Troubleshooting Your Custom CSS

Download the PDF of this article.

In this Article

Related Articles

Introduction

> **Note:** Our support team **does not** provide custom code and cannot help edit, write, or design custom code for your forms.

If your custom CSS is not working as it should, there are a couple of methods you can use to troubleshoot your code. These methods have been listed in the recommended order, and following them step by step will help to ensure that you discover what may be going wrong within your custom CSS.

Begin Troubleshooting

Please perform this entire troubleshooting process within your direct FormAssembly form URL. If you try to view custom CSS in an embedded form, there's a chance the external webpage's code could conflict with your CSS.

1. First, **check to make sure that your code is correctly tagged** with <style> tags like so:

   ```
   <style type="text/css">
   Your code goes here!
   </style>
   ```

2. Remove all of the code from your Custom Code box, save your form, and then **add in some simple CSS** to check if the Custom Code box is working. For example, you could try to use this code to remove the submit button:

   ```
   <style type="text/css">
   input.primaryAction {
   display: none;
   }
   </style>
   ```

   If simple code is not working, please reach out to our Support Team!

3. **Check to make sure you're using the correct form element** in your CSS. We have a list of form elements here, and you can also click here to view our help document on styling individual elements .

4. You can also check to be sure that all of your colons (:) and semicolons (;) are being used in the correct locations. A colon is used between properties and values in your code, essentially to state what you want done to the specific property of the element. For example, if you want to change the color of an element to

blue, you would add a colon between the property and the value:

```
{color: blue;}
```

When working with CSS, every declaration always ends with a semicolon:

```
{color: blue;}
```

If your colons and semicolons are switched, the CSS for that element will fail.

5. Have you added **too many spaces** to your code? For example, there should not be spaces between your values and your units since values and units can not be separated. Having the space there will make the value invalid, which will make the entire rule invalid.

   This code will not work:

   ```
   {margin-left: 50 px;}
   ```

   However, this code will work:

   ```
   {margin-left: 50px;}
   ```

   This is only one example of having too many spaces, so if you are unsure, you can try using a CSS validator to check for unwanted spaces or other similar errors! These validators can be found online.

6. **Do you have any code besides CSS** within your <style> tags? HTML, JavaScript, and more will cause CSS to fail or will fail themselves within <style> tags.

7. **Try changing the form's theme to be "blank"**. This will allow you to check if any of the theme's code is interfering with your custom code.

8. If you want to keep your theme, you can instead **try using !important in your code** to see if this helps. Sometimes your code and the code of the form or the theme will conflict, so you can use !important to make your custom code a priority if needed. However, this is not always recommended because using !important could also override other custom code or theme values, so please use with caution! For example, this code can remove the background from your form's theme by overriding the theme CSS with !important:

   ```
   <style type="text/css">
    .wForm form, .wForm fieldset {
      background-image: none !important;
      background-color: transparent !important;
    }
   </style>
   ```

9. Finally, if none of these methods were effective, then the issue may be within your code itself. Please try using an automatic CSS testing tool (there are plenty online!) or test the CSS in another webpage to see if the code works outside of FormAssembly.