## Assignment 4 Report: Converting Fahrenheit to Celsius using JS

Elizabeth Reid (41071602)

April 17th, 2023

LIB 2024 – Internet Applications

Prof.: Dr. Huthaifa Abderahman

### Table of Contents

1. Introduction page 1
2. List of Elements and Tricks Used page 1
3. Script Blocks and Comments page 2
4. SEO Standards page 3
5. Conclusion page 4
6. References page 6

### Introduction

In order to recreate the converter given as an example with HTML, CSS, and JS coding, I started by laying out a template for what I wanted the content to contain in the HTML file. This allowed me to visualize what text elements I needed, like the heading and the description, and what new elements I would need to include that would work with both the CSS and the JS. This meant looking for examples of other temperature converters that used JS functions, some of which I have included under my Reference section of this report.

Once the bare-bones content for the webpage had been laid out and I selected another converter that I was going to use for inspiration, I began to apply various CSS attributes in order to have the converter look as much like the vision I had for it as possible. Once all of the aesthetic changes were applied, I began to work on the JS script in order for the webpage to actual function as its intended use. This was, again, with inspiration and code-sharing from the resources referenced below.

### List of Elements and Tricks Used

*The following tags are in order of use in the HTML file. Only tags not previously used are fully explained.*

**<!DOCTYPE html>**

**<html lang=“en”>**

**<link href=[…]>**

**<meta charset=“utf-8”>**

**<meta name="description" content="[…]”>**

**<meta name="keywords" content="[…]”>**

**<title>**

**<body>**

**<h1 class=“[…]”>**

**<!—description above […]-->**

**<p class=“[…]”>**

**<img class=“[…]”>**

**<div class=“[…]”>**

**<input type=“text” […] >**

This tag creates a box intended for the user to add their own input on the webpage, in this case text, as defined by the type=“text” element.

**<button id=“[…]”>**

This tag creates a button to be interacted with by the user. Its effect on the webpage is added via the JS file.

**<script src=“[…]”>**

This tag links the contents within the HTML file to the functions defined in the JS file. This tag must be included in the <body> of the HTML file.

*The following tags are in order of use in the CSS file.*

**body {**

**.head {**

**.description {**

**.converter {**

**#convert {**

This tag, much like the previous tags, takes the unique identifier given to an element in the HTML file in order to apply changes to it. The # denotes that it is identifying this element by its ID rather than its class.

**#convert :hover {**

This applies changes to an HTML element with the ID ‘convert’ while the mouse is hovering on this element.

**input {**

This tag applies changes to all elements of the same type, in this case all input elements, without regard for individual class or ID.

**button {**

*The following tags are in order of use in the JS file.*

**document.getElementById(‘[…]’).onclick=[…]**

This tag prompts the browser to interpret the data input in the ID defined in the brackets, then use said data to run the function following the onclick action.

**function […]() {**

This tag defines the actions to be performed when other tags, like the previously explained getElementById, call upon the function with this name.

**if ([…] !=‘ ‘) {**

This tag begins the conditional aspect of the tempConvert function, whose exact task will be defined in the Script Blocks and Comments section of this report.

**[…].value = parseFloat([…]).toFixed(1);**

This tag allows for the browser to interpret values added to the input area up to the first non valid character.

### Script Blocks and Comments

*The screencaptures, from left to right, represent the HTML, CSS, and JS scripts. Where a fourth screencapture of the webpage content is included, HTML and CSS scripts will be above the JS and webpage screencaptures.*

Graphical user interface, text

Description automatically generated

The first 12 lines of code from this file include the standard opening (<!DOCTYPE html>, <html lang=“en”>, etc.) tags as well as the keyword tags that allow for SEO standards to be met. The title tag, which defines the title of the webpage as it is displayed on the browser’s tabs, is also given on line 11.

Text

Description automatically generated with low confidenceText

Description automatically generated with medium confidence

When working in conjunction, these two portions of the script create the content before the actual converter. Line 13 and 14 of the HTML file opens the body of the content and creates a heading to be styled by the attached CSS file. Lines 17 displays instructions for the user to read before actually using the converter, while line 20 imports the photo used for decoration and as a way to stylize the page slightly.

In the attached CSS file, the body { tag defines the overall color of the text and the background, allowing entire page to have the same aesthetic feel. The .head { and .description { tags stylize the main heading on the page and the previously mentioned instructions in order to follow the theme set out by the body { tag. The .gauge { tag sets out the same function, but is also used to invert the colour of the image as it was a dark line drawing, and inverting it would force it to match the white text of the rest of the webpage.

Text

Description automatically generatedText

Description automatically generated

This section of the HTML file creates an input area for the user to type the temperature they are hoping to convert to either Fahrenheit or Celsius, depending on their starting unit. Without the JS script attached to this file, which will be discussed in the following section, this is simply a placeholder that would allow the user to type any text in either input section. The tags on lines 27 and 28 are used to create buttons that will also be programmed later via the JS script.

The CSS portions pictured above are used to create the stylistic elements around the input elements, including the border around the actual converter, the padding, the margins, and the placement of the calculator.

Graphical user interface, text

Description automatically generatedText

Description automatically generatedText

Description automatically generated

This portion of the code is the main component of the webpage as it establishes the link between the HTML files and the JS programming that makes it work. On line 32 of the HTML file, the <script> tag instructs the browser to use the functions and values created in the source JavaScript file in order to run the appropriate processes for the converter.

The screencapture just below the HTML file shows the JS code that sets out the functions used to actually calculate the conversion between Fahrenheit and Celsius according to the input value from the user. Lines 4 through 19 define the conditions and calculations done behind the scenes in order to give the user the proper conversion once they click convert. Lines 22 through 24 programs the functions applied to the button that allows the user to clear both the Celsius and Fahrenheit fields simultaneously in order to restart their calculations.

The photo to the right of the other code snippets is from the attached CSS file, which is used to change the appearance of the product of the HTML file. Most changes are basic, including changing the colour of the buttons and applying a set border size and colour. There are, however, changes to the hover colour and appearance of the buttons, laid out on lines 55-58 and 66-69. This will create a deeper colour when the user hovers over the buttons.

The screencapture below showcases what the final product from these specific sections of code produces.

A screenshot of a computer

Description automatically generated with medium confidence

### SEO Standards

While writing the code for this temperature converter, applying SEO standards was similar to the last few assignments in the sense that much of the process was the same. According to HubSpot’s “Introduction to Search Engine Optimization” when displaying results through a search engine, script elements are used as a way to qualify whether a website matches the user’s search criteria by “looking for clues about what topics your website covers,” (HubSpot 7). This is done through the HTML tags on lines 8 and 9, which include descriptions and keyword indicators.

According to the same resource from HubSpot, it is important to have both a short headline that accurately describes the content of the page and a sub-heading that offers more detail, both of which should “include the keyword or phrase that the webpage covers,” (HubSpot 17). My headline – “Temperature Converter” plain and simply describes the purpose of the webpage, while the sub-heading – the text below the heading explaining the process to the user – also features several of the keywords included in the keyword tag at the beginning of the HTML file.

### Conclusion

Overall, this webpage was, again, simple to code in terms of the HTML and CSS aspects of the page. Though the JS was a new concept to me, trying to get the functions to work properly was only a matter of trial and error and reading through different tricks used by others who have had the same issues with the same webpage idea.

### References

Carolyn. (n.d.). *Old artwork illustration gauge clip art gauge instrument device images*. Digital Stamp Design. Retrieved April 7, 2023, from http://digitalstampdesign.blogspot.com/2017/08/old-artwork-illustration-gauge-clipart.html

*CSS box alignment - CSS: Cascading style sheets: MDN*. CSS: Cascading Style Sheets | MDN. (n.d.). Retrieved April 16, 2023, from https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\_Box\_Alignment

*CSS Text*. W3Schools. (n.d.). Retrieved April 16, 2023, from https://www.w3schools.com/css/css\_text.asp

*Day 8: Create a Button*. HackerRank. (n.d.). Retrieved April 16, 2023, from https://www.hackerrank.com/challenges/js10-create-a-button/topics

HubSpot. (n.d.). *Introduction to Search Engine Optimization: Getting Started with Seo to Achieve Business Goals*. *HubSpot*. Retrieved from https://www.hubspot.com/hs-fs/hub/53/file-13221845-pdf/docs/ebooks/introduction-to-seo-ebook.pdf.

*Invert() - CSS: Cascading style sheets: MDN*. CSS: Cascading Style Sheets | MDN. (n.d.). Retrieved April 16, 2023, from https://developer.mozilla.org/en-US/docs/Web/CSS/filter-function/invert

*Temperature converter*. CodePen. (n.d.). Retrieved April 16, 2023, from https://codepen.io/Caparico/pen/AXOBBY

*Using an HTML button to call a JavaScript function*. Stack Overflow. (2020). Retrieved April 16, 2023, from https://stackoverflow.com/questions/1947263/using-an-html-button-to-call-a-javascript-function

### Content Used on Webpage

Carolyn. (n.d.). *Old artwork illustration gauge clip art gauge instrument device images*. Digital Stamp Design. Retrieved April 7, 2023, from http://digitalstampdesign.blogspot.com/2017/08/old-artwork-illustration-gauge-clipart.html