Assignment 4

JS Function into HTML:

Marsha Jang

Library and Information Technician

Sunday, April 16th, 2023

Algonquin College

Contents

Introduction

The main purpose of this assignment is to understand java-script and the relationship between HTML and JS so to create a webpage more functional and active. Especially in this assignment, I will show a temperature converter from Fahrenheit to Celsius or the opposite and will add some animation factor to show the temperature graphically.

Object

- Program the conversion button
- Input filter for the numeric type
- Error message for the wrong input
- Add a thermometer picture
- Add some styles
- Check the SEO

Default form.

First having those title, headline as well as the input boxes.

```
<div class="thermo">
    <div id="thermo-img">
        <img id="thermometer-img" src="themometer.png" >
        <canvas id="thermometer-bar" ></canvas>
    </div>
    <div class="wrapper" id="thermo-calc">
        <br />
        <br />
        <label>Enter a temperature in degrees Fahrenheit:</label>
        <input type="text" id="Fahrenheit" />
        <br />
        <br />
        <br />
        <button onclick="calcTemp()">Convert to Celsius</button>
        <br />
        <br />
        <br />

        <label>Equivalent temperature in degrees Celsius:</label>
        <input type="text" id="Celsius" />
    </div>
</div>
```

Figure 1: there are two text type inputs and one button on the thermos-calc div

To add a function or action, the button is used to call a script, and those two input has used either to get the input value from users or show the result.

Calculation and filter function

Once the user click the button, it calls calcTemp() function by onclick event. The calcTemp() function is defined in the script tag, do the both filtering work and the calculating Temperature work.

```
// it takes the value from the input box, check the condition, and refresh the
function calcTemp() {
    var fahrenheit = parseFloat(document.getElementById("Fahrenheit").value);
    var celsius = parseFloat(document.getElementById("Celsius").value);

    if (!Number.isNaN(fahrenheit)) {
        celsius = (fahrenheit - 32) * 5 / 9;
        // (x°F – 32) × 5/9 = y°C
        document.getElementById("Celsius").value = celsius;
        drawTemperature(celsius);
    }
    else if (!Number.isNaN(celsius)) {
        // y°F = (x°C * 9/5) +32
        fahrenheit = (celsius * 9 / 5) + 32;
        document.getElementById("Fahrenheit").value = fahrenheit;
        drawTemperature(celsius);
    }
    else {
        window.alert("Please put a numeric value.");
    }
}
```

Figure 2 : calcTemp() function

For the detail:

- First user set one of the input box and click the calculate button
- The onclick calls this calcTemp() function
- First the function takes the value from the input box and change the input into numeric form: I used float to take the decimal form as well.
- The main function is F -> C conversion, but in case the user wants to change C -> F, I also added that condition as well (if the user emptied F section but put value in C, it converts C to F)
- The conversion only happens the converted value is number, and I used “isNaN” function to find if it is a number or not. And either Fahrenheit or Celsius input box is not a number, than it notifies an error message to the user.
- Once number is detected, it calculates based on the formula, and put the value into the input box.
- After the data update, it also call another function drawTemperature() at the end. (I will describe more about this later.)

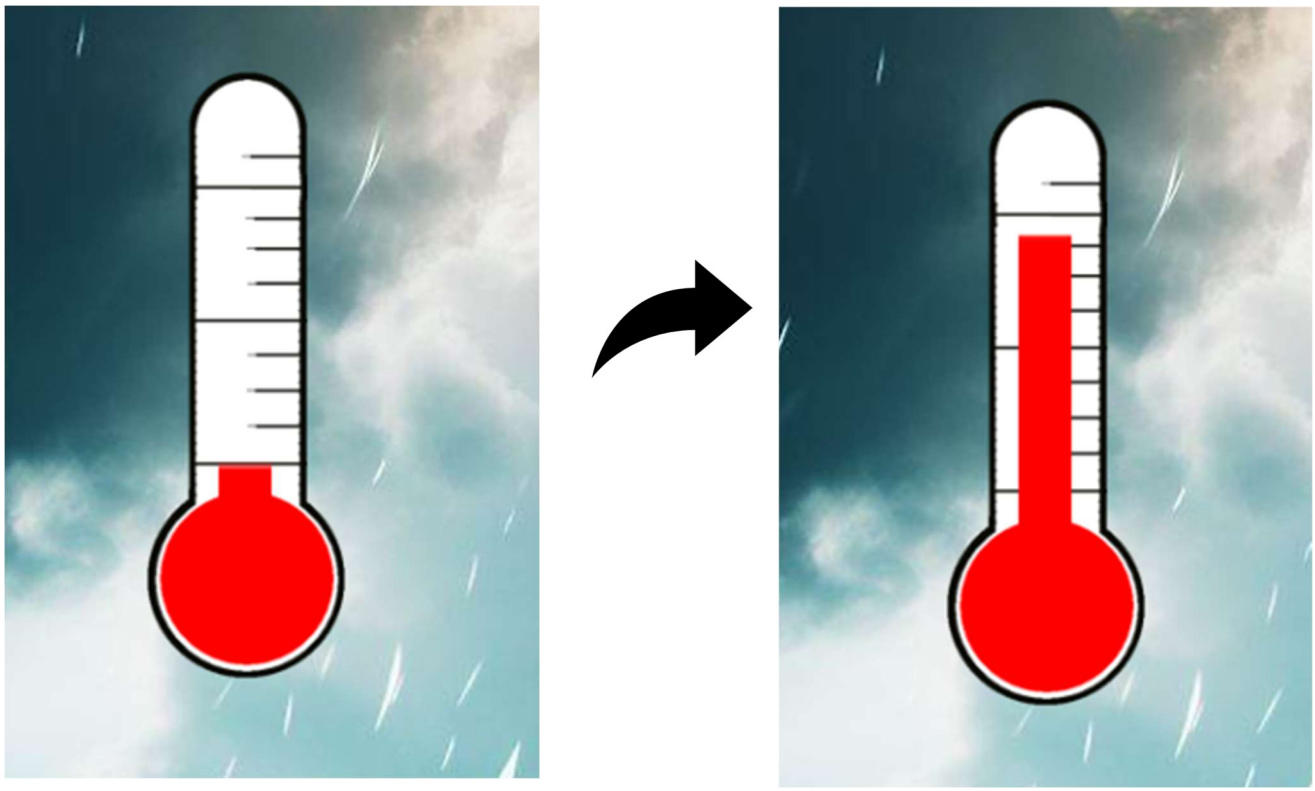
Add a thermometer picture

```
<div id="thermo-img">
    <img id="thermometer-img" src="themometer.png" >
    <canvas id="thermometer-bar" ></canvas>
</div>
```

Figure 3 Thermometer image and canvas

The requirement is for adding a thermometer image, but I also wanted to show the Celsius temperature as well. So I added “canvas” as well.

- To setup the image in front of the converter section, I set those two div sections into a bigger div, and set the div style set as display:flex.
- Because the thermometer image is very simple, I just drew by myself using a painter tool.
- The Canvas was covered on top of image tag setting the same size and the absolute condition because the red temperature bar needs to show on top of the thermometer image.
- The drawing red bar action happens whenever the temperature has been calculated.
- And simply it just shows the Celsius temperature even though you did a C->F conversion.



The drawing function()

The function itself is simple, but figuring it out where to draw the bar was very tricky, and every time I adjust the style I had to setup the detail again since the slight location change moved the bar condition as well.

```
function drawTemperature(celsius) {
    if (celsius > 100) {
                celsius = 100;
            }
    if (celsius < -50) {
                celsius = -50;
            }

    var canvas = document.getElementById("thermometer-bar");
    if (canvas.getContext) {
                var context = canvas.getContext('2d');
                context.clearRect(0, 0, canvas.width, canvas.height);
                context.fillStyle = "#FF0000";
                context.fillRect(138, 83 - celsius*0.7, 24, celsius*0.7 +40);
            }
}
```

- First I set the minimum value for the presentable temperature within the thermometer for a realistic boundary.(-50 for minimum and 100 for maximum)
- The Celsius value ls passed as a parameter into this function, so just drawing a simple rectangular is enough to show the temperature.

Add some styles

- I add the background behind the converter section so it appeals more.
- Also used the wrapper div that shows the calculator looks neat.

Figure 3: full shot.

Having a nice background and shadow effect is always make a bit more elegance.

Check the SEO condition

- Metadata tags has been added

```
<meta name="description" content="Temperature converter">
<meta name="keywords" content="HOMEWORK, HTML, CSS, JavaScript, temperature, converter">
```

- Has title tag in the head section

```
<title>Temperature Conversion Page</title>
```

- Use only 1 “h1” tag

```
}
</script>    <h1>Temperature Conversion Page</h1>
<hr>
```

Conclusion

It has been a big challenge to understand about JS, and how it connects to HTML and CSS. It was hard to understand, especially, when functions did not work even though I followed exact examples from the JavaScript tutorial. I have made often spelling mistakes, but I couldn’t figure out what was wrong. It took time to solve problems. It was nice to see the result that it worked as I expected, but at the same time, I have noticed how much effort is required to make a well-made website. I recognize that more practice time will be needed to draw a good result. I learned a lot within the period while doing this assignment.

References

W3Schools (n.d.). *CSS Tutorial*. W3schools.com. https://www.w3schools.com/css/default.asp

W3C. (n.d.). *CSS Validation Service*. Jigsaw.w3.org. https://jigsaw.w3.org/css-validator/

W3Schools (n.d.). *HTML Canvas Graphics*. W3schools.com. https://www.w3schools.com/html/html5_canvas.asp

W3Schools (n.d.). *HTML JavaScript*. W3schools.com. https://www.w3schools.com/html/html_scripts.asp

W3Schools (n.d.). *HTML Tutorial*. W3schools.com. https://www.w3schools.com/html/html_classes.asp

W3Schools (n.d.). *JavaScript Functions*. W3schools.com. https://www.w3schools.com/js/js_functions.asp

W3C. (n.d.). *Markup Validation Service*. Validator.w3.org. https://validator.w3.org/nu/#file