

Code Editor

This assignment is intended for you to understand building an application using HTML, CSS, and JavaScript. You will be able to complete this challenge by using the content from the course and reading online resources. Please complete the requirements listed below.

Many of the examples on W3Schools have a "Try it Yourself" button. Those buttons launch a new window with a code editor and a preview pane. Create your own version of this tool that meets the requirements listed below.

- Here's an example on w3schools:
 - http://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst
 -

Note: There may be some differences between the video and the instructions here. If you find differences, then make sure you follow the instructions listed here in this document. The video is meant to be an example to help you but may not resemble the actual assignment.

Essential Requirements:

- **The file should be named "<YourName>Editor.html".** Replace YourName with your real name. For example, my name is Hao Zhang, my filename would be "HaoZhangEditor.html". This requirement will be applied to every assignment, therefore, always remember to rename your file accordingly.

- **Write and complete the following comment in your HTML document:**

```
<!--  
  name:  
  date:
```

```
-->
```

- The challenge must be a well-formed HTML5 document **(Including the 5 mandatory tags discussed in class)** that uses CSS3 for the page design and layout.
- Use **External CSS** and use a **<link>** in head of document
 - **DO NOT** use Inline CSS or Internal CSS
 - You can name your file <Pawprint>Editor.css
- **Your file must past validator without error(s)**

Code Editor

- Include the additional items to pass the validator at <https://validator.w3.org/> by using the "Validate by File Upload"
 - When you are finished writing your code, take your code and run it through the validator. In order to pass the validation, the validator may ask to add additional items or fix your code. You should add/fix these request(s) to your code. **Make sure there are no warnings or errors with the validator by receiving the following message:**

Document checking completed. No errors or warnings to show.

Requirements:

1. **All scripts (JavaScript)** should be in an **external JS file**. You can name your file <Pawprint>Editor.js and follow the example listed above for naming the HTML file.
2. Start by creating **two text areas**, one for **writing your CSS code** and one for **writing your HTML code**.
 - a. Put a border so you can easily see the text areas
3. **CSS and HTML for text areas:**
 - a. Text areas should have a **non-empty placeholder indicating which text area holds what code**.
 - b. If you **double click on the CSS text area**, then the following default CSS will appear inside the text area as a String of Text (Should be formatted nicely in order to be readable):
 - `h2 {color: #FF6F61; text-align: center;}`
 - `p {font-family: helvetica; font-size: 20px; border: 2px solid red; border-radius: 12px;}`
 - c. If you **double click on the HTML text area**, then the following default HTML will appear inside the text area as a String of Text (Should also be formatted correctly):
 - `<h2> Welcome to my text editor!</h2>`
 - `<p>You can test and create your own HTML and CSS in this text editor</p>`
4. Use a **div to create a preview pane**
 - a. Where you show the result of your code, there are many examples from class, put a border so you can easily see the preview pane
5. The **two text areas and the preview pane** must be **next to each other**.
 - a. Use a wrapper div like shown in class, look at screen shot example for the visualization
6. Create **three buttons: (HTML, CSS, and Erase)**
 - a. **HTML: Clicking the HTML button** should **retrieve the contents** of the **HTML** text area and use them to **populate the preview pane**.
 - b. **CSS: Clicking the CSS button** should toggle the contents of the **CSS** text area **on and off**. This means to literally display the style in the CSS text area on the page or make the styles take effect dynamically and also not display the CSS depending on the button click

Code Editor

- c. **Erase: Clicking the Erase button should reset the contents of both text areas and the preview pane.** Write your own JavaScript function for resetting the contents of both text areas.
7. Create an H1 on the top of you page saying "Code Editor"
 - a. Look at screenshots for an example but the text may not be the same
8. Create H4 in the bottom of your page and put your name/nickname there
 - a. Look at screenshots for an example but the text may not be the same
 - b. If you **click H4 twice**, a **prompt will appear** and ask the user what they would like **to change the title of the webpage to. What the user enters will be displayed in the H1 and the title tag** which will display at the top of the page and the top of the browser in the tab, respectively.

CSS Requirement:

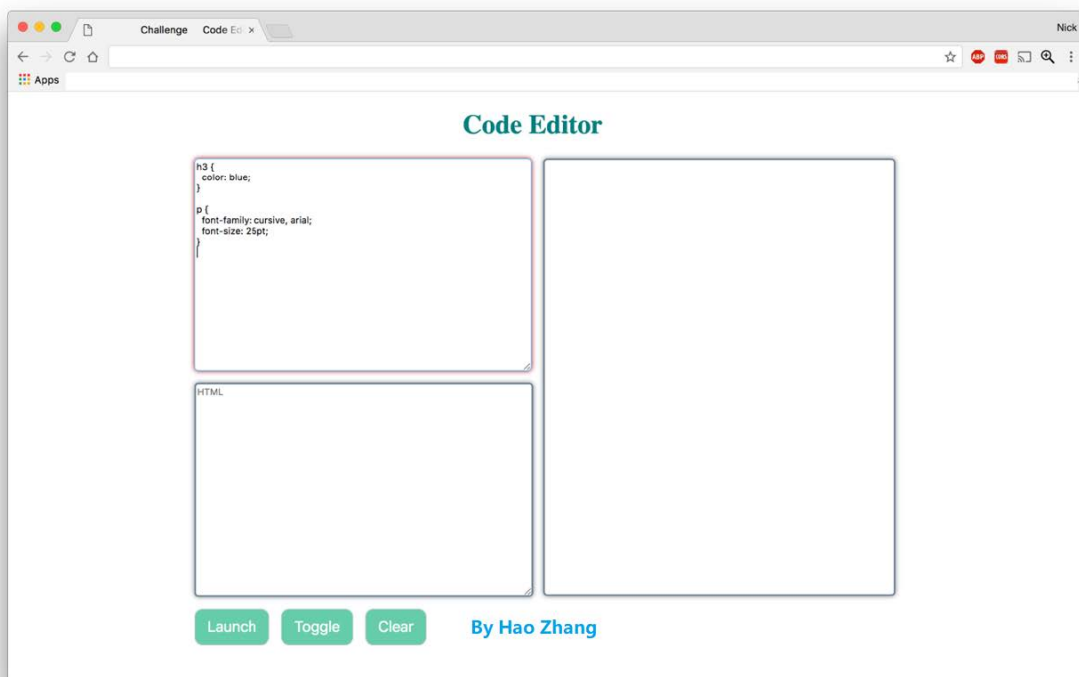
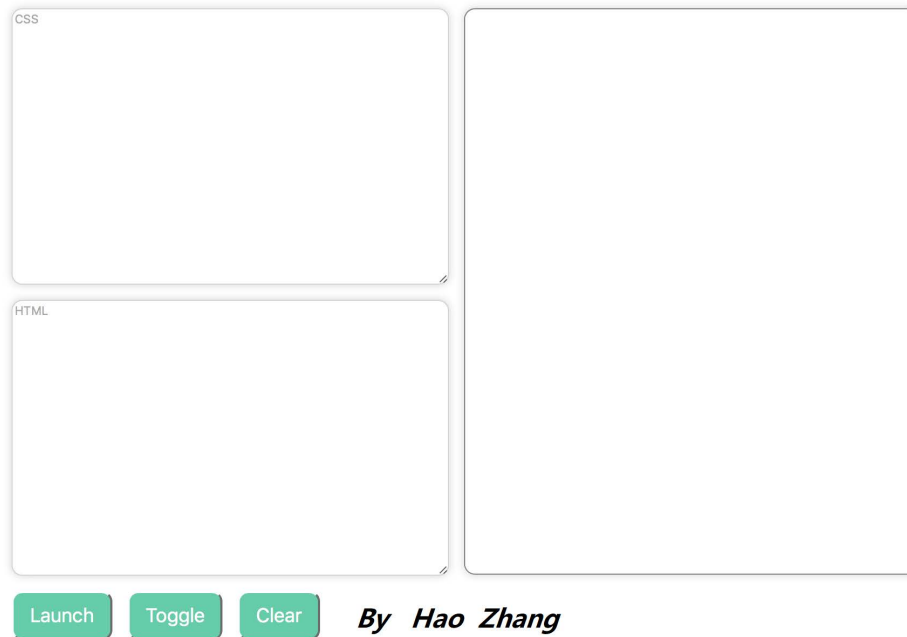
1. Center everything in the middle of the page
2. Make the toggle button change color when it is on/off
3. Make sure both text areas and preview pane div have a border with size, style, and color other than the default values
4. Make sure both text areas and preview pane div have a box shadow, with values other than the default
 - a. The text areas and preview pane div will have a default box shadow as described above, but when the user clicks on one of the text areas, they will change in color to notify the user they selected this text area. The two colors should contrast in color from light to dark or dark to light. The preview pane will not change.
5. Create a tooltip for each button explaining the use of each button so when you hover over the button it will show you an explanation on how to use the button. See the screenshots for examples even though the text in the screenshot may be different.
 - a. HTML: This button is used to show the result of your HTML code
 - b. CSS: This button is used to turn on or off the CSS
 - c. ERASE: This button is used to erase your code and reset the preview pane
6. Add 1 second Opacity Transition to the tool tip.
7. **The rest of the design of the application is up to you. I want you to have fun and show me your creativity. If you can make a good design you may get extra points.**
8. **Take a look at the screenshots below as an example**

Code Editor

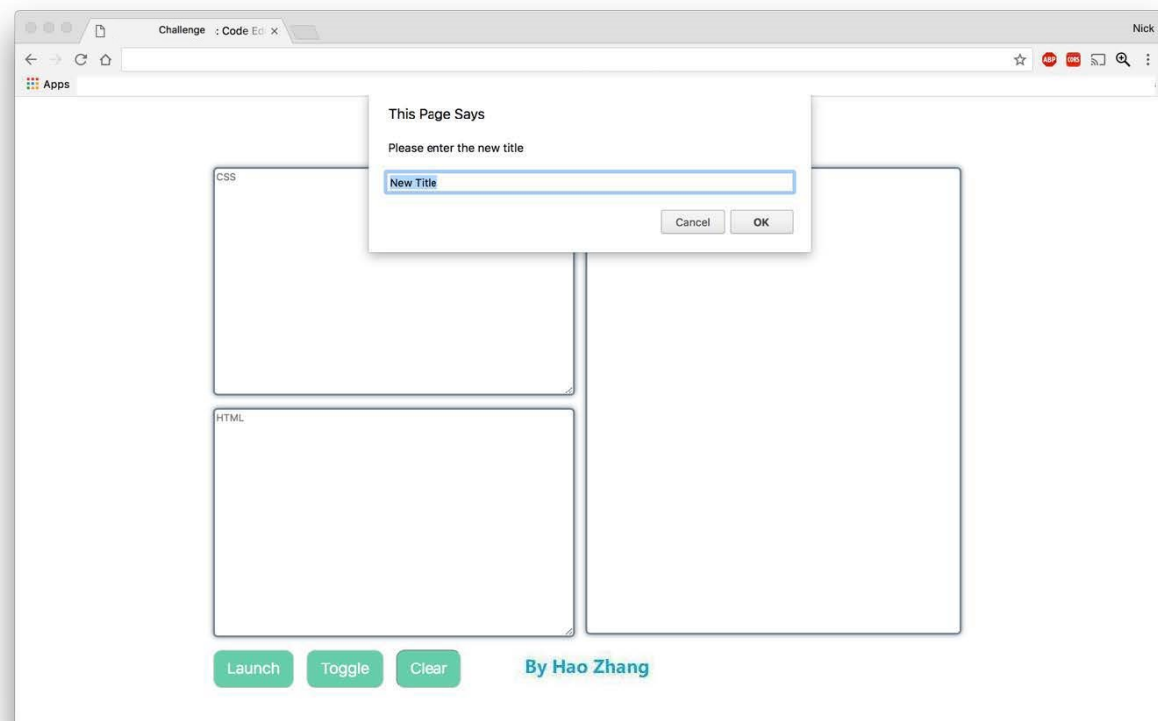
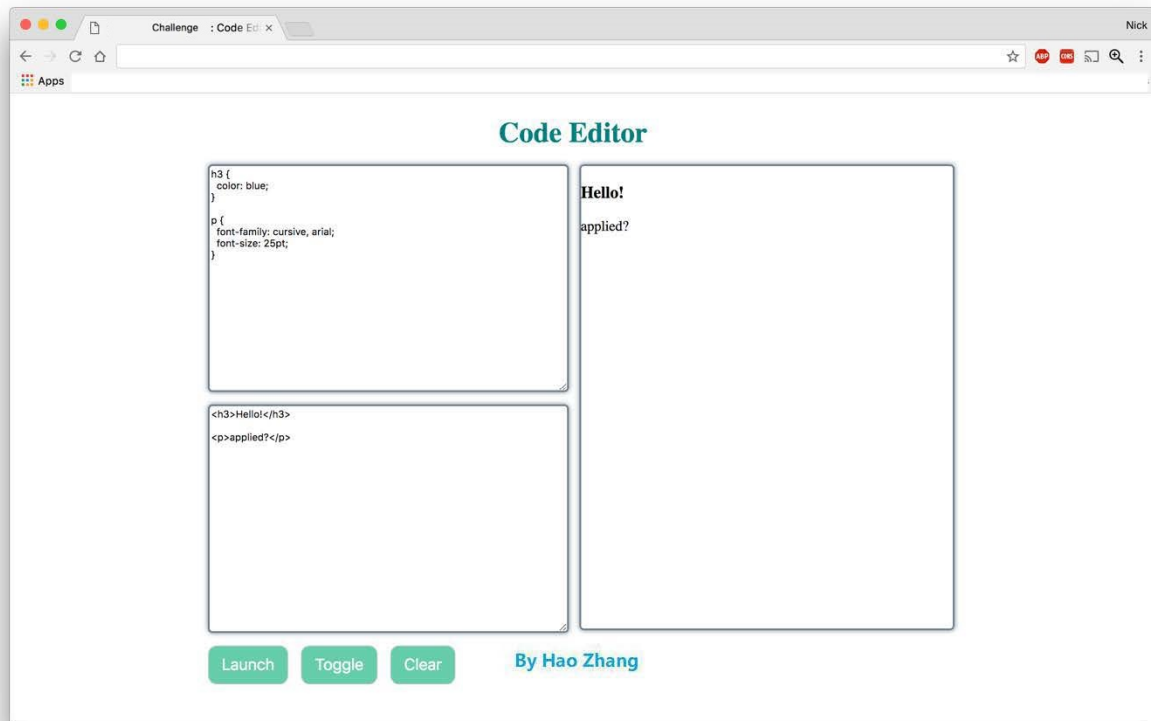
Screenshots:

Note: There may be some differences between these screenshots and the instructions above. If you find differences, then make sure you follow the instructors above. These are meant to be an example to help you but may not resemble the actual assignment.

Sixth Challenge



Code Editor



Code Editor

Sixth Challenge



The interface consists of three main panels. On the left, there are two stacked text editors: the top one is labeled 'CSS' and the bottom one is labeled 'HTML'. To the right of these is a large, empty rectangular area for the preview. Below the editors are three green buttons: 'Launch', 'Toggle', and 'Clear'. A black tooltip with white text points to the 'Toggle' button, stating: 'This button used to show the result of your html code'. At the bottom right, the text 'By Hao Zhang' is displayed.

CSS

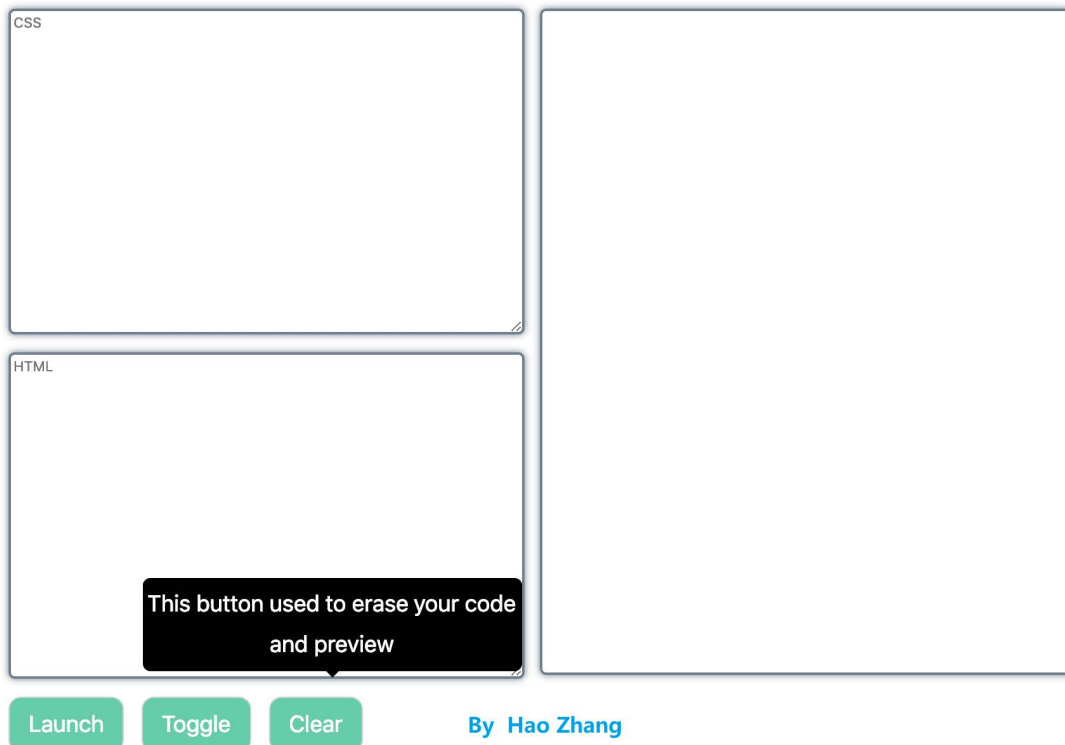
HTML

This button used to show the result of your html code

Launch Toggle Clear

By Hao Zhang

Sixth Challenge



This version of the interface is simplified. It features two stacked text editors on the left, labeled 'CSS' and 'HTML'. To the right is a large, empty rectangular area for the preview. Below the editors are three green buttons: 'Launch', 'Toggle', and 'Clear'. A black tooltip with white text points to the 'Clear' button, stating: 'This button used to erase your code and preview'. At the bottom right, the text 'By Hao Zhang' is displayed.

CSS

HTML

This button used to erase your code and preview

Launch Toggle Clear

By Hao Zhang

Code Editor

Sixth Challenge

CSS

HTML

This button used to turn on or off
your css code

Launch

Toggle

Clear

By Hao Zhang