

Challenge 2: Multi-Column Layout

This assignment is intended for you to understand multi-column layouts, floats, physical width and calculations, as well as styling your page properly using CSS techniques. 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.

Essential Requirement:

- The filename should be named "<Yourname>Columns.html". Replace yourname with your real name.
- Write and complete the following comment in your HTML document:

```
<!--  
    Name:  
    Name:  
    Date:  
    Challenge 2: Multi-Column Layout  
-->
```

- Must be a well-formed HTML5 document that uses CSS for the page design and layout
- **DO NOT use Inline CSS**

Requirements:

1. You can use the template provided on the assignment page to help you get started
2. Make sure your page contains **the 5 required tags** that were covered in class with an ".html" extension.
3. Before you submit, your **document must pass the validator** (<https://validator.w3.org/>)
4. On the page, create a container **wrapper div** (called the **Main Wrapper**) that will contain and perfectly fit 2 columns with two stacked rows in between them.
5. Inside the Main Wrapper div, at the top, create an **H2 including** this text "**CS/IT 2830 – Multi-Column Layout**", and make the H2 **center aligned**.
6. Create **3 columns (divs) inside the container div (Main Wrapper)**, sitting next to each other, like the news column example shown in class. The first column (div) will be placed on the left side under the **H2**. You will then need to make another **wrapper div** (called the

Challenge 2: Multi-Column Layout

Row Wrapper) which can hold two rows that will be stacked and placed in the middle of the left and right column, and you will need to make one last column (**div**) to be positioned on the right of those rows.

7. For the **left column**, you should put the definition of web development and make a citation to the source you copied it from in your source code.
 - a. The definition does not need to be specific but use a reliable source.
8. The **middle column**, in the **Top Row** of the row wrapper, will contain the physical width calculations for the **left column**.
 - a. You will have to calculate this using what you know about the CSS Box Model from lecture and/or this source: https://www.w3schools.com/css/css_boxmodel.asp
9. The **middle column**, in the **Bottom Row** of the row wrapper, will contain the physical width calculations for the **Right Column**.
10. The **Right Column** should provide the definition of physical width versus content width in your own words.
11. **Include the navbar you used in challenge one and two** and create a third link to this challenge at the top of your page. Be sure the navbar is horizontal and the buttons are adjacent ... not stacked ... as you should have done in challenge two.

Element Specifications:

1. **Main Wrapper div** should be:
 - a. **Centered** on the page (Do not use inline CSS)
 - b. Have a content width of 1090px
 - c. Have a 10-pixel **outset border** with a color of your choosing, other than the default
 - d. The **border radius** set other than the default
 - e. Contains 10-pixel **padding**
 - f. And a **margin** of 20 pixels.
2. **Row Wrapper div** (middle column) should be:
 - a. **438** pixels wide with a margin of 20 pixels.
 - b. Put both of your rows in this wrapper and stack them with a spacing of 30 pixels between them. (Hint: one of the rows in the row wrapper will need a specific margin)
3. **Left and Right columns** inside the **Main wrapper** must have:
 - a. 20 pixels of **padding**
 - b. 20 pixels of **margin**
 - c. A height of 400 pixels

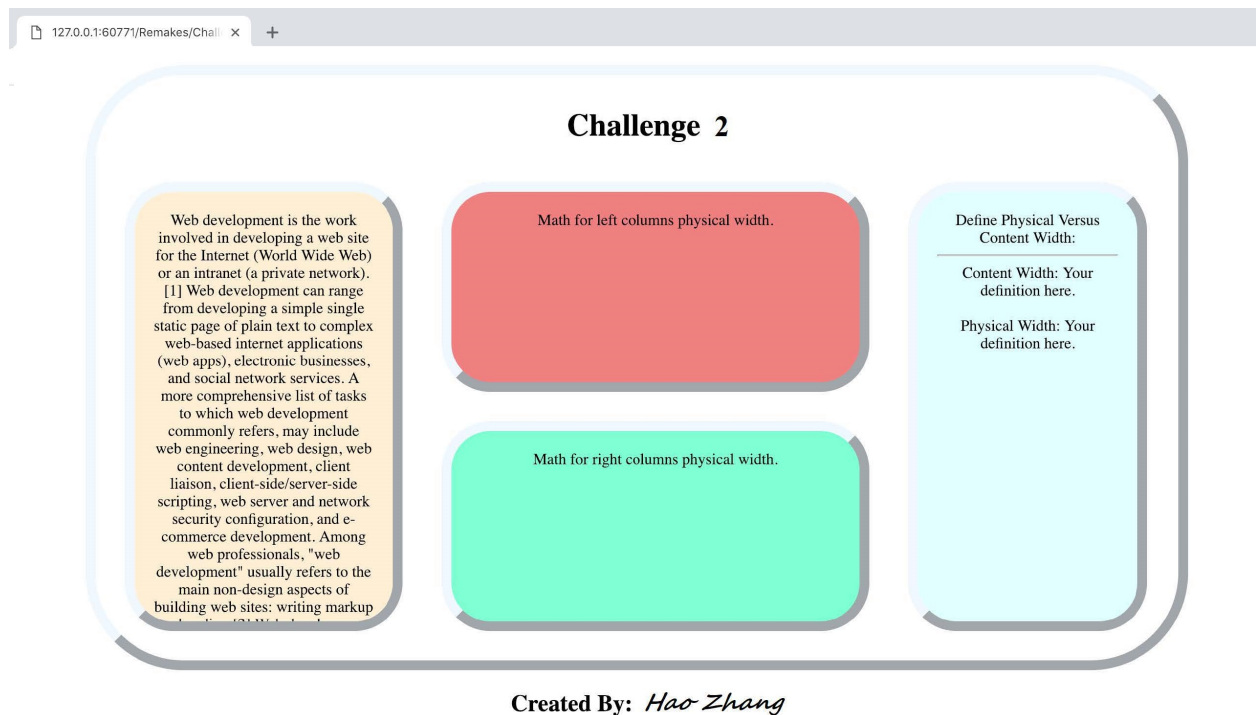
Challenge 2: Multi-Column Layout

- d. A **10-pixel outset border** with a color of your choosing, other than the default
 - e. A **border radius** of your choosing other than the default
- 4. The **content width** of the **left column** should be **225 pixels**.
- 5. The **content width** of both **rows in the Row Wrapper** should be **378 pixels**, with a **height** of **155 pixels**, **20 pixels** of **padding**, a **10-pixel outset border** with a color of your choosing other than default, and a **border-radius** of your choosing other than the default.
- 6. The **Left Column**, **Top Row** and **Bottom Row** in the Row Wrapper (middle column), and the **Right Column** should have a background color of your choosing. However, make sure you have a good UI design where the text is easy to read and the page looks professional. Points will be deducted for pages not having a good design.
- 7. The **3 columns** and rows inside the row wrapper should **fit perfectly** in the **main wrapper div**, no white spaces on each side, that has a **content width** of **1090px**.
 - a. You can test this by subtracting one pixel from your wrapper div's content width, if the columns move down to the next level, then you have the correct number. If they don't move down, then re-calculate. (Use the techniques shown in class).
- 8. You are not given the content width of the **right column** and you should use the remaining width of the main wrapper div to define it.
 - a. You will have to **calculate this width using** what you have learned about the **CSS Box Model** in class: https://www.w3schools.com/css/css_boxmodel.asp
- 9. The **three columns** should be **floated left**, so they will display in a row horizontally.
 - a. Do not use other techniques shown in class ... Use Float! There is a reason we are using float and other techniques will receive deductions.
- 10. Don't forget to use the **clear hack** to achieve the desired result.
 - a. Do not use tricks on this step like defining a height on the Main Wrapper Div in order to make it appear the clear is working. Make sure you implement the clear hack properly. These tricks will receive point deductions.
- 11. After the wrapper div, create an **H3 outside the main wrapper div (the main container)** that says, "Created By: (your name)". This H3 should be **centered** on the page.
 - a. This is used to determine if the clear is working properly. If the H3 is below the main container (the 3 columns), then most likely the clear is working properly, if it gets put towards the top of the page, or beside the other columns in the main container, then the clear is not working properly.

Challenge 2: Multi-Column Layout

- b. Remember to not use any tricks on this step like setting a margin-top on the H3 to push it down or by using the position attribute. These tricks will receive point deductions. Instead, make sure you use the clear properly.
12. Don't forget to update your navbar with a link to your new challenge on **ALL** pages.
13. A screenshot of what the final result should look like can be found below along with an opportunity to earn optional bonus points

Note: Roughly, here is how the final result should look. You will notice it does not include the navbar. **If there are any differences between the screenshot and the requirements, then follow the requirements.**



Note:

- The height of the columns should be the same, and the rows should not go below or above the height of the columns. (They should be flush).

Challenge 2: Multi-Column Layout

Design more:

- Explore CSS Animations
- For bonus points, find a way to make the main wrapper's background fill up like a glass of water with a color of your choosing. This does not need to happen infinitely. When the wrapper is filled with the color, it should stay that color while the page is still open.
- You are going to have to research this, but the reward may be worth the additional work
- If you decide not to explore CSS Animations then you will not receive a deduction, these bonus points can only help you and will not hurt your grade