Hands-On Lab: Simple Interest Calculator

Skills Network

Estimated time: 30 mins

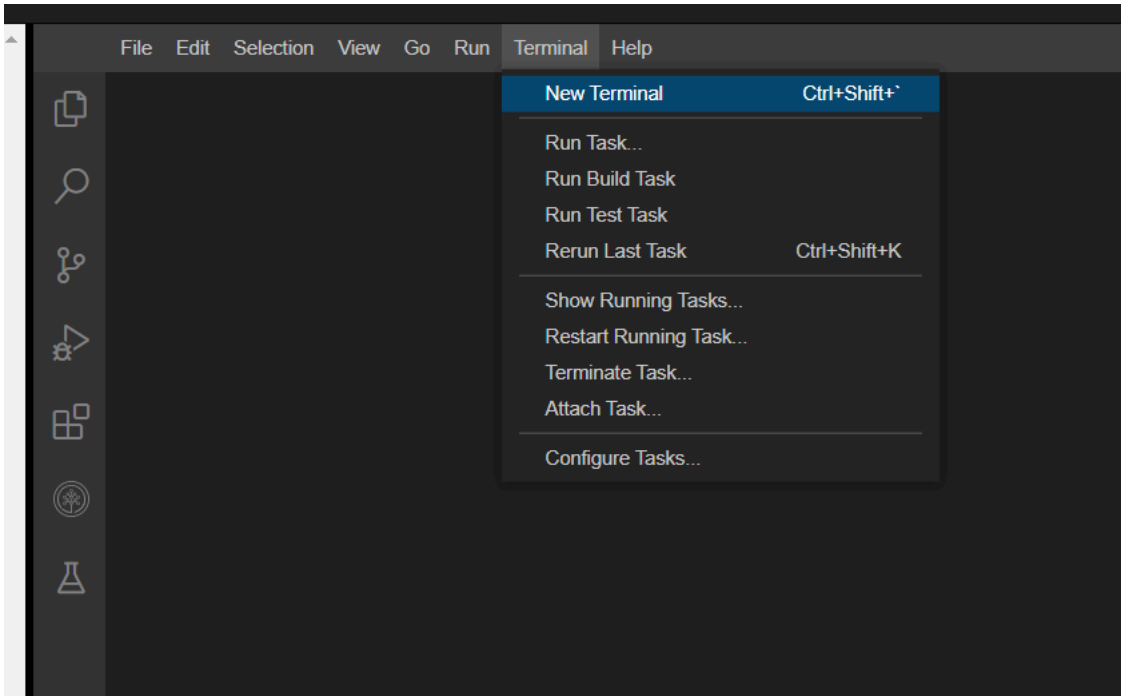
Objectives:

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.

   This lab environment does not store your changes across multiple sessions. To avoid losing your work, please complete your project in one session or store your files elsewhere and add them back here to continue at another time.

Exercise 1: Download the project folder

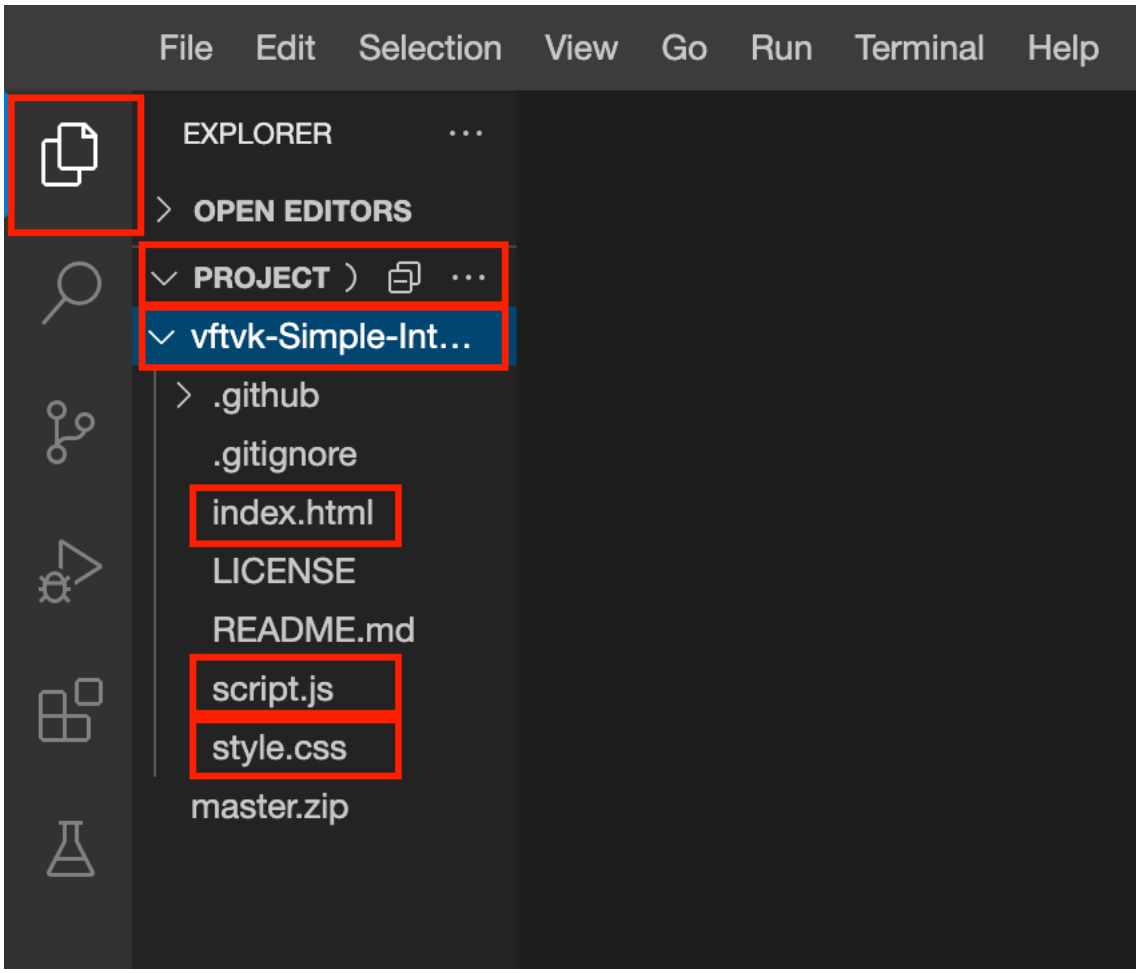
1. Open a new terminal



2. Paste the following commands into your terminal to download the project folder and unzip it

```
wget https://github.com/ibm-developer-skills-network/vftvk-Simple-Interest-Calculator/archive/refs/heads/master.zip
unzip master.zip
```

3. Open the file explorer and verify that the project contains the following files:
   - index.html
   - style.css
   - script.js



4. Throughout this project, you will need to test your web page using the Live Server extension. For steps on how to use the extension, click here. Note that opening Live Server will only work if you click on the `index.html` file, it will not appear for any other file type.

Exercise 2: 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`.

▼ Click here to see how it should look

```
<!DOCTYPE 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.

▶ Click here to see the code

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`.

▶ Click here to see the code

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**.

```
<input type="range" id="rate">
```

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**

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

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

Range is an elegant way to input numeric input, but the drawback is that it does not visually show 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`.

▼ Click here to see the code

```
<span id="rate_val">
</span>
```

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.

▼ Click here to see the code

```
<span id="rate_val">
  10.25
</span>% <br/>
```

10. 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

```
No. of Years
<input type="number" id="years" list="all_years">
<datalist id="all_years">
    <option value="1">1</option>
    <option value="2">2</option>
    <!-- Fill in the rest of the values -->
</datalist>
```

11. Change the name of "Compute" button to "Compute Interest".

▼ Click here to see the code

```
<button onclick="compute()">Compute Interest</button>
```

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.

▶ Click here to see the code

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, like below:

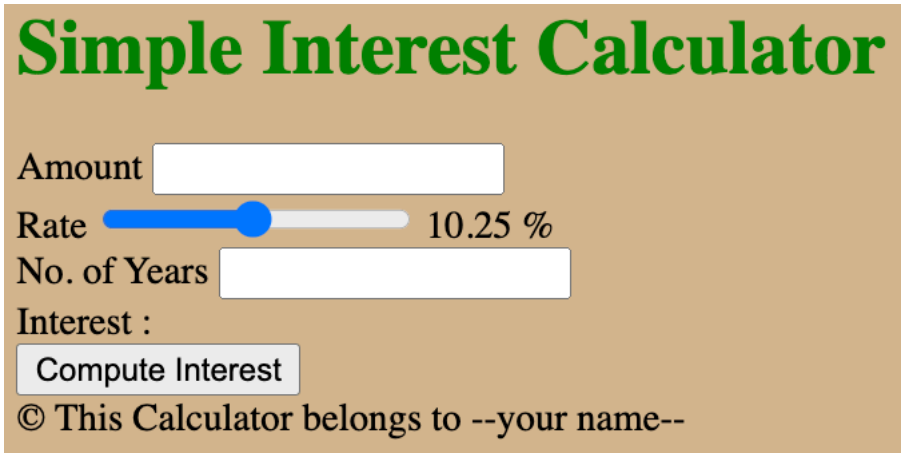
```
<footer>
    &#169; This Calculator belongs to --your name--
</footer>
```

14. Save the changes made in `index.html`.

▼ Click to see the updated code in index.html

```
<!DOCTYPE html>
<html>
    <head>
        <script src="script.js"></script>
        <link rel="stylesheet" href="style.css">
        <title>Simple Interest Calculator</title>
    </head>
    <body>
        <div class="maindiv">
            <h1>Simple Interest Calculator</h1>
            Amount <input type="number"  id="principal">  <br/>
            Rate <input type="range" id="rate" min="1" max="20" value="10.25" step="0.25">
            <span id="rate_val">
                10.25
            </span>% <br/>
            No. of Years
            <input type="number" id="years" list="all_years">
            <datalist id="all_years">
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
                <option value="4">4</option>
                <option value="5">5</option>
                <option value="6">6</option>
                <option value="7">7</option>
                <option value="8">8</option>
                <option value="9">9</option>
                <option value="10">10</option>
            </datalist>
            <br />
            Interest : <span id="result"></span><br/>
            <button onclick="compute()">Compute Interest</button>
        </div>
        <footer>
            &#169; This Calculator belongs to --your name--
        </footer>
    </body>
</html>
```

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



Exercise 3: 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'.

▼ Click here to see how it should look

```
body {
    background-color: black;
    font-family: arial;
    color: white;
}
```

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

▼ Click here to see how it should look

```
h1 {
    color: grey;
    font-family: verdana;
}
```

4. Create an entry for class 'maindiv'.

▼ Click here to see how it should look

```
.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'

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

```
.maindiv {
    background-color: white;
    color: black;
    width: 300px;
    padding: 20px;
    border-radius: 25px;
    text-align: center;
}
```

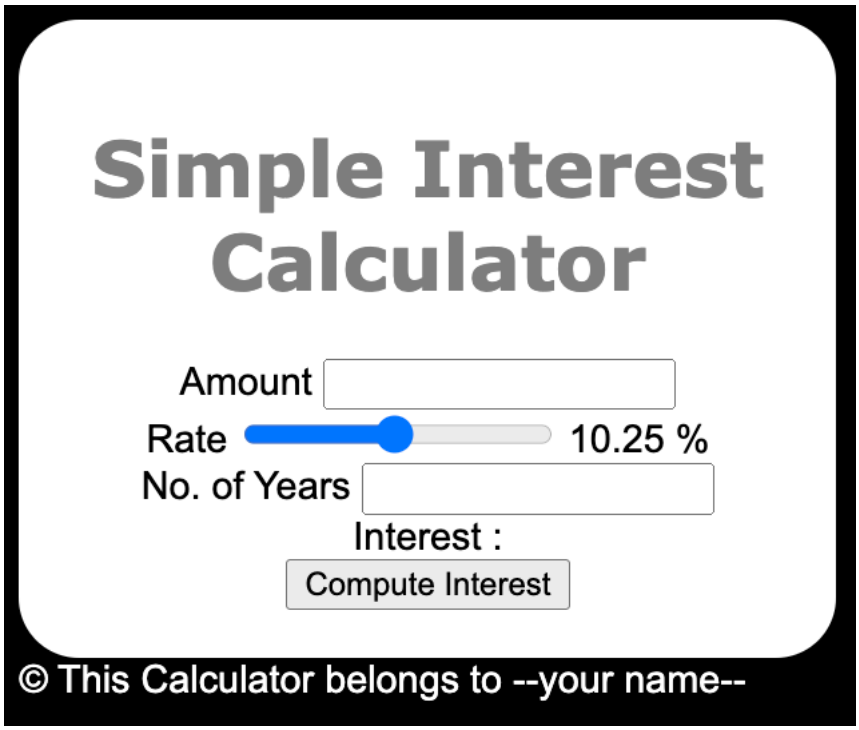
6. Save the changes made in `style.css`.

▼ Click here to see the updated code in style.css

```
body {
    background-color: black;
    font-family: arial;
    color: white;
}
h1 {
    color: grey;
    font-family: verdana;
}
```

```
.maindiv {
    background-color: white;
    color: black;
    width: 300px;
    padding: 20px;
    border-radius: 25px;
    text-align: center;
}
```

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



Exercise 4: 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

```
function updateRate() {
}
```

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`

▼ Click here to see how it should look

```
function updateRate()
{
    var rateval = document.getElementById("rate").value;
}
```

3. 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

```
function updateRate()
{
    var rateval = document.getElementById("rate").value;
    document.getElementById("rate_val").innerText = rateval;
}
```

4. 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

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

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

▼ Click here to see the code implementation

```
var principal = document.getElementById("principal").value;
var rate = document.getElementById("rate").value;
var years = document.getElementById("years").value;
var interest = principal * years * rate / 100;
var amount = parseInt(principal) + parseFloat(interest);
var result = document.getElementById("result");
```

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.

▼ Click here to see the code

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

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"

▼ Click here to see the code

```
if (principal <= 0) {
    alert("Please 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:

▼ Click here to see how it should look

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

You can refer to the Javascript Form Validation lab.

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 \>

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

▼ Click here to see the code

```
else {
    result.innerHTML = "If you deposit $" + principal + ",\<br\>at an interest rate of " + rate + "%\<br\>You will receive an amount
}
```

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

```
else {
    result.innerHTML = "If you deposit $" + "<mark>" + principal + "</mark>" + ",\<br\> at an interest rate of " + "<mark>" + rate +
}
```

7. The code in the updated `script.js` file should be:

▼ Click here to see the updated code in script.js

```
function compute() {
    var principal = document.getElementById("principal").value;
    var rate = document.getElementById("rate").value;
    var years = document.getElementById("years").value;
    var interest = principal * years * rate / 100;
    var year = new Date().getFullYear() + parseInt(years);
    var amount = parseInt(principal) + parseFloat(interest);
```

```
    var result = document.getElementById("result");

    if (principal <= 0) {
        alert('Please enter a positive number!');
        document.getElementById("principal").focus();
    }
    else {
        result.innerHTML = "If you deposit $" + "<mark>" + principal + "</mark>" + ",\<br\> at an interest rate of " + "<mark>" + ra
    }
}
function updateRate()
{
    var rateval = document.getElementById("rate").value;
    document.getElementById("rate_val").innerText = rateval;
}
```

Exercise 5: Test the Calculator

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

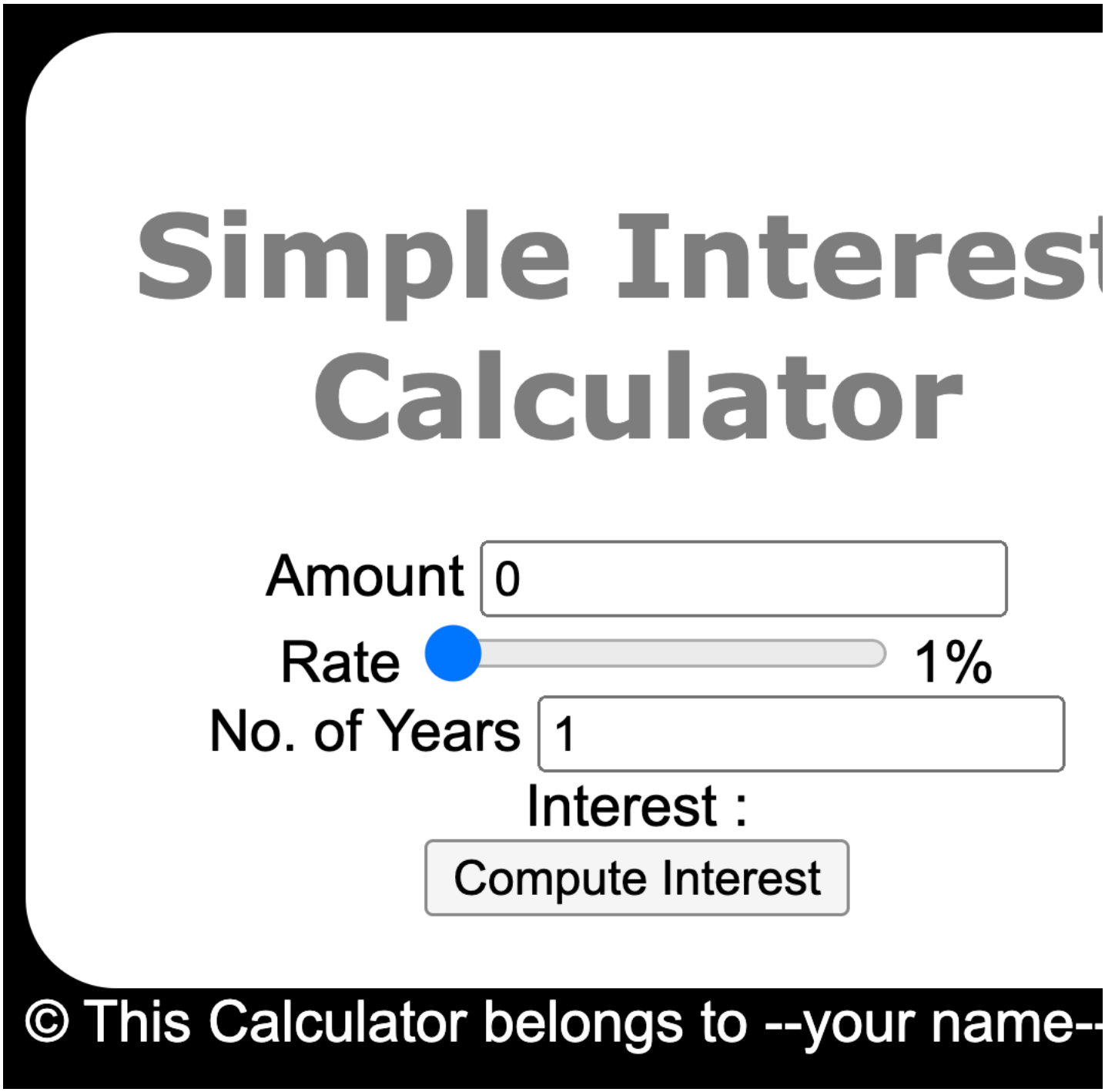
> Write comments in your code which will help debug and maintain the code in the long term.

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`.

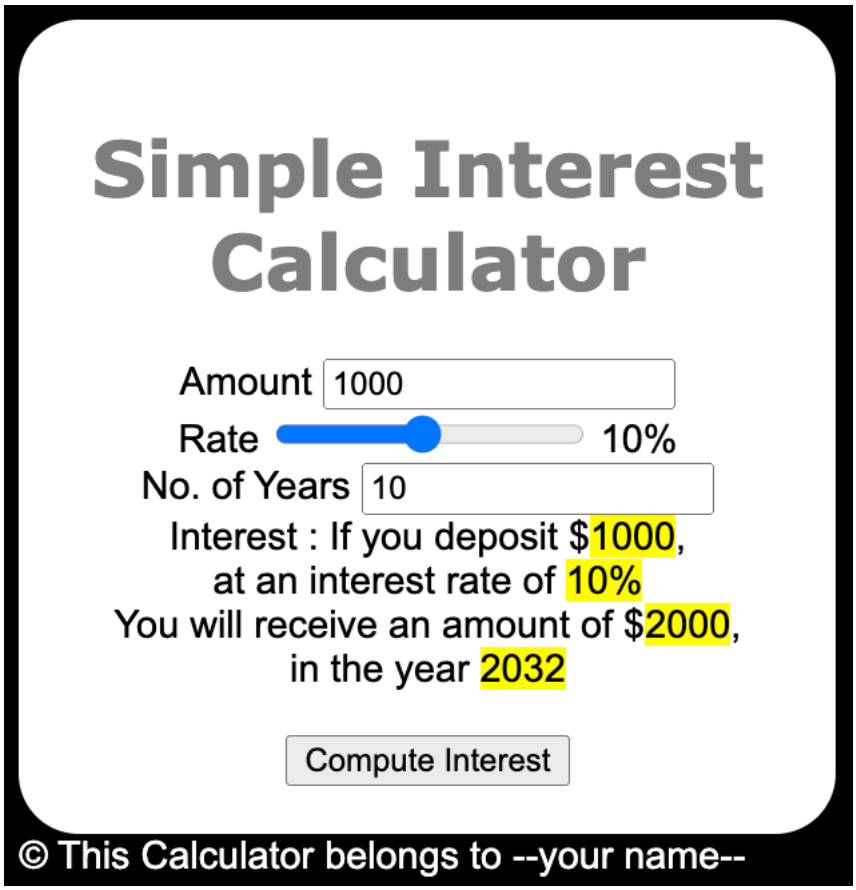


2. Enter these values in the form.

```
Amount = 1000
Rate = 10%
No. of Years = 10
```

Click on Compute button.

The output should be:



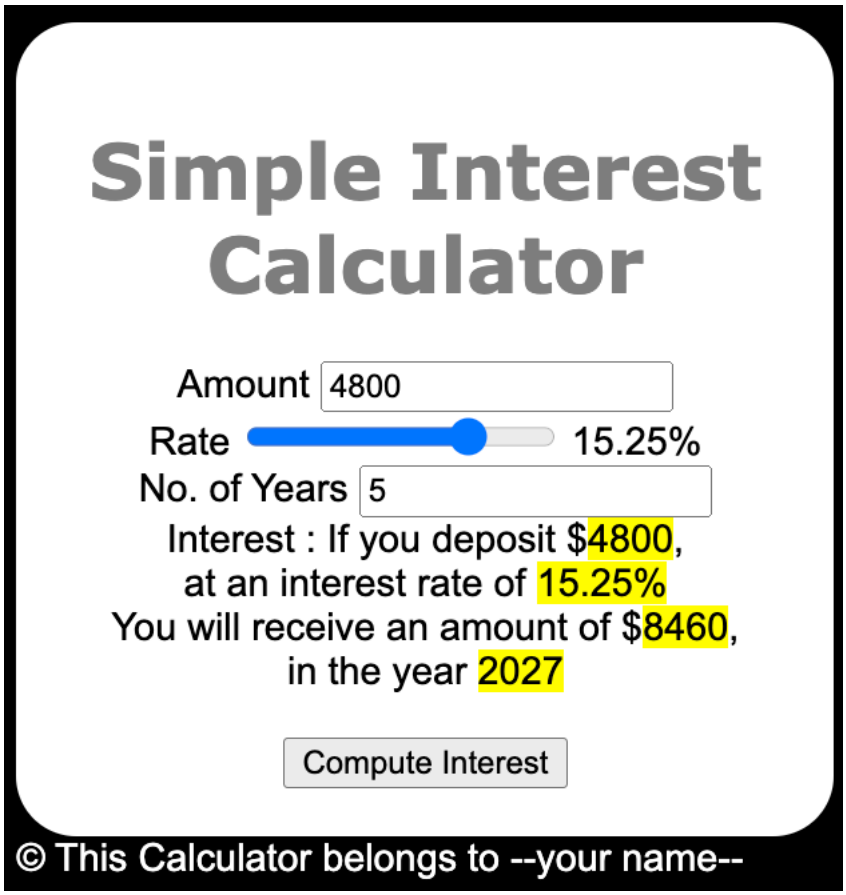
The year is 2022 at the time this lab was written, therefore 2032 is correct. If you are doing this in a different year, make sure the year value is the current year + 10 years.

3. Enter these values in the form.

```
Amount = 4800
Rate = 15.25%
No. of Years = 5
```

Click on Compute button.

You should see the following output:



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.

Congratulations! You have completed the lab!

Author(s)

K Sundararajan

Other Contributor(s)

Michelle Saltoun

© IBM Corporation. All rights reserved.