# HTML Code Elements

| Element of Code | Description and usage of the function | How I Used the Element |
|---|---|---|
| <!DOCTYPE html> | • Specifies the HTML document type declaration, indicating an HTML5 version document.<br>• Used at the beginning of the code. | This is used in all HTML Coding |
| <html> | • Represents the root element of the HTML page.<br>• Encloses the entire content of the HTML document. | This holds all content inside of my HTML document |
| <head> | • Contains metadata and other head elements like title, stylesheets, etc.<br>• Holds the metadata and stylesheets used in the document. | The head is holding my main header and all the CSS styling |
| <title> | • Defines the title of the HTML document.<br>• Sets the title displayed in the browser's title bar or tab. | When on the page it displays in the search bar "Rent vs Buy: ZG" |
| <link> | • References an external CSS +stylesheet used for styling the HTML document.<br>• Links the HTML document to an external CSS file. | I have that it is referencing for CSS off a page named "stylesheet"<br>but I decided to hold all CSS on my main coding page |
| <style> | • Contains CSS rules for styling the elements within the HTML document.<br>• Defines the CSS rules directly within the HTML document. | This is controlling all elements inside of my Page.<br>Headers, input, and output boxes etc.<br>Inside of this section is CSS elements. |
| <body> | • Represents the main content area of the HTML document.<br>• Contains the visible content of the HTML document. | This section has all input and output fields as elements. These are where you set classes, ids, and types of text boxes in my code. |
| <header> | • Defines the header section of the document. | This is the blue box at the top of my page. |

| | | |
|---|---|---|
| | • Contains the title of the calculator. | Header is just a section that classes headers |
| <h1-h5> | • Defines various headings used to structure and label sections of the calculator.<br>• Sets different levels of headings within the document. | I have these set throughout the calculator to state what each section of input or output boxes are for.<br>separating the Loan and Rent sections to the results as well. |
| <form> | • Wraps the input fields and buttons, creating a form for user input.<br>• Contains the input fields and buttons for user interaction. | This holds all the boxes. Input sections can only be used inside of a “form”. |
| <div> | • Divides the content into logical sections and provides containers for grouping related elements.<br>• Creates containers for organising elements in the HTML document. | This is the most used command, Sectioning and creating groups for classing styling and identifying. |
| <label> | • Associates a label with an input element, providing a description or name for the input.<br>• Labels the input fields with descriptive text. | This is where you set what the text will show on the website.<br>id and classes are what the elements are referred to in the code but the name that is displayed is in the label. |
| <input> | • Creates input controls such as text boxes and buttons for user interaction.<br>• Allows user input and control interaction within the form. | These are all boxes where the user can enter numbers for the calculations. |
| <placeholder> | • Provides a short hint or example text that is displayed in an input field before the user enters data.<br>• Used to indicate the expected input in an input field. | This is used to display behind where numbers are inputted |
| <button> | • represents a clickable button that triggers specific actions when clicked.<br>• Initiates actions when clicked by the user. | The button function was used for the “calculate” buttons but after a bug, we changed the buttons to styled id and divisions. |

| | | |
|---|---|---|
| <type=”number”> | • Changes what is allowed to be typed into the input box.<br>• Will only allow numbers to be entered into the input box. | All input boxes were set as “number” but later was changed to tel due to a scroll bar creating an unwanted feature in the code. |
| <type="tel"> | • Type tel changes the restrictions to is able to be entered into an input box.<br>• Mobile users when clicking on the input box will be immediately prompted with the telephone number input box.<br>Also only allowing numbers in the input field. | Using type tel, I can stop users entering alphabetic characters into input boxes and only allowing numbers. |
| <Value> | • Set a designated number value in an input box.<br>• Possible uses are:<br>Setting numbers for users to see, a stationary example number. | I use value to test the calculations, making it easier to refresh the test page without resetting all my inputted numbers. |

# CSS Code Elements

| Element of Code | Description of function | How I Used the Element |
|---|---|---|
| “#” | The # symbol is used in CSS to select an element with a specific ID. It is followed by the ID name of the element. | In the code, the # symbol is used to select elements by their ID. For example, $("#calculate") is used in JavaScript to select the element with the ID "calculate". |
| “.” | The . symbol is used in CSS to select an element with a specific class. It is followed by the class name of the element. | The symbol is used to select elements by their class. For example, in the CSS styling, the .input-box class is used to target elements with the class "input-box". This allows for applying specific styling and behavior to those elements across the webpage. |
| Button, body, header | Just the name of the div/ section is used when referencing to a main element of a page. | These are used as they would in any other document, simply setting padding and margins for looks. |
| margin | The margin property sets the margin space around an element. It controls the space between the element's border and surrounding elements. | In my code, I have used the margin property to create spacing around various elements. For example, in the CSS styling, I have used margin-top: 25px; to add a margin space of 25 pixels at the top of the input box |
| Padding | The padding property sets the padding space inside an element. It controls the space between the element's content and its border. | In the code, the padding property is used to create space within various elements. For example, in the CSS styling, the. input-box class uses padding: 15px; to add padding of 15 pixels inside the input box. |
| Background-color | The background-color property sets the background color of an element. It allows for customizing the color behind the element's content and padding area. | In the code, the background-color property is used to define the background color of various elements. For instance, in the |

| | | |
|---|---|---|
| | | CSS styling, the <header> element uses background-color: #004f99; to set its background color to a shade of dark blue. This helps in visually distinguishing and styling different sections of the webpage. |
| Color | The color property sets the color of the text content within an element. It allows for customizing the text color using various color formats such as named colors, hexadecimal codes, RGB values, or HSL values. | In the code, the color property is used to define the color of text within different elements. |
| text-align | The text-align property controls the horizontal alignment of the text content within an element. It determines whether the text is aligned to the left, right, center, or justified within the container. | The text-align property is used to align the text content within different elements. For example, in the CSS styling, text-align: center; can be applied to a <div> element to horizontally center-align the text content within the container. |
| Display | The display property specifies how an element should be displayed on the webpage. It determines the type of box model used for the element, such as block, inline, inline-block, or flex. | In the code, the display property is used to create a flex container and enable a flex layout. For example, in the CSS styling, the display: flex; rule is applied to the body element, making it a flex container, and allowing its child elements to be flex items. |
| width | | |
| Height | The height property sets the height of an element. It determines the vertical dimension of the element's content box, excluding padding, borders, and margins. | In the code, the height property is used to define the height of various elements. For example, in the CSS styling, the output-box1 class uses height: 100px; to set a fixed height of 100 pixels for output boxes. |
| Font-size | The font-size property sets the size of the text content within an element. It determines the height of the font glyphs and affects the overall visual appearance of the text. | The font-size property is used to define the font size of various elements. For instance, in the CSS styling, the h1 element within the header section uses font-size: calc(1.325rem + 0.9vw); to dynamically adjust the font size based on the viewport width. |

| | | |
|---|---|---|
| font-weight | The font-weight property sets the thickness or boldness of the text within an element. It determines the visual weight of the text characters, ranging from normal to bold or even numeric values. | In the code, the font-weight property is used to make certain text elements bold. For example, in the CSS styling, the h2, h3, and h4 elements within the header section use font-weight: bold; to give them a bold appearance. |
| Box-shadow | The box-shadow property adds a shadow effect to an element. It allows for creating visually appealing depth and dimension by specifying the shadow's position, blur radius, spread distance, color, and other properties. | The box-shadow property is used to add a shadow effect to various elements. For instance, in the CSS styling, the .input-box class uses box-shadow: 0 2px 6px rgba(0, 0, 0, 0.3); |
| Cursor | The cursor property specifies the type of cursor to be displayed when hovering over an element. It allows for customizing the mouse cursor's appearance to provide visual feedback to the user. | In the code, the cursor property is used to change the cursor style for interactive elements. For example, in the CSS styling, the button, #calculate, #compare, and #compare2 elements use cursor: pointer; to change the mouse cursor to a hand pointer when hovering over them. |
| Min-height | The min-height property sets the minimum height of an element. It ensures that the element's height is never smaller than the specified minimum height, even if its content or other factors would make it smaller. | The min-height property is used to set a minimum height for the body element. For instance, in the CSS styling, min-height: 100vh; ensures that the body element occupies at least the full height of the viewport, regardless of the content size. |
| float | The float property specifies whether an element should float to the left or right of its container, allowing other content to wrap around it. It is commonly used to create multi-column layouts or to position elements in relation to surrounding content. | In the code, the float property is used to position elements to the left or right. For example, in the CSS styling, the .rent_container1 class uses float: left; to position the container to the left side of its parent element |

| | | |
|---|---|---|
| Max-width | The max-width property sets the maximum width of an element. It ensures that the element's width does not exceed the specified maximum width, allowing it to be responsive and prevent overflow on smaller screens. | The max-width property is used to limit the width of various elements. For instance, in the CSS styling, the .input-box, .input-box2, and .output-box1 classes use max-width: 500px; to ensure that the elements do not exceed a width of 500 pixels. |
| Opacity | The opacity property sets the transparency level of an element. It determines the degree to which the element is see-through, with a value of 1 indicating fully opaque and a value of 0 indicating fully transparent. | the opacity property is used to adjust the transparency of various elements. For example, in the CSS styling, the button:hover selector uses opacity: 0.7; to reduce the opacity of the button when it is being hovered over. |
| Flex-direction | The flex-direction property specifies the direction of flex items within a flex container. It determines whether the items are laid out in a horizontal row or a vertical column, and the order in which they appear | In the code, the flex-direction property is used to control the direction of flex items. For instance, in the CSS styling, the body element uses flex-direction: column; to arrange its child elements in a vertical column. |
| Border-radius | The border-radius property rounds the corners of an element's border, creating a smooth and curved appearance. It allows for customizing the degree of curvature by specifying the radius of the rounded corners. | n the code, the border-radius property is used to round the corners of various elements. For example, in the CSS styling, the .input-box, .input-box2, and .output-box1 classes use border-radius: 5px; to give their borders a slightly rounded appearance with a radius of 5 pixels. This makes buttons and boxes have rounded edges for appearance. |

# JavaScript Code Elements

| Element of Code | Description of function | How I Used the Element |
|---|---|---|
| Parsefloat | parseFloat() is a JavaScript function used to convert a string into a floating-point number. It parses a string and extracts the numerical value, allowing for mathematical operations and calculations. | The parseFloat() function is used to convert input values from string format to floating-point numbers. For example, in the JavaScript code, parseFloat() is used to convert the values obtained from input fields, such as $("#loan_amount").val(), into floating-point numbers for further calculations. |
| ParseInt | When this function is applied to a string, it parses it from left to right until it reaches a character that isn't a valid initial part of an integer number representation. Then it returns the integer number it has read up to that point. | This element was used when specifying the term. We don't want users to enter decimal numbers into the term, so parseInt will take the number, and round it to the closest whole number. In my case the number of years. |
| const | The const keyword is used in JavaScript to declare a constant variable. A constant is a value that cannot be reassigned or modified once it has been assigned. It provides a way to | In the code, the const keyword is used to declare constant variables. For example, in the JavaScript code, const annualReturnRate = parseFloat($("#annual_return_rate").val()) |

| | | |
|---|---|---|
| | create variables that remain unchanged throughout the program. | / 100; is used to declare a constant variable named annualReturnRate. The value of annualReturnRate is obtained by parsing the input value from the #annual_return_rate element and dividing it by 100. The const keyword ensures that the value of annualReturnRate remains constant and cannot be changed later in the code. |
| var | The var keyword is used in JavaScript to declare a variable with function scope or global scope. Variables declared with var are not block-scoped, meaning they are accessible within the function or the global scope where they are defined. However, they can be reassigned and modified throughout the program. | The var keyword is used to declare variables. For example, in the JavaScript code, var loan_amount = parseFloat($("#loan_amount").val()); is used to declare a variable named loan_amount and assign it the parsed float value of the input value from the #loan_amount element. The var keyword allows the variable loan_amount to be accessed and modified within the scope it is defined. |
| | | |
| | | |
| | | |