Simple Interest Calculator Using HTML, CSS and Javascript

Objective:

1. Download the project folder.
2. Modify the HTML file as per the requirements.
3. Modify the CSS file as per the requirements.
4. Modify the JavaScript file as per the requirements.
5. Verify that the webpage is working properly.

Exercise 1: Modify the HTML file

In this exercise, you will correct any mistakes in the existing code and also add all missing tags.

1. In the file explorer, navigate to the index.html file.
2. All HTML files must begin with a doctype tag, to indicate that HTML content will be placed in the file. Add this tag to the beginning of index.html.
3. Use the title tag to change the browser title to be "Simple Interest Calculator". Remember that the title tag should be placed in the head section of your markup.
4. Move all the content which is currently in the <body> to a new <div> tag.
5. Set the class attribute of this new div to maindiv.
6. Modify the input id="rate" tag for the interest rate to be a slider. Recall from earlier lessons that this can be done by changing the type to range.
7. For the rate input, add the following attributes and their corresponding values:
   - min should be set to 1
   - max should be set to 20
   - value should be set to 10.25
   - step should be set to 0.25

Range is an elegant way to input numeric input, but the drawback is that it does not visually show the value the user has selected.

8. To show the value selected by the range, create a <span> element right after the range, with the id rate_val.
9. Inside the <span> tag, add the text "10.25" to represent the default value (as specified above). Add a "%" outside this span tag. The span will be updated dynamically later on, but the "%" should always remain, so this is placed outside the tag.

Insert a line break after this tag, so the next input appears on a new line.

10. Modify the input text box for "No. of Years" into a dropdown box with options 1 to 10.
11. Change the name of the "Compute" button to "Compute Interest".

12. Below the "Compute Interest" button, create an empty <span> and set its id to result. This will be used to dynamically display the result of the calculation when the "Compute Interest" button is clicked.

This will be used to display the output of the user's calculation.

13. Outside the maindiv, add a copyright message using the <footer> tag
14. Open your application using the LocalServer and make sure that you have not missed anything. Your page should look similar to this:

Simple Interest Calculator

Amount [   ]
Rate ———●——————— 10.25 %
No. of Years [   ]
Interest :
[Compute Interest]
© This Calculator belongs to --your name--

Exercise 2: Modify the CSS file

In this exercise, you will correct the look and feel of the web page.

1. On the file explorer navigate to the style.css style sheet.

2. Set the body background color to 'black', font family to 'arial' and font color to 'white'.

3. Set the h1 color to 'grey' and font to 'verdana'.

4. Create an entry for class 'maindiv'.

5. In the newly created maindiv class, set the following styles:
   - Background color to 'white'
   - Font color to 'black'
   - Width to '300px'
   - Padding to '20px'
   - Border radius to '25px'
   - Text alignment to 'center'

6. Save the changes made in style.css.

7. Open your application using the LocalServer and make sure that you have not missed anything. Your page should look similar to below:

Exercise 3: Modify the JavaScript file

In this exercise, you will write the JavaScript code in the script.js file, to implement the logic for the Simple Interest Calculator.

Display Rate Slider Value

1. Create an empty function called updateRate(). This will be used to display the value of the 'Rate' slider.
2. Inside the updateRate() function, create a variable rateval that gets the *value* from the 'Rate' slider.

Hint: the Rate slider is the element with an id rate

3. Modify the `<span id="rate_val">` value to display the value of the rateval variable created above.
4. Link this function with an "onchange" event on the range input.
5. Save the file and open your web page with the Live Server extension. Change the slider and verify that the value to the right of it updates with a new value each time the slider is changed.

If this does not work as expected, go back to your code to identify where the problem is.

Compute Button Functionality

1. Create the following variables inside the compute() function, and assign them to the corresponding value listed:
   - principal initialized to the value of the input element with an id of principal, parsed as an int. This is needed to calculate the final amount, as well as the interest amount
   - rate initialized to the value of the input element with an id of rate, parsed as a float. This is needed to calculate the interest amount

- **years** initialized to the value of the input element with an id of **years**, parsed as an int. This is needed to calculate the interest amount
- **interest** with the value **principal * num_years * rate / 100**. This is needed to calculate the total amount
- **amount** which is the sum of the integer value of **principal and the float value of interest**
- **result** initialized to the input element with an id of **result**. This is needed to modify the text when the Compute button is pressed

2. Write the logic to convert the 'No. of Years' into the actual year in the future. This can be done by adding the number of years (**years**) to the current year inside the compute() function.

This will ensure that the input taken as "No. of Years" is converted into an actual year (e.g. 2022).

3. Add validation for the "Principal" input box. If the user enters zero or a negative value, show an alert which says "Enter a positive number"
4. Once the user clicks on the alert "OK" button, take the user back to the "Principal" input box, by setting the focus on this box using the **focus** method in the code for principal input validation:
5. Within an **else** clause, set the inner html property of the result to the text below, replacing anything within the square brackets [] with its actual value.

Note that when writing **<** or **>** within quotation marks, you must instead type **₩<** or **₩>**

6. Make sure the numbers in the result are highlighted by using the **mark** HTML tag around each variable value:

Exercise 4: Test the Calculator

Now that you have finished coding your calculator, you must do some basic testing.

Write comments in your code which will help debug and maintain the code in the long term.

1. Enter these values in the form.
   - **Amount** = **0**
   - **Rate** = **1%**
   - **No. of Years** = **1**

Click on the **Compute Interest** button.

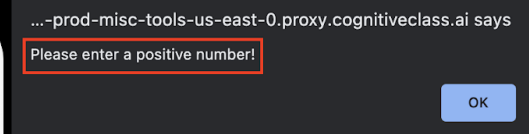The output should be an alert saying **Enter a positive number**.

© This Calculator belongs to --your name--

2. Enter these values in the form.
   - **Amount** = **1000**
   - **Rate** = **10%**
   - **No. of Years** = **10**

Click on the Compute button.

The output should be:

Simple Interest Calculator

Amount `1000`

Rate ●━━━━━━━━━━ 10%

No. of Years `10`

Interest : If you deposit $`1000`,
at an interest rate of `10%`
You will receive an amount of $`2000`,
in the year `2032`

Compute Interest

© This Calculator belongs to --your name--

3. **Enter these values in the form.**

   - **Amount = 4800**
   - **Rate = 15.25%**
   - **No. of Years = 5**

Click on the Compute button.

You should see the following output:

Simple Interest Calculator

Amount `4800`

Rate ▬▬▬▬●▬▬ **15.25%**

No. of Years `5`

Interest : If you deposit $**4800**,
at an interest rate of **15.25%**
You will receive an amount of $**8460**,
in the year **2027**

Compute Interest

© This Calculator belongs to --your name--

The year is 2022 at the time this lab was written, 2027 is therefore correct. If you are doing this in a different year, make sure the year value is the current year + 5 years.