. Save your changes to the file and then load **project03-02.html** in your browser. Verify that the page displays the 14 images in the slide gallery along with their captions.