Lab 9: Core JavaScript

Due date: Oct 21, 2023, 4:00 am

Overview

In this lab, you will implement a script that formats form input for display.

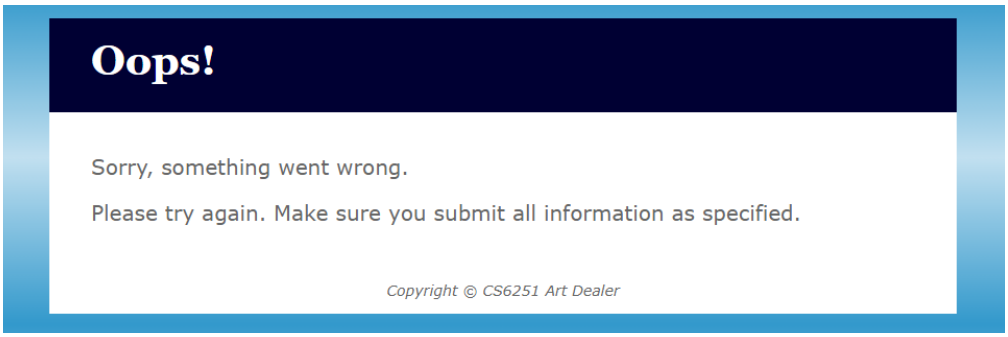
The Strater Project

Download the starter project from the course website. The project consists of three files:

1. An HTML file that implements a form.
    - The form contains a hidden <input> element with name and id `summary`. Its value is set to an empty string.
    - When the form is submitted, the JavaScript function `prepareSummary` is called. A stub of the function is implemented in the JavaScript file. However, the body of the function only returns true. You will complete the function and its helper functions in this lab.
    - The form is submitted to an external script. If the value of the hidden <input> element is an empty string, then the resulting page displays an error message and asks the user to submit the form again. If the value of the hidden <input> element is not empty, the external script displays the value on the resulting page.

    You will not have to make changes to the HTML file.
2. A style sheet, which is a stripped-down version from the pacific trails resort case study. You will not have to make changes to the style sheet.
3. A JavaScript file with three function stubs. You will have to implement the body of these three functions in this lab.

The given function `prepareSummary` in the starter project sets the hidden form field to an empty string, no matter which form input is submitted by the user. If the hidden form field value is the empty string, the external PHP script will display the following page upon form submission:



Instructions

Complete the three functions as specified in the subsections below and as specified in the JSDoc comments:

- `prepareSummary()`
- `validateAndParseArtworkData(artwork)`
- `formatSummary(artwork)`

If you need help with this lab, check out the example code in lab9_example.zip first.

Function `prepareSummary`

Implement the body of the function as follows:

1. Retrieve the values entered by the user in the form fields and create an object with the corresponding properties:
   - `title`
   - `artist`
   - `year`
   - `price`
   - `description`

   Store each form field value as a string in the corresponding property. Thus, all properties are of type string initially.
2. Call function `validateAndParseArtworkData` where the newly created object is passed in.
3. If the function `validateAndParseArtworkData` returns `false` (meaning that the user input is invalid), the value of the hidden <input> element is set to an empty string.
4. If the function `validateAndParseArtworkData` returns `true` (meaning that the user input is valid and the `year` and `price` properties are set to the number value they originally represented as string), the function `formatSummary` is called. The value of the hidden form field is set to the return value of function `formatSummary`.
5. The function should return true in all scenarios.

Function `validateAndParseArtworkData`

Implement the body of the function as follows:

1. Validate the properties of the passed-in object artwork. As specified in the JSDoc comment for the function, the properties have to meet the following criteria:
   - `artwork.title`: must be a non-empty string
   - `artwork.artist`: must be a non-empty string

- `artwork.year`: must be an empty string or represent a non-negative integer. If `artwork.year` can be parsed to an integer using `Number.parseInt`, it can be considered to be an integer. A *non-negative* value is 0 or greater than 0.
- `artwork.price`: must be a non-negative value (If `artwork.price` can be parsed to a float using method `Number.parseFloat`, it can be considered to be a value.)
- The property `artwork.description` does not need to meet any criteria.

2. If one or more of the above criteria are violated, the function `validateAndParseArtworkData` should return `false`.
3. If all criteria are met, then `validateAndParseArtworkData` should do the following:
   a. The property `artwork.year` should be set to `NaN` if the original property value is an empty string; otherwise, the property `artwork.year` should be set to the integer number it represents. Thus, when `validateAndParseArtworkData` returns, the property `artwork.year` has changed its type from string to number.
   b. The property `artwork.price` should be set to the number it represents. Thus, when `validateAndParseArtworkData` returns, the property `artwork.price` has changed its type from string to number.
   c. The function should return `true`.

Function `formatSummary`

Function `formatSummary` creates and returns a formatted string from the form input for display in an HTML document. As specified in the JSDoc comment for the function, the formatted string has to be of the following form:

{artwork.title}
by {artwork.artist}
[Year: {artwork.year}
]Price: ${artwork.price}
[Description: {artwork.description}]

where

- {artwork.title} is the value of `artwork.title` in uppercase letters
- {artwork.artist} is the name of the artist.

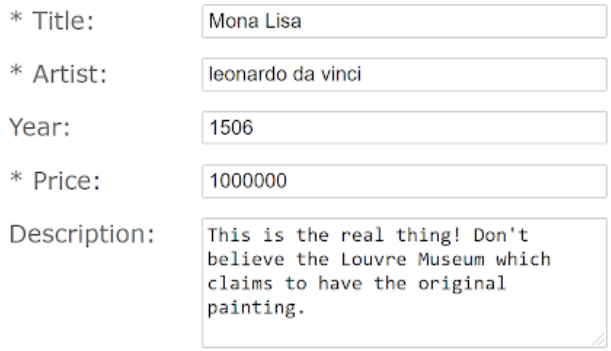
  Optional task for bonus point: Format the artist name such that the first letter of the first word is capitalized and all other letters are lowercase.
- {artwork.price} is the value of `artwork.price` with two decimal digits after the decimal point.
- {artwork.description} is the value of `artwork.description`

  Optional task for bonus point: If the string `artwork.description` has more than 100 characters, then {artwork.description} should consist of the first 97 characters of `artwork.description` followed by three dots.

The brackets [] around “Year: {artwork.year}
” and around “Description: {artwork.description}” should not be included in the formatted string. The brackets indicate that the part is not necessarily included. In particular, the part [Year: {artwork.year}
] should be omitted in case `artwork.year` is not a number (NaN). The part [Description: {artwork.description}] should be omitted if

`artwork.description` is an empty string. The following pages list several examples that demonstrate the formatted string for different input scenarios. The demonstrations include the optional formatting of the artist name and the artwork description.

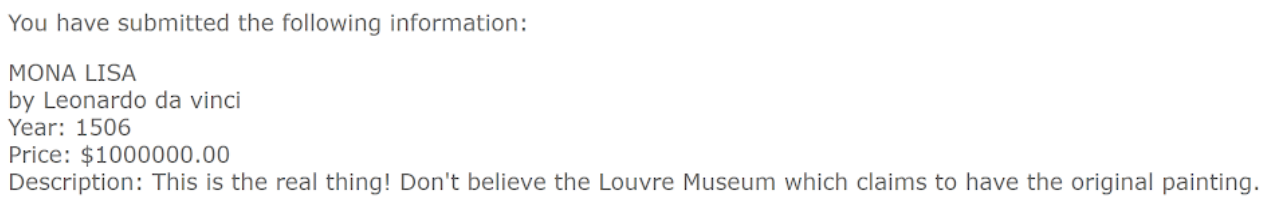
Example 1: All fields contain correct input



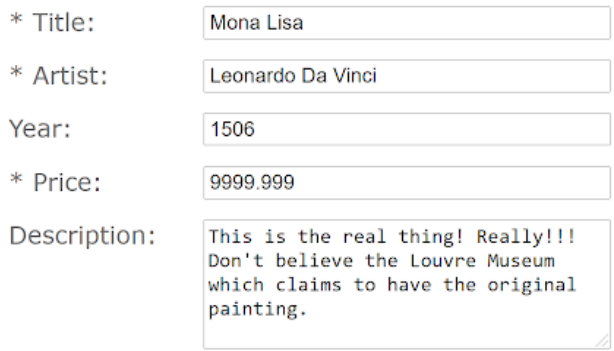
String returned by `formatSummary`:

```
MONA LISA<br>by Leonardo da vinci<br>Year: 1506<br>Price: $1000000.00<br>Description: This is the real thing! Don't believe the Louvre Museum which claims to have the original painting.
```

Resulting page content:



Example 2: Price rounded to two decimal digits, description over 100 characters



String returned by `formatSummary`:

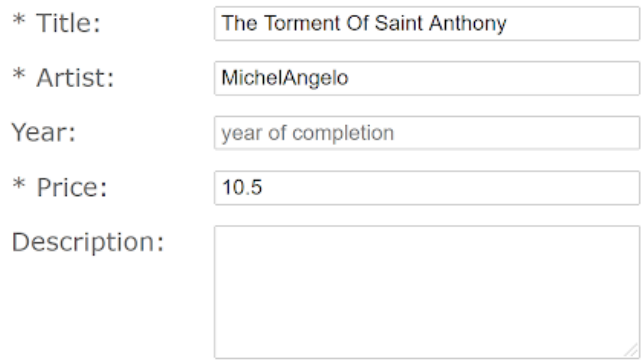
```
MONA LISA<br>by Leonardo da vinci<br>Year: 1506<br>Price: $10000.00<br>Description: This is the real thing! Really!!! Don't believe the Louvre Museum which claims to have the origin…
```

Resulting page content:

You have submitted the following information:

MONA LISA
by Leonardo da vinci
Year: 1506
Price: $10000.00
Description: This is the real thing! Really!!! Don't believe the Louvre Museum which claims to have the origin...

Example 3: No year and no description



String returned by `formatSummary`:

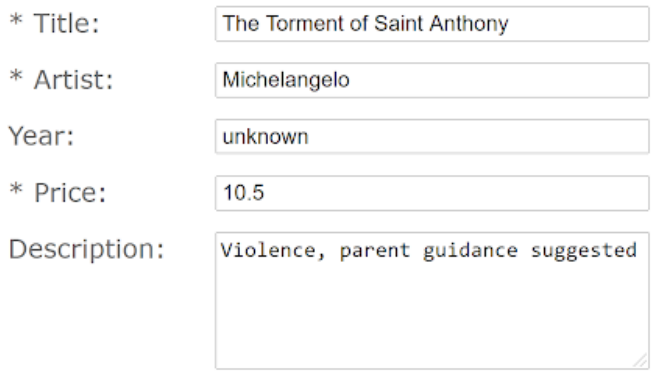
```
THE TORMENT OF SAINT ANTHONY<br>by Michelangelo<br>Price: $10.50<br>
```

Resulting page content:

You have submitted the following information:

THE TORMENT OF SAINT ANTHONY
by Michelangelo
Price: $10.50

Example 4: Invalid year



Function `formatSummary` should not be called. Due to the invalid form field, the hidden form field value is set to an empty string.

Resulting page content:

> Sorry, something went wrong.
>
> Please try again. Make sure you submit all information as specified.

Submission

Archive the folder that contains all files with your solution using the ZIP format and submit your solution on the course website before the due date. Make sure that the name of the submitted zip file contains your name.

Grading

- Function `prepareSummary`: 3 points
- Function `validateAndParseArtworkData`: 4 points
- Function `formatSummary`: 3 points
- Bonus points: 2 points
  - Optional formatting of artist name
  - Optional formatting of artwork description

  No help will be provided for the optional tasks.

I recommend using Git for your own convenience, but using a repository is not required.