**HTML**

# **Modify the HTML file**

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

In the file explorer, navigate to the index.html file.

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.

Click here to see how it should look

1. <!DOCTYPE html>

Copied!

1. 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.

Click here to see the code

1. <title>Simple Interest Calculator</title>

Copied!

Move all the content which is currently in the <body> to a new <div> tag.

Set the class attribute of this new div to maindiv.

Click here to see the code

1. <body>
2. <div class="maindiv">
3. <h1>Simple Interest Calculator</h1>
4. Amount <input type="number" id="principal"> <br/>
5. Rate <input type="number" id="rate"> <br/>
6. No. of Years <input type="number" id="years"> <br/>
7. Interest : <span id="result"></span><br>
8. <button onclick="compute()">Compute</button>
9. </div>
10. </body>

Copied!

1. 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****.
2. <input type="range" id="rate">

Copied!

1. 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****

Click here to see the code for the interest rate tag after making the above updates

1. Rate <input type="range" id="rate" min="1" max="20" value="10.25" step="0.25">

Copied!

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

1. To show the value selected by the range, create a <span> element right after the range, with the id rate\_val.

Click here to see the code

1. <span id="rate\_val">
2. </span>

Copied!

1. 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.

Click here to see the code

1. <span id="rate\_val">
2. 10.25
3. </span>% <br/>
4. Modify the input text box for “No. of Years” into a dropdown box with options 1 to 10.

*Recall from the “HTML5 Input Element” video, the correct way to insert a dropdown in HTML*

Click here to see how to insert a dropdown

1. 1
2. No. of Years
3. <input type="number" id="years" list="all\_years">
4. <datalist id="all\_years">
5. <option value="1">1</option>
6. <option value="2">2</option>
7. <!-- Fill in the rest of the values -->
8. </datalist>

Copied!

1. Change the name of “Compute” button to “Compute Interest”.
2. <button onclick="compute()">Compute Interest</button>

Copied!

1. 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.

Click here to see the code

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

1. Outside the maindiv, add a copyright message using the <footer> tag, like below:
2. <footer>
3. &#169; This Calculator belongs to --your name--
4. </footer>

Copied!

1. Save the changes made in index.html.

Click to see the updated code in index.html

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <script src="script.js"></script>
5. <link rel="stylesheet" href="style.css">
6. <title>Simple Interest Calculator</title>
7. </head>
8. <body>
9. <div class="maindiv">
10. <h1>Simple Interest Calculator</h1>
11. Amount <input type="number" id="principal"> <br/>
12. Rate <input type="range" id="rate" min="1" max="20" value="10.25" step="0.25">
13. <span id="rate\_val">
14. 10.25
15. </span>% <br/>
16. No. of Years
17. <input type="number" id="years" list="all\_years">
18. <datalist id="all\_years">
19. <option value="1">1</option>
20. <option value="2">2</option>
21. <option value="3">3</option>
22. <option value="4">4</option>
23. <option value="5">5</option>
24. <option value="6">6</option>
25. <option value="7">7</option>
26. <option value="8">8</option>
27. <option value="9">9</option>
28. <option value="10">10</option>
29. </datalist>
30. <br />
31. Interest : <span id="result"></span><br/>
32. <button onclick="compute()">Compute Interest</button>
33. </div>
34. <footer>
35. &#169; This Calculator belongs to --your name--
36. </footer>
37. </body>
38. </html>

Copied!

1. Open your application using the Live Server extension and make sure that you have not missed anything. Your page should look similar to this:

**CSS**

# **Modify the CSS file**

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

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

Set the body background color to ‘black’, font family to ‘arial’ and font color to ‘white’.

Click here to see how it should look

1. body {
2. background-color: black;
3. font-family: arial;
4. color: white;
5. }

Copied!

1. Set the h1 color to ‘grey’ and font to ‘verdana’.

Click here to see how it should look

1. h1 {
2. color: grey;
3. font-family: verdana;
4. }

Copied!

1. Create an entry for class ‘maindiv’.

Click here to see how it should look

1. 3
2. .maindiv {
4. }

Copied!

1. 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’

Click to see how the new `maindiv` class should look like

1. .maindiv {
2. background-color: white;
3. color: black;
4. width: 300px;
5. padding: 20px;
6. border-radius: 25px;
7. text-align: center;
8. }

Copied!

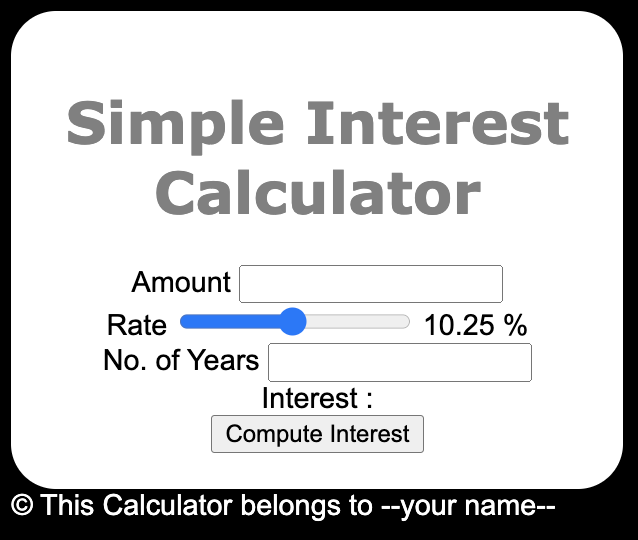
1. Save the changes made in style.css.

Click here to see the updated code in style.css

1. 19
2. body {
3. background-color: black;
4. font-family: arial;
5. color: white;
6. }
7. h1 {
8. color: grey;
9. font-family: verdana;
10. }
11. .maindiv {
12. background-color: white;
13. color: black;
14. width: 300px;
15. padding: 20px;
16. border-radius: 25px;
17. text-align: center;
18. }

Copied!

1. Open your application using the Live Server extension and make sure that you have not missed anything. Your page should look similar to below:



**JAVASCRIPT**

# **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.

Click here to see how it should look

1. function updateRate() {
2. }

Copied!

1. 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

Click here to see how it should look

1. function updateRate()
2. {
3. var rateval = document.getElementById("rate").value;
4. }

Copied!

1. Modify the <span id="rate\_val"> value to display the value of the rateval variable created above.

Click here to see how it should look

1. function updateRate()
2. {
3. var rateval = document.getElementById("rate").value;
4. document.getElementById("rate\_val").innerText = rateval;
5. }

Copied!

1. Link this function with an “onchange” event on the range input.

Click here for a hint of how to do this

Add onchange="updateRate()" inside the Rate input tag in index.html, so that it becomes:

Click here to see the code

1. 1
2. Rate <input type="range" id="rate" min="1" max="20" value="10.25" step="0.25" onchange="updateRate()">

Copied!

1. 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

Click here to see the code implementation

1. var principal = document.getElementById("principal").value;
2. var rate = document.getElementById("rate").value;
3. var years = document.getElementById("years").value;
4. var interest = principal \* years \* rate / 100;
5. var amount = parseInt(principal) + parseFloat(interest);
6. var result = document.getElementById("result");

Copied!

1. 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.

Click here to see the code

1. var year = new Date().getFullYear() + parseInt(years);

Copied!

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

1. 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”

Click here to see the code

1. if (principal <= 0) {
2. alert("Please enter a positive number!");
3. }

Copied!

1. 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:

Click here to see how it should look

1. if (principal <= 0) {
2. alert('Please enter a positive number!');
3. document.getElementById("principal").focus();
4. }

Copied!

*You can refer to the [Javascript Form Validation lab](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CD0101EN-SkillsNetwork/labs/Theia Labs/03 - Javascript/instructions.md.html" \t "https://author-ide.skills.network/_new).*

1. 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 \>

1. If you deposit $[PRINCIPAL],<br>
2. at an interest rate of [RATE]%.<br>
3. You will receive an amount of $[INTEREST],<br>
4. in the year [YEAR]<br>

Copied!

Click here to see the code

1. else {
2. result.innerHTML = "If you deposit $" + principal + ",\<br\>at an interest rate of " + rate + "%\<br\>You will receive an amount of $" + amount + ",\<br\>in the year " + year + "\<br\>";
3. }

Copied!

1. Make sure the numbers in the result are highlighted by using the mark HTML tag around each variable value:
2. else {
3. result.innerHTML = "If you deposit $" + "<mark>" + principal + "</mark>" + ",\<br\> at an interest rate of " + "<mark>" + rate + "%" + "</mark>" + "\<br\> You will receive an amount of $" + "<mark>" + amount + "</mark>" + ",\<br\> in the year " + "<mark>" + year + "</mark>" + "\<br\>";
4. }
5. The code in the updated script.js file should be:

Click here to see the updated code in script.js

1. function compute() {
2. var principal = document.getElementById("principal").value;
3. var rate = document.getElementById("rate").value;
4. var years = document.getElementById("years").value;
5. var interest = principal \* years \* rate / 100;
6. var year = new Date().getFullYear() + parseInt(years);
7. var amount = parseInt(principal) + parseFloat(interest);
8. var result = document.getElementById("result");
10. if (principal <= 0) {
11. alert('Please enter a positive number!');
12. document.getElementById("principal").focus();
13. }
14. else {
15. result.innerHTML = "If you deposit $" + "<mark>" + principal + "</mark>" + ",\<br\> at an interest rate of " + "<mark>" + rate + "%" + "</mark>" + "\<br\> You will receive an amount of $" + "<mark>" + amount + "</mark>" + ",\<br\> in the year " + "<mark>" + year + "</mark>" + "\<br\>";
16. }
17. }
18. function updateRate()
19. {
20. var rateval = document.getElementById("rate").value;
21. document.getElementById("rate\_val").innerText = rateval;
22. }