Styles.css (default):

### 1. :root {}

This selector is typically used to define CSS custom properties (variables) that you want to be accessible globally in your document. Since it's empty, you might consider removing it if you don't plan to add any variables, or use it to define values you'll reuse, such as colors or font sizes. For example:

:root {

--primary-color: #333;

--font-stack: Arial, Helvetica, sans-serif;

}

### 2. \*, \*::before, \*::after

This global style block sets box-sizing: border-box; for all elements, including all pseudo-elements (::before and ::after). This is a common best practice because it makes dealing with widths and heights much easier, as it includes padding and border in the element's dimensions.

### 3. body

You've set a default font and removed the margin from the body, which is also a common practice to handle consistent styling:

* font-family: Arial, Helvetica, sans-serif; ensures that your text uses these fonts if available, falling back to sans-serif if neither Arial nor Helvetica is available.
* margin: 0; removes the default margin browsers apply to the body.

### 4. h1, h2, h3

Setting margin: 0; on these headings removes default top and bottom margins. While this is fine, consider whether you might need some space around these elements. If you do, you could later add specific margins in your CSS for particular instances, or provide a more nuanced default style here.

### 5. button

The font-family: inherit; rule is a thoughtful inclusion. It ensures that your buttons inherit the font style from their parent elements, which keeps your typography consistent without needing to redefine the font family for each button.

Box sizing and \* selector:

Sure, let's dive deeper into the box-sizing: border-box; property and the \* selector in CSS.

### box-sizing: border-box

The box-sizing property in CSS controls how the width and height of an element are calculated. There are mainly two values for box-sizing:

**content-box** (default): This is the standard CSS box model. If you set an element’s width and height, these dimensions do not include padding, borders, or margins. Instead, the width and height only apply to the element's content area. The total width and height of the element, including the padding and border, will be larger than what you set.

**border-box**: When you set box-sizing to border-box, the width and height of an element include the padding and border. This means if you set a width and height for an element, those dimensions will encompass not just the content but also the padding and borders. The margin is not included in these calculations. This is particularly useful because it simplifies the way you can lay out elements without having to do additional math to account for padding and borders.

Here’s a visual comparison:

* **content-box**: width = content's width
* **border-box**: width = content's width + padding + border

Using border-box is often more intuitive as it prevents the element from expanding unexpectedly due to padding and border widths, making layout design simpler and more predictable.

### The \* Selector

The \* selector is known as the universal selector in CSS. It matches every single element on the page. This is incredibly powerful when you want to apply a style broadly to all elements within a document. Here are a few things to consider about the universal selector:

* **Performance**: Historically, there were concerns about the performance implications of using the universal selector, particularly on large documents. However, modern browsers handle this efficiently, and it’s commonly used in CSS resets or to set box-sizing globally, as in your stylesheet.
* **Cascading**: Styles defined with the \* selector have the lowest specificity. This means any other styled elements with more specific selectors will override the universal styles.
* **Use Cases**: Besides box-sizing, other typical uses of the universal selector include resetting margin and padding to zero or setting a consistent box model across all elements.