

## **Project Requirements/Information**

Questionnaire Builder for FHIR Questionnaires

Develop a JavaScript client side editor (run in the browser) to develop questionnaires:

<https://www.hl7.org/fhir/questionnaire.html>

Consider using something like:

- <https://github.com/helsenorge/refero>
- <https://github.com/helsenorge/structor>
  - [https://static.helsenorge.no/static\\_skjemabygger/](https://static.helsenorge.no/static_skjemabygger/)
- <https://editorjs.io/>
- <https://github.com/kevinchappell/formBuilder>

Internal Ticket: <https://pm.mieweb.com/issues/106574>

Read the discussion about observations and Questionnaire Responses:

<https://chat.fhir.org/#narrow/streams/public/search/observation.20.2B.20questionnaireresponse>

## **January 14, 2024**

- Created a new project proposal for internship.
- Project Selected: Questionnaire Builder
- Link to Proposal:  MIE Summer '24 Internship New Project Proposal - Keerthi Bala...

## **May 16, 2024**

### **Initial Research (10 - 11 AM)**

- Researched FHIR Questionnaires and reviewed the provided Github repositories and documentation links. (provided in Project Requirements)

- Evaluated potential libraries and tools that could be used for the project.

## Javascript Learning (11AM - 2PM)

- Since, I don't have enough Javascript knowledge to proceed, I decided to go through some basic documentation and execute some basic commands using tutorials.
- I went through and executed basic commands on [JavaScript Introduction \(w3schools.com\)](#)
- I learnt basic syntax for javascript by doing 67 exercises.

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

```
<!DOCTYPE html>
<html>
<body>

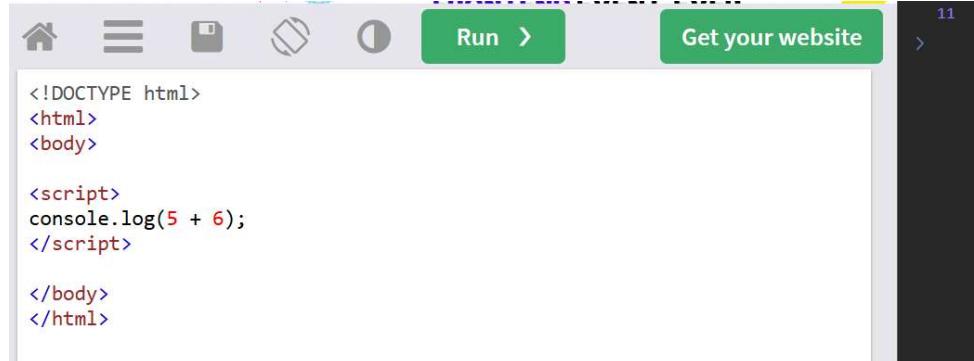
<h2>Demo External JavaScript</h2>
<p id="demo">A Paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
<p>This example links to "myScript.js".</p>
<p>(myFunction is stored in "myScript.js")</p>
<script src="myScript.js"></script>
</body>
</html>
```

### Demo External JavaScript

Paragraph changed.

[Try it](#)

This example links to "myScript.js".  
(myFunction is stored in "myScript.js")



The screenshot shows a browser window with a toolbar at the top. In the main area, there is a code editor containing the following HTML and JavaScript:

```
<!DOCTYPE html>
<html>
<body>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

Below the code editor, the browser's developer tools console displays the result of the `console.log` statement: "11".

### JavaScript Statements

A **JavaScript program** is a list of **statements** to be executed by a computer.

The value of z is 11.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Statements</h2>
<p>A <b>JavaScript program</b> is a list of <b>statements</b> to be executed by a computer.</p>
<p id="demo"></p>
<script>
let x, y, z; // Statement 1
x = 5; // Statement 2
y = 6; // Statement 3
z = x + y; // Statement 4
console.log(`The value of z is ${z}`);
document.getElementById("demo").innerHTML =
"The value of z is " + z + ".";
</script>
</body>
</html>
```

```

<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Expressions</h2>

<p>Expressions compute to values.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "John" + " " + "Doe";
</script>

</body>
</html>

```

The screenshot shows a browser-based code editor interface. On the left, the code for 'JavaScript Variables' is displayed:

```

<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Variables</h1>

<p>In this example, price1, price2, and total are variables.</p>

<p id="demo"></p>

<script>
const price1 = 5;
const price2 = 6;
let total = price1 + price2;
price1 = price1+1;
document.getElementById("demo").innerHTML =
"The total is: " + total;

document.getElementById("demo").innerHTML =
"The price1 value does not change: " + price1;

</script>
</body>
</html>

```

In the center, the page title is 'JavaScript Variables'. Below it, a paragraph states: 'In this example, price1, price2, and total are variables.' To the right, a tooltip provides information about a runtime error:

- Uncaught TypeError: Assignment to constant variable.
- at <anonymous>:5:8
- at submittryif (<tryit.asp?filename=t\_ables\_total:85:127)
- at HTMLButtonElement.onclick (<tryit.asp?filename=t\_ables\_total:753:123)

[NEW] Explain Console errors by using Copilot in Edge: click ⚡ to explain an error. Learn more

Don't show again

## Exercise:

Use the correct **assignment operator** that will result in `x` being 50 (same as `x = x * y`).

```

x = 10;
y = 5;
x *= y;

```

## Exercise:

Create an object called person with name = John, age = 50.  
Then, access the object to alert("John is 50").

```
const person = {  
    name: "John", age: 50  
};  
alert(person.name + " is " + person.age);
```

## Exercise:

When the button is clicked, the function "myFunction" should be executed.

```
<button onclick="myFunction()">Click me.</button>
```

## Exercise:

Use the correct String method to replace the word "Hello" with the word "Welcome".

```
let txt = "Hello World";  
txt = txt.replace("Hello", "Welcome");
```

## Exercise:

Change the first item of cars to "Ford".

```
const cars = ["Volvo", "Jeep", "Mercedes"];  
cars[0] = "Ford";
```

# Exercise:

Use the correct Array method to remove the last item of the `fruits` array.

```
const fruits = ["Banana", "Orange", "Apple"];
fruits.pop();
```

Completed 33 of 67 Exercises:

- JS Strings ✓
- JS String Methods ✓
- JS Arrays ✓
- JS Array Methods

# Exercise:

Use the correct Array method to add "Kiwi" to the `fruits` array.

```
const fruits = ["Banana", "Orange", "Apple"];
fruits.push("Kiwi");
```

Completed 36 of 67 Exercises:

- JS String Methods ✓
- JS Arrays ✓
- JS Array Methods ✓
- JS Array Sort ✓

# Exercise:

Create a Date object and alert the current date and time.

```
const d = new Date();
alert(d);
```

Completed 42 of 67 Exercises:

- JS Array Sort ✓
- JS Dates ✓
- JS Math

✓ Exercise 1

✓ Exercise 2

Exercise 3

# Exercise:

Use the correct Math method to round a number to the nearest integer.

```
let x = Math.round(5.3);
```

Completed 47 of 67 Exercises:

- JS Math ✓
- JS Comparisons

✓ Exercise 1

✓ Exercise 2

✓ Exercise 3

# Exercise:

Choose the correct conditional (ternary) operator to alert "Too young" if age is less than 18, otherwise alert "Old enough".

```
var age = n;
var voteable = (age < 18) ? "Too young" : "Old enough";
alert(voteable);
```

## Exercise:

Fix the if statement to alert "Hello World" if `x` is greater than `y`, otherwise alert "Goodbye".

```
if (x > y) {  
    alert("Hello World");  
} else {  
    alert("Goodbye");  
}
```

Completed 59 of 67 Exercises:

- JS For Loops ✓
- JS While Loops ✓
- JS Break Loops ✓
- JS HTML DOM
- ✓ Exercise 1
- Exercise 2

## Exercise:

Use the `getElementsByTagName` method to find the *first* `<p>` element, and change its text to "Hello".

```
<p id="demo"></p>  
  
<script>  
document.getElementsByTagName("p")[0].innerHTML = "Hello";  
</script>
```

Completed 63 of 67 Exercises:

- ✓ Exercise 1
- ✓ Exercise 2
- ✓ Exercise 3
- ✓ Exercise 4
- ✓ Exercise 5
- Exercise 6

## Exercise:

Change the text color of the `<p>` element to "red".

```
<p id="demo"></p>  
  
<script>  
document.getElementById("demo").style.color = "red";  
</script>
```

# Congratulations!

You have finished all 67 JS exercises.

Share your score:

(Lunch break 2PM - 3PM)

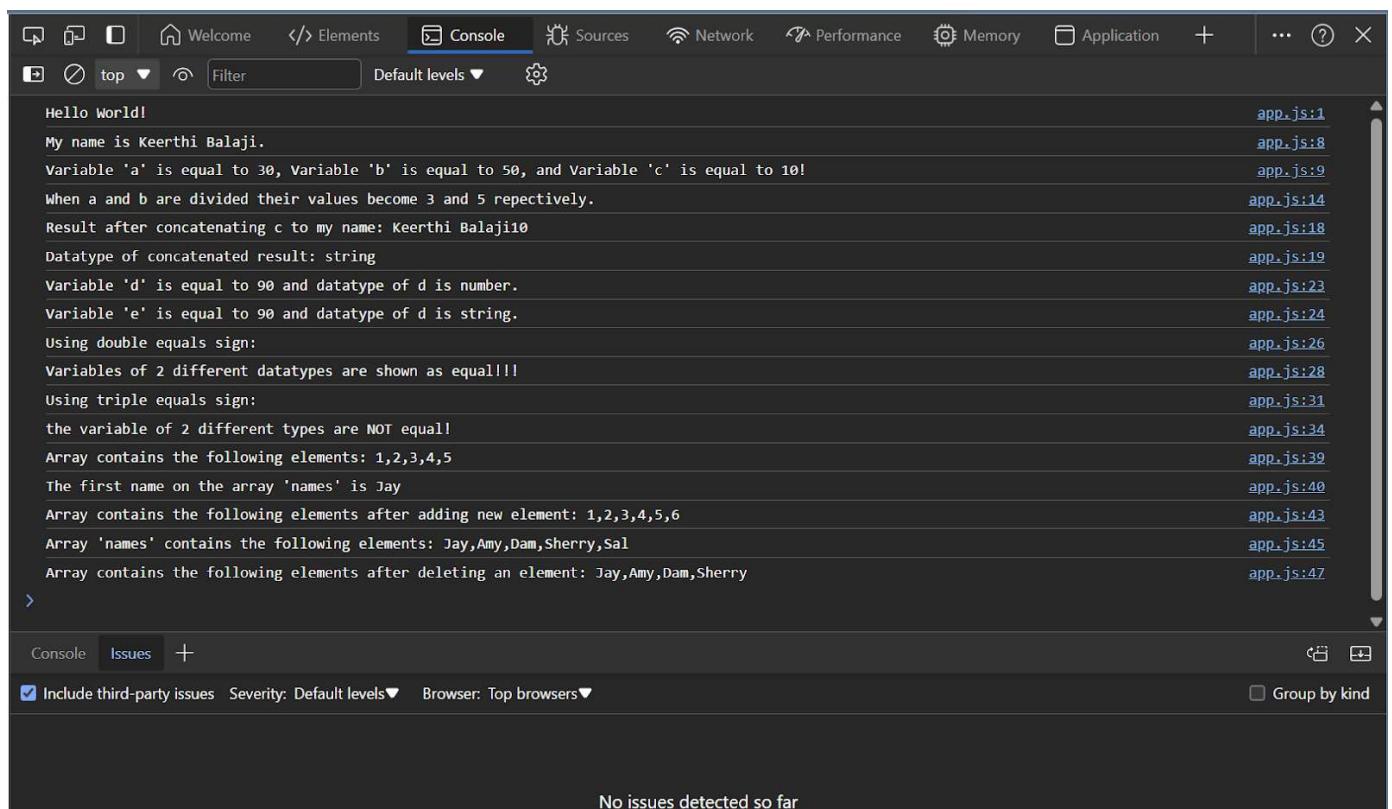
## JS Tutorial Videos (3PM - 5PM)

- Then, after some research, I decided to watch the playlist “[Javascript Tutorial For Beginners](#)” by youtuber <https://www.youtube.com/@developedbbyed>. I chose this because he goes over some of the basics. Which gives me a good start with a good foundation.
- He went over basics like variables, strings, data types, concatenation, if-else loops, for loops, while loops, arrays in javascript.
- Downloaded and installed Visual Code on my system.
- Practiced the concepts alongside the tutorials by writing and executing code in Visual Studio Code.

While watching tutorials:-

- Executed simple commands using visual studio code.
- Code: [JS Basic Tutorials](#)

Output:-



The screenshot shows the Visual Studio Code interface with the 'Console' tab selected. The terminal window displays a series of JavaScript console logs. The logs include:

- Hello World!
- My name is Keerthi Balaji.
- Variable 'a' is equal to 30, Variable 'b' is equal to 50, and Variable 'c' is equal to 10!
- When a and b are divided their values become 3 and 5 respectively.
- Result after concatenating c to my name: Keerthi Balaji10
- Datatype of concatenated result: string
- Variable 'd' is equal to 90 and datatype of d is number.
- Variable 'e' is equal to 90 and datatype of d is string.
- Using double equals sign:
- Variables of 2 different datatypes are shown as equal!!!
- Using triple equals sign:
- the variable of 2 different types are NOT equal!
- Array contains the following elements: 1,2,3,4,5
- The first name on the array 'names' is Jay
- Array contains the following elements after adding new element: 1,2,3,4,5,6
- Array 'names' contains the following elements: Jay,Amy,Dam,Sherry,Sal
- Array contains the following elements after deleting an element: Jay,Amy,Dam,Sherry

At the bottom of the terminal, there are filter and grouping options: 'Include third-party issues', 'Severity: Default levels', 'Browser: Top browsers', and 'Group by kind'. A message at the bottom right states 'No issues detected so far'.

The screenshot shows a browser's developer tools console tab. The output area displays the following log entries:

```
Array 'names' contains the following elements: Jay,Amy,Dam,Sherry,Sai app.js:42
Array contains the following elements after deleting an element: Jay,Amy,Dam,Sherry app.js:47
After removing first element in number array: 2,3,4,5,6 app.js:50
After adding a new first element in names array: Carrie,Jay,Amy,Dam,Sherry app.js:53
Object 'user' created. app.js:61
User =
  ▶ {name: 'Keerthi', age: 24, purchases: Array(4), sayUser: f, sayAge: f} app.js:62
User's age is 24 app.js:65
  Jake
  Adam
  Emily
  Natalie
  Morgan
Morgan is on my list! app.js:75
Counting to 10 app.js:82
  1
  2
  3
  4
  5
  6
  7
  8
  9
  10
```

After click:



### Demo Mini Project (5PM - 8PM)

- Created a demo project “Rock paper scissors” to apply learned JavaScript skills.
- Set up the project structure and did basic HTML and CSS styling.
- While creating the project I went through documentation for all commands used.
- The documentation page I went through for doing the HTML structure:  
[https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML/Getting\\_started#anatomy\\_of\\_an\\_html\\_document](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started#anatomy_of_an_html_document)

Scores for player and computer displayed along with play button (“go”). The player (user) will be able to select the option between rock, paper or scissors by clicking on the respective buttons.

The screenshot shows a web browser window with the URL 127.0.0.1:5501/index.html. The page displays a game interface. At the top, it says "Player" with a score of 0 and "Computer" with a score of 0. Below that, the text "Rock! Paper! Scissors!" is displayed. A "Go!" button is present. The next section, "Choose an option", contains three buttons labeled "rock", "paper", and "scissors". Two hand icons are shown below this section, one on each side.

- Added some basic styling to elements, making it look cleaner.
- Alignment of text, font styles, spacing and background color.

Documentation I went through to understand: [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)

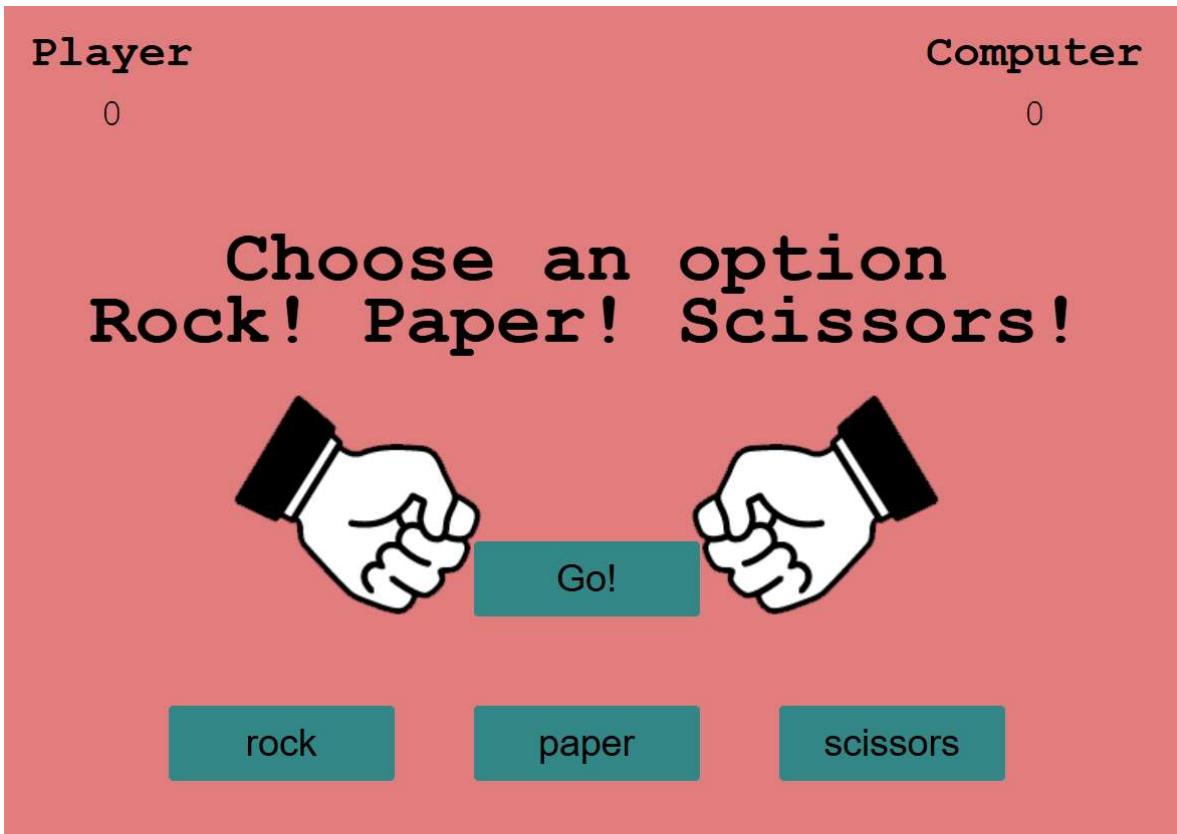


Rotated the player hand to face the computer hand.

[https://www.w3schools.com/cssref/css3\\_pr\\_transform.php](https://www.w3schools.com/cssref/css3_pr_transform.php)

And added styling for buttons to make it look better.

[https://www.w3schools.com/css/css\\_margin.asp](https://www.w3schools.com/css/css_margin.asp)



**May 17, 2024**

**Mini Project Completion (10 AM IST to 2 PM IST)**

- Finished the styling of the Rock-Paper-Scissors project.
- Added JavaScript to handle user interactions, game logic, and score updates.
- Tested and debugged the project to ensure correct functionality.

Added transition on the intro screen and added specific styling for each element on the intro screen.

Documentation I went through for each element styling:

Transition: [https://www.w3schools.com/css/css3\\_transitions.asp](https://www.w3schools.com/css/css3_transitions.asp)

Display property: [https://www.w3schools.com/cssref/pr\\_class\\_display.php](https://www.w3schools.com/cssref/pr_class_display.php)

Flex: [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)

Buttons:

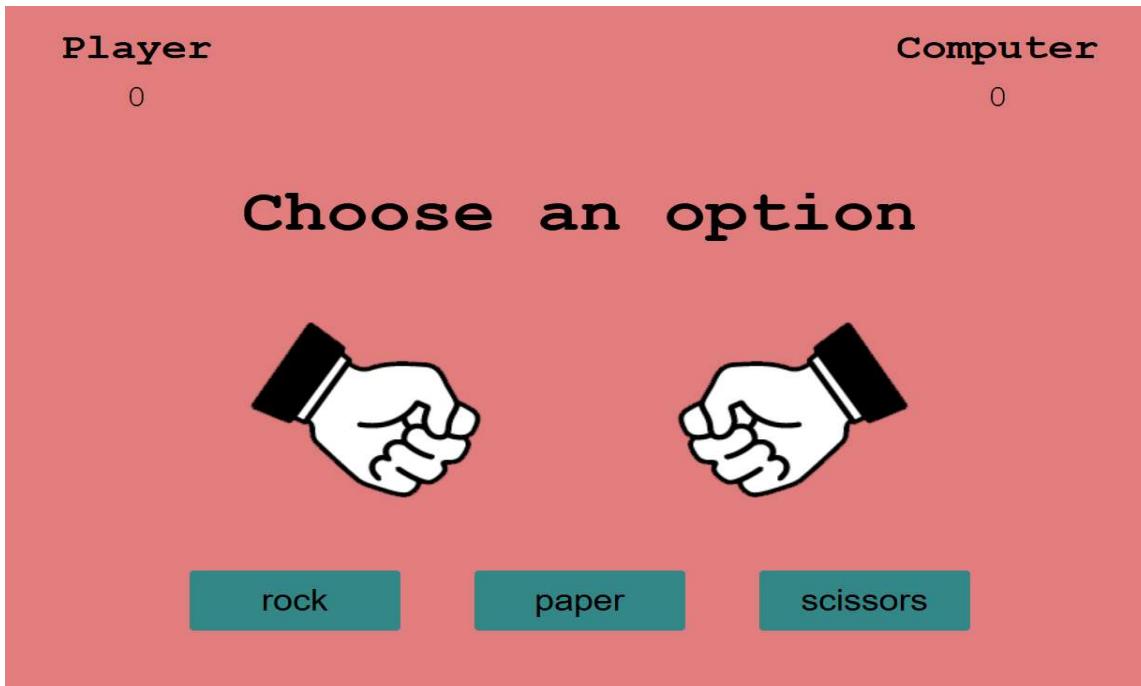
[https://www.w3schools.com/cssref/css3\\_pr\\_border-radius.php](https://www.w3schools.com/cssref/css3_pr_border-radius.php)

[https://www.w3schools.com/cssref/css3\\_pr\\_align-items.php](https://www.w3schools.com/cssref/css3_pr_align-items.php)



Screen 1

- Created animations for the hands
- Went through the Documentation for animations.
- Used transform for the animation:  
[https://www.w3schools.com/cssref/css3\\_pr\\_transform.php](https://www.w3schools.com/cssref/css3_pr_transform.php)
- Created some delay and had the hands go back to rock after each animation.  
Documentation: [https://www.w3schools.com/cssref/css3\\_pr\\_transition-delay.php](https://www.w3schools.com/cssref/css3_pr_transition-delay.php),  
[https://www.w3schools.com/cssref/css3\\_pr\\_transform.php](https://www.w3schools.com/cssref/css3_pr_transform.php),  
[https://www.w3schools.com/cssref/css3\\_pr\\_pointer-events.php](https://www.w3schools.com/cssref/css3_pr_pointer-events.php),  
[https://www.w3schools.com/cssref/css3\\_pr\\_opacity.php](https://www.w3schools.com/cssref/css3_pr_opacity.php)



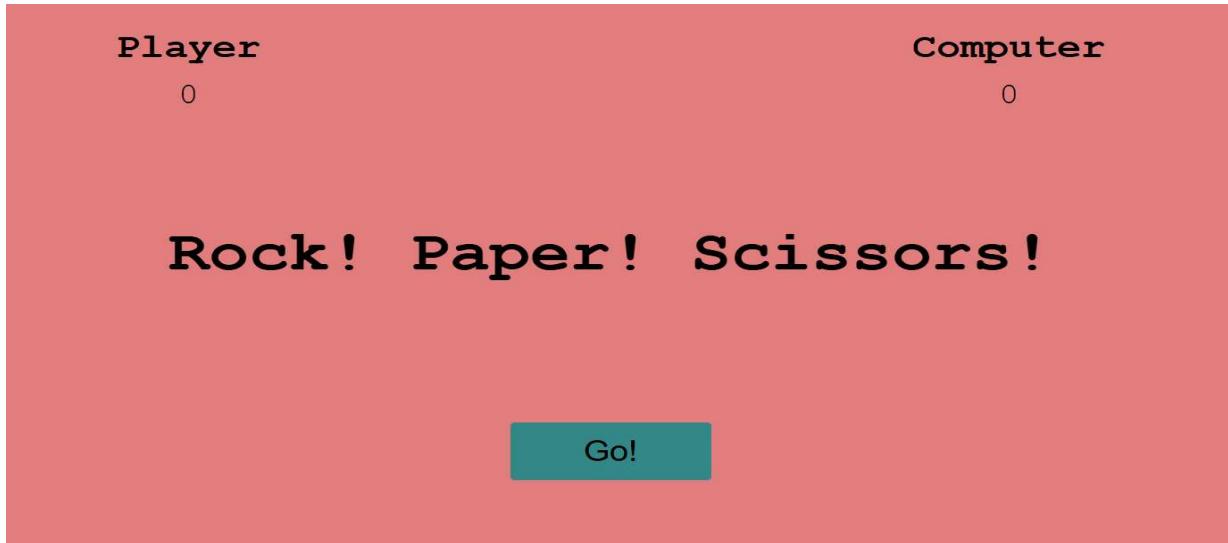
Screen 2

- Added in the JavaScript for programming the game.
- The user chooses between the three choices (rock, paper, scissors). The left side hand changes the icon accordingly. (player hand)
- The computer hand on the right is generated randomly.
- The conditions were programmed according to game rules.
- Both the scores are updated as per the game conditions.
- The winner of each instance of the game is declared immediately and the score is updated.

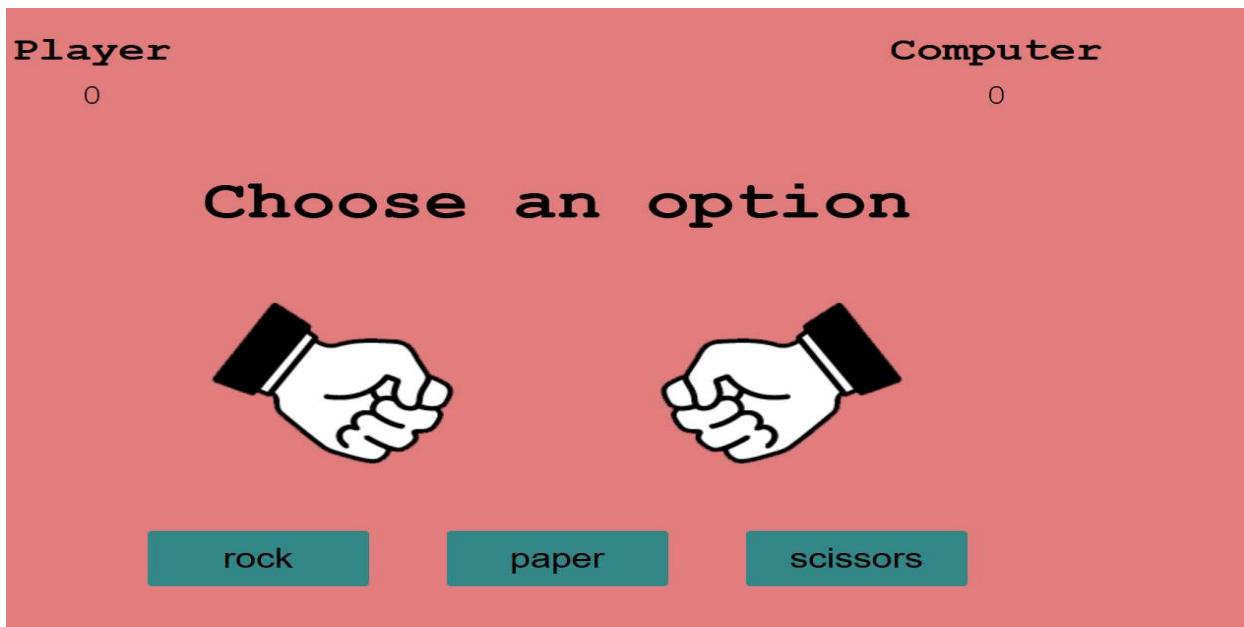
JS Documentation:

[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/Variables](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables)  
[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/Math](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Math)  
[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building\\_blocks/Functions](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Functions)  
[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building\\_blocks/Events](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events)  
[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/Arrays](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Arrays)  
[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building\\_blocks/conditionals](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/conditionals)  
[https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)  
[https://www.w3schools.com/jsref/prop\\_node\\_textcontent.asp](https://www.w3schools.com/jsref/prop_node_textcontent.asp)

The first screen is displayed.



Once “go!” button is clicked, 2nd screen is displayed. And now, the player can select the options and the score updates.



Text is displayed indicating who won and score is updated accordingly.

**Player**

0

**Computer**

1

**Computer Won!**



rock

paper

scissors

**Player**

1

**Computer**

4

**Player Won!**



rock

paper

scissors

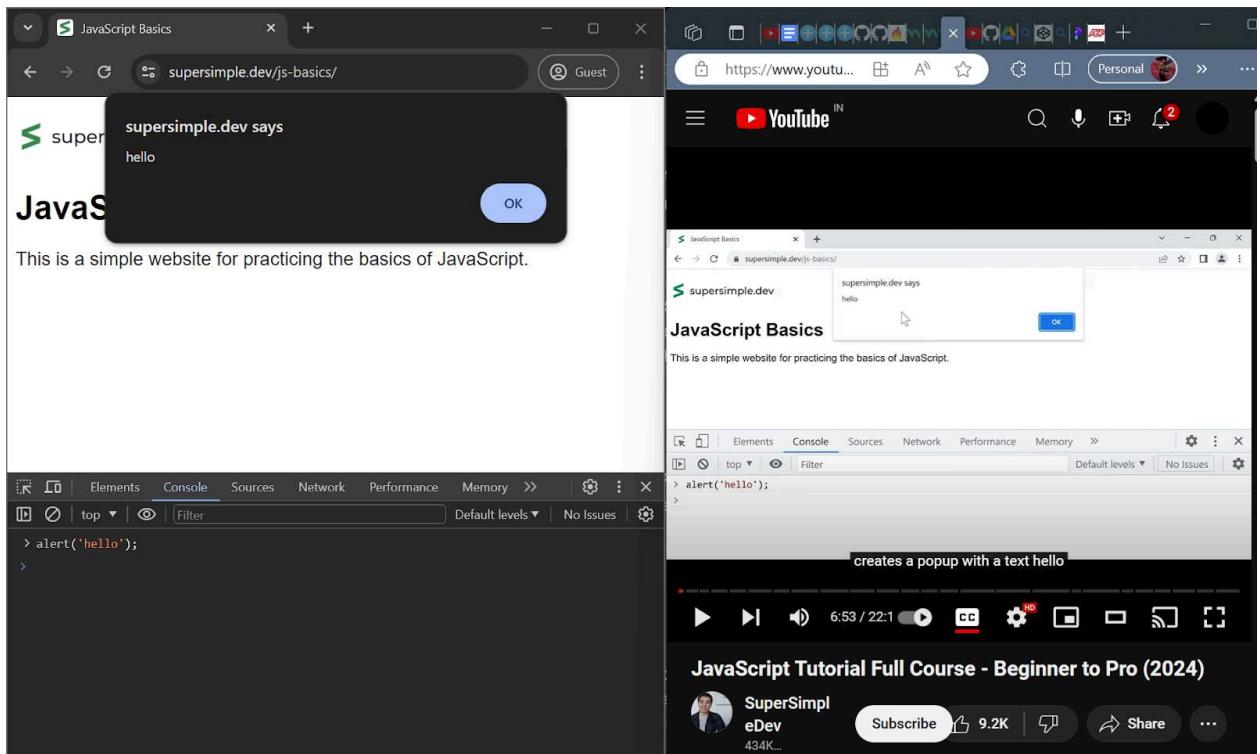
- The demo project helped me understand basic javascript.
- Code: [rock paper scissors](#)

## (Lunch break 2PM - 3PM)

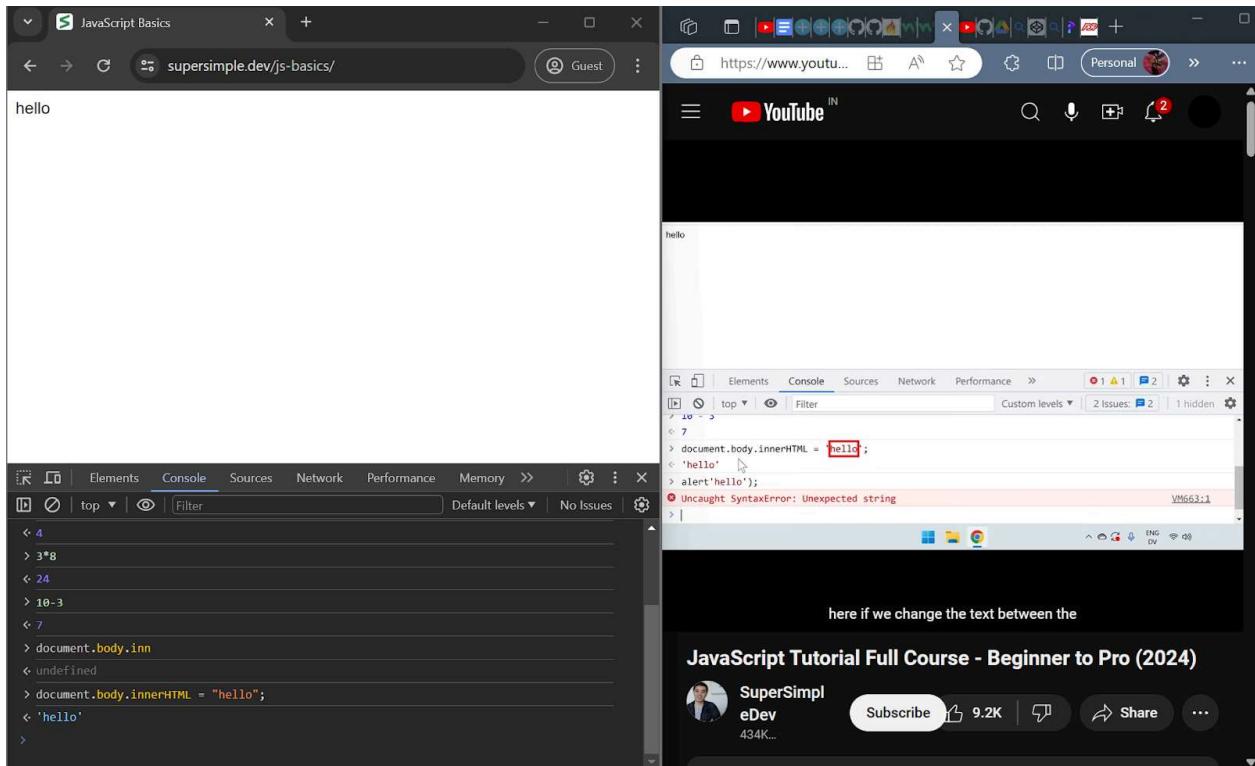
## JS Tutorial Video Course (3PM - 6PM)

- After the mini project, I was confident in basic JS but wasn't confident enough to start the questionnaire builder so I researched and found a more recent (May 2024) tutorial course with exercises that will provide me a more thorough understanding of JavaScript.
- The video course is linked [here](#).
- The video started with the basics and as done previously, I simultaneously practiced on my own system.
- This tutorial was slightly different in that he started programming within the Chrome browser in the console.

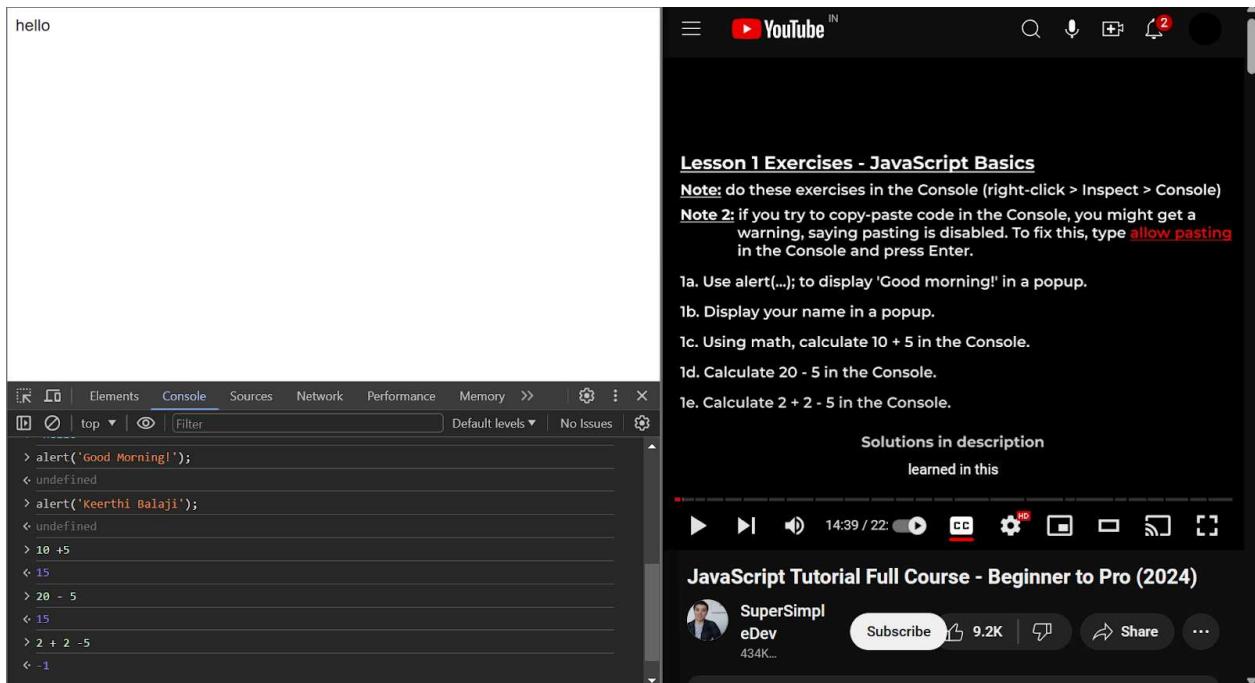
Printing using console through browser dev tools.



Using innerHTML. [https://www.w3schools.com/jsref/prop\\_html\\_innerhtml.asp](https://www.w3schools.com/jsref/prop_html_innerhtml.asp)



Alert in JS. Math operations in JS. The tutor went over the basics and gave exercises which were useful for practice.



The screenshot shows a browser window with two tabs open. The left tab is titled 'JavaScript Basics' and displays a challenge exercise: '1f. Use document.body.innerHTML = ...; to display 'Good morning!' on the web page.' The right tab is a YouTube video titled 'JavaScript Tutorial Full Course - Beginner to Pro (2024)' by SuperSimpleDev. The video player shows the first few frames of the tutorial, which includes a list of challenges like '1g. Display your name on the web page.' and '1h. You order a T-shirt for \$10, socks for \$8, and dinner plates for \$20. Use JavaScript to calculate the total cost of your order.' Below the video player, there's a section for 'Solutions in description' and a link to 'these exercises in the video'. The browser's developer tools are visible at the bottom, showing the console output of the challenge code.

## Operator precedence in JS

The screenshot shows a browser window with a single tab titled 'Checkout' from supersimple.dev. The page displays an 'Order Summary' for 3 items with a total of \$40.62. Below the summary, there's a checkbox labeled 'Use PayPal'. The browser's developer tools are open at the bottom, showing the console output of a complex arithmetic expression involving multiple operators:  $20.95 + 7.99 * 2$ . The output shows the step-by-step evaluation: 36.93, 36.93 \* 0.1, 3.693, and finally  $(20.95 + 7.99 * 2) * 0.1$  resulting in 3.693.

## Issues with float values and how to handle them.

The screenshot shows a browser window with two tabs. The left tab is an Amazon checkout page for two items, totaling \$31.83. The right tab is a YouTube video titled "JavaScript Tutorial Full Course - Beginner to Pro (2024)" by SuperSimpleeDev. A black box on the video screen contains text about floating-point issues:

**Calculations with floats are sometimes inaccurate.**  
**When working with money**  
**1. Do the calculation in cents**  
**2. Convert back to dollars**

The browser's developer tools console is visible at the bottom, showing JavaScript calculations related to the floating-point issue.

## Exercises:

The screenshot shows a browser window with two tabs. The left tab is an Amazon checkout page for two items, totaling \$31.83. The right tab is a YouTube video titled "Lesson 2 Exercises - Numbers and Math" by SuperSimpleeDev. The video description includes exercises:

**Note:** do these exercises in the Console (right-click > Inspect > Console)

- At a restaurant, you order 1 soup for \$10, 3 burgers for \$8 each, and 1 ice cream for \$5. Use JavaScript to calculate the cost of the order.
- You're at a restaurant with 2 friends (3 people in total) and make the same order as 2a. Calculate how much each person pays.
- Calculate the total cost of a toaster (\$18.50) and 2 shirts (\$7.50 each).
- Calculate a 10% tax for the total in exercise 2c.
- Calculate a 20% tax for the total in 2c (remember that 1% = 1 / 100, so 20% = 20 / 100 = 0.2).

Solutions in description  
and math

The browser's developer tools console is visible at the bottom, showing JavaScript calculations for the exercises.

## Challenge Exercises:

String and Using the tilde for printing

The screenshot shows a browser window with two tabs open. The left tab is a custom-built checkout page from [supersimple.dev](https://supersimple.dev/projects/amazon/checkout), displaying an order summary for two items: an Intermediate Size Basketball and a FREE Shipping option. The total order amount is \$31.83. A modal window titled "supersimple.dev says" displays the message "Total cost: \$8". The right tab is a YouTube video player for a tutorial titled "JavaScript Tutorial Full Course - Beginner to Pro (2024)". The video has 182,266 views and was uploaded on May 9, 2024. The video content discusses concatenation and provides exercises for string manipulation.

## Exercises:

This screenshot continues the theme of the previous one, showing the same browser setup. The left tab shows the same custom checkout page with the modal "supersimple.dev says" now displaying "Total cost: \$8". The right tab is still the YouTube video player for the JavaScript tutorial. The video has reached the "Lesson 3 Exercises - Strings" section. It provides several exercises related to string manipulation, such as creating strings, concatenating them, and calculating totals. The video also encourages viewers to practice working with strings.

**Lesson 3:**

The image shows two screenshots of a browser session illustrating a challenge exercise from a JavaScript tutorial.

**Screenshot 1 (Top Left):** A screenshot of a browser window showing an Amazon checkout page. A modal dialog box displays the text "supersimple.dev says" followed by "Total cost: \$8.94" and "Thank you, come again!". An "OK" button is at the bottom right of the dialog. Below the dialog, the checkout summary table shows the following details:

Order total:	\$31.83
Shipping & handling:	\$0.00
Total before tax:	\$28.94
Estimated tax (10%):	\$2.89

At the bottom, there is a "Use PayPal" checkbox. The browser's developer tools console is open, showing the following code:

```

<'My name is: Keerthi'
> `Total cost: ${5+3}`
<'Total cost: $8'
> alert(`Total cost: ${5+3}`);
<undefined
> `Total cost: ${((599+295)/100)}`
<'Total cost: $8.94'
> alert(`Total cost: ${((599+295)/100)}
    Thank you, come again!`);
>

```

**Screenshot 2 (Top Right):** A screenshot of a YouTube video player titled "JavaScript Tutorial Full Course - Beginner to Pro (2024)". The video is at 1:07:22 / 2. The video content discusses calculating the total cost of an order and displaying it in a modal. The video player interface includes a progress bar, volume control, and sharing options.

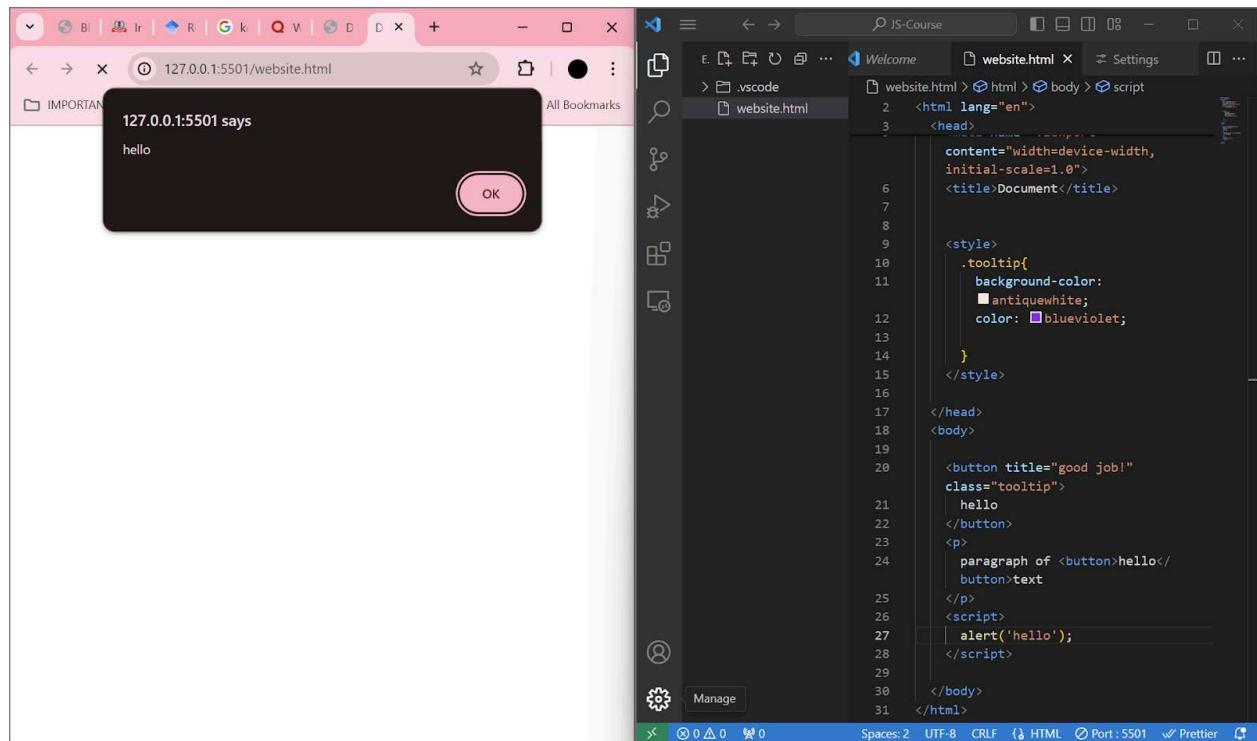
**Screenshot 3 (Bottom Left):** A screenshot of a browser window showing the same Amazon checkout page as Screenshot 1. The delivery date dropdown is open, showing three options: "Monday, June 3" (selected), "Tuesday, May 28" (\$4.99 - Shipping), and "Friday, May 24" (\$9.99 - Shipping). The "Delivery date: Monday, June 3" is highlighted in green.

**Screenshot 4 (Bottom Right):** A screenshot of the same YouTube video as Screenshot 2. The video content continues with challenge exercises related to the Amazon project. It lists several steps (3g, 3h, 3i, 3j) and provides a code snippet for calculating the total cost and displaying it in a modal. The video player interface is identical to Screenshot 2.

Basic HTML and CSS reviewed. I followed the tutorial because he is very detailed and goes over syntax and special rules.

Uses Visual Studio Code.

Goes over basic components in HTML and how to style them.



The screenshot shows a split interface of Visual Studio Code. On the left is a browser window displaying a local file at 127.0.0.1:5501/website.html. The page content is "127.0.0.1:5501 says" and "hello" with an "OK" button. On the right is the code editor with the file website.html open. The code includes HTML, CSS, and JavaScript:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
5     <title>Document</title>
6
7     <style>
8       .tooltip{
9         background-color: #antiquewhite;
10        color: #blueviolet;
11      }
12    </style>
13
14  </head>
15  <body>
16
17    <button title="good job!" class="tooltip">
18      hello
19    </button>
20    <p>
21      paragraph of <button>hello</button>text
22    </p>
23
24    <script>
25      alert('hello');
26    </script>
27
28  </body>
29
30</html>
```

A screenshot of a browser window and a VS Code editor side-by-side. The browser shows a simple product page for 'Adults Plain Cotton T-shirt' with a price of \$7.99 and two buttons: 'Add to cart' and 'Buy now'. The VS Code editor shows the corresponding HTML and CSS code. The HTML includes a style section with class definitions for 'add-to-cart-button' (yellow background) and 'buy-now-button' (orange background). The CSS uses hex codes for colors.

```
<!DOCTYPE html>
<html>
<head>
<style>
.add-to-cart-button {
background-color: #yellow;
}
.buy-now-button {
background-color: #orange;
}
</style>
</head>
<body>
<p>Adults Plain Cotton T-shirt</p>
<p>Price: $7.99</p>
<!-- I added some extra spaces to make the code easier to read. --&gt;
&lt;button class="add-to-cart-button" onclick="alert('Added');"&gt;Add to cart&lt;/button&gt;
&lt;button class="buy-now-button" onclick="alert('Loading...'); alert('Purchased');"&gt;Buy now&lt;/button&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

## Exercises:

The exercises took me longer time than I expected.

A screenshot of a browser window and a VS Code editor side-by-side. The browser shows a button labeled 'Click'. The VS Code editor shows the source code for 'exercise.html', which contains a single button element with the text 'Click'. A YouTube video player is overlaid at the bottom of the screen, showing a lesson titled 'Lesson 4 Exercises - HTML CSS Review, console.log'. The video player has a note: 'Note: to do these exercises, create an HTML file (like 4a.html, 4b.html, ...)' and a task: '4a. Create a button with the text 'Click' inside.' with a 'Click' button.

```
<button>Click</button>
```

The screenshot shows a code editor (VS Code) and a browser window. The code editor displays the file `exercise.html` with the following content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width,
        initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <button>Keerthi</button>
    <button>Rice</button>
</body>
</html>
```

The browser window shows the rendered HTML at `127.0.0.1:5501/exercise.html`. It displays two buttons: "Keerthi" and "Rice". Below the browser is a YouTube video player showing a lesson titled "Lesson 4 Exercises - HTML CSS Review, console.log". The video player has a note: "Note: to do these exercises, create an HTML file (like 4a.html, 4b.html, ...)" and two tasks: "4a. Create a button with the text 'Click' inside." and "4b. Create 2 buttons. 1 with your name and 1 with your favorite food." with input fields for "Click", "Simon", and "chocolate".

The screenshot shows a code editor (VS Code) and a browser window. The code editor displays the file `exercise.html` with the following content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width,
        initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <button>Keerthi</button>
    <button>Rice</button>
    <p>Hello, world!</p>
    <p></p>
    <p>Today, I practiced Javascript.</p>
</body>
</html>
```

The browser window shows the rendered HTML at `127.0.0.1:5501/exercise.html`. It displays the text "Hello, world!" and "Today, I practiced Javascript.". Below the browser is a YouTube video player showing a lesson titled "Lesson 4 Exercises - HTML CSS Review, console.log". The video player has a note: "4c. Create a paragraph with the text 'Hello, world!' inside." and "4d. Continuing from exercise 4c, create a paragraph below the previous paragraph and write something you did today." with input fields for "Hello, world!" and "Hello, world!". At the bottom, there is a link "Solutions + more exercises in description" with three downward arrows.

Added a bit of JS script with the html and CSS for the challenge exercises.

JS-Course

website.html exercise.html 4i.html Settings

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width,
initial-scale=1.0">
6     <title>Document</title>
7
8     <style>
9       .cart{
10         background-color: yellow;
11     }
12     .buy {
13       background-color: orange;
14     }
15   </style>
16
17 </head>
18 <body>
19   <p>Adults Plain Cotton T-shirt</p>
20   <p></p>
21   <p>Price: $7.99</p>
22   <button class="cart" onclick="alert('Added');">Add to
cart</button>
23   <button class="buy" onclick="alert('Loading...'); alert
('Purchased');">Buy Now</button>
24 </body>
25 </html>

```

IMPORTANT Content Courses Member Home All Bookmarks

127.0.0.1:5501/4i.html

Adults Plain Cotton T-shirt

Price: \$7.99

Add to cart Buy Now

YouTube IN

**Challenge Exercises**

**Setup:** create a new file 4i.html for exercises 4i and 4j.

4i. Using HTML and CSS, try to copy the design on the right.  
(Make sure to follow the HTML structure.)

4j. Continuing from 4i, add onclick="..." attributes to the two buttons:

- When clicking 'Add to cart', create a popup with the text 'Added'.
- When clicking 'Buy now', display the message 'Loading...' in the Console, and then create a popup with the text 'Purchased'.

Adults Plain Cotton T-shirt

Price: \$7.99

Add to cart Buy now

This page says

Added

OK

The screenshot shows a code editor on the left displaying an HTML file named 4i.html. The code includes a CSS block for button classes .cart and .buy. A browser window on the right shows a local host page at 127.0.0.1:5501/4i.html. A modal dialog box is open with the text "127.0.0.1:5501 says" and two buttons: "Added" and "Purchased". Below the dialog, the text "Price: \$7.99" and an "Add to cart" button are visible. An "OK" button is in the top right corner of the dialog. At the bottom of the browser window, there's a message: "design on the right. (Make sure to follow the HTML structure.)" followed by "Add to cart" and "Buy now" buttons. A YouTube video player is partially visible below the browser window.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7
8   <style>
9     .cart{
10       background-color: yellow;
11     }
12     .buy {
13       background-color: orange;
14     }
15   </style>
16
17 </head>
18 <body>
19   <p>Adults Plain Cotton T-shirt</p>
20   <p></p>
21   <p>Price: $7.99</p>
22   <button class="cart" onclick="alert('Added');">Add to cart</button>
23   <button class="buy" onclick="alert('Loading...'); alert('Purchased');">Buy Now</button>
24 </body>
25 </html>

```

**May 18, 2024**

## HTML & CSS Basics (9AM - 10AM)

- Since I know very little about HTML and CSS, I decided to do the HTML and CSS course with the same tutor.
- The link to the video: [HTML and CSS \(Beginner to Pro\) Course](#) [6.5 hours].

### HTML Basics:

- Different tags in HTML- p, a, button tags

The screenshot shows a code editor on the left and a browser window on the right.

**Code Editor (website.html):**

```

1 <button>Hello</button>
2 <p>paragraph of text</p>
3 <a href="https://www.youtube.com/" target="_blank">Link to YouTube</a>
4 <p></p>
5 <a href="https://www.google.com/" target="_blank">Search with Google</a>
6 <p></p>
7 <a href="https://www.amazon.com/">Back to Amazon</a>
8 <p>Nike Black Running Shoes</p>
9 <p></p>
10 <p>$39 - in stock.</p>
11 <p></p>
12 <p></p>
13 <p>Free delivery tomorrow.</p>
14 <p></p>
15 <button>Add to Cart</button>
16 <button>Buy now</button>

```

**Browser:**

- Address bar: <https://www.y...>
- Content area:
  - Challenge Exercise: "Create a link to google.com (or a similar website for your country)"
  - Task: "If. Make the link from the previous exercise (1e.) open in a new tab"
  - Exercise details: "1g. Try to copy this design using HTML (the link goes to amazon.com)"
  - Links: "Link to YouTube", "Search with Google", "Back to Amazon", "Nike Black Running Shoes", "\$39 - in stock.", "Free delivery tomorrow.", "Add to Cart", "Buy now"
  - Solutions: "Solutions + more exercises in description"
  - Page title: "HTML & CSS Full Course - Beginner to Pro"
- Bottom status bar: "127.0.0.1:5501/website.html"

## CSS basics:

Styled three buttons using different styling properties.

The screenshot shows a code editor on the left and a browser window on the right.

**Code Editor (buttons.html):**

```

1 <style>
2   .sub-button {
3     background-color: #rgb(200, 0, 0);
4     color: #white;
5     border: none;
6     height: 36px;
7     width: 105px;
8     border-radius: 2px;
9     cursor: pointer;
10    margin-right: 8px;
11  }
12  .join-button {
13    background-color: #white;
14    border-color: #rgb(2, 130, 255);
15    border-style: solid;
16    border-width: 1px;
17    color: #rgb(2, 130, 255);
18    height: 36px;
19    width: 62px;
20    border-radius: 2px;
21    cursor: pointer;
22  }
23  .tweet-button {
24    background-color: #rgb(2, 158, 255);
25    color: #white;
26    border: none;
27    height: 36px;
28    width: 76px;
29    border-radius: 18px;
30    font-weight: bold;
31    font-size: 15px;
32    cursor: pointer;
33    margin-left: 8px;
34  }

```

**Browser:**

- Address bar: <https://127.0.0.1:5501/buttons.html>
- Content area: "SUBSCRIBE", "JOIN", "Tweet" buttons
- Bottom status bar: "127.0.0.1:5501/buttons.html"

Exercise: Styled as per the image shown in exercise.

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the following CSS:

```

1 <style>
2   a {
3     color: #rgb(0,113,133);
4   }
5   .heading {
6     font-size: 35px;
7     font-weight: 600;
8   }
9   .price {
10    color: #rgb(4, 143, 4);
11    font-weight: bold;
12  }
13  .delivery {
14    font-weight:600;
15  }
16  .cart {
17    background-color: #rgb(255, 216, 18);
18    border: none;
19    height: 30px;
20    width: 110px;
21    border-radius: 50px;
22    font-weight: bold;
23  }
24  .buy {
25    background-color: #rgb(255, 154, 39);
26    border: none;
27    height: 30px;
28    width: 110px;
29    border-radius: 50px;
30    font-weight: bold;
31  }
32 </style>
33

```

The browser window shows a challenge exercise from YouTube. It displays a product card for "Nike Black Running Shoes" with a price of \$39, stating "in stock" and "Free delivery tomorrow". There are "Add to Cart" and "Buy now" buttons at the bottom.

## Advanced CSS, Box Model & Text Styles (10 AM - 12 PM)

- Hover, transition and shadows.
- Dev Tools.
- CSS Box Model.

When hovered over the buttons, there is change in opacity, color and shadows.

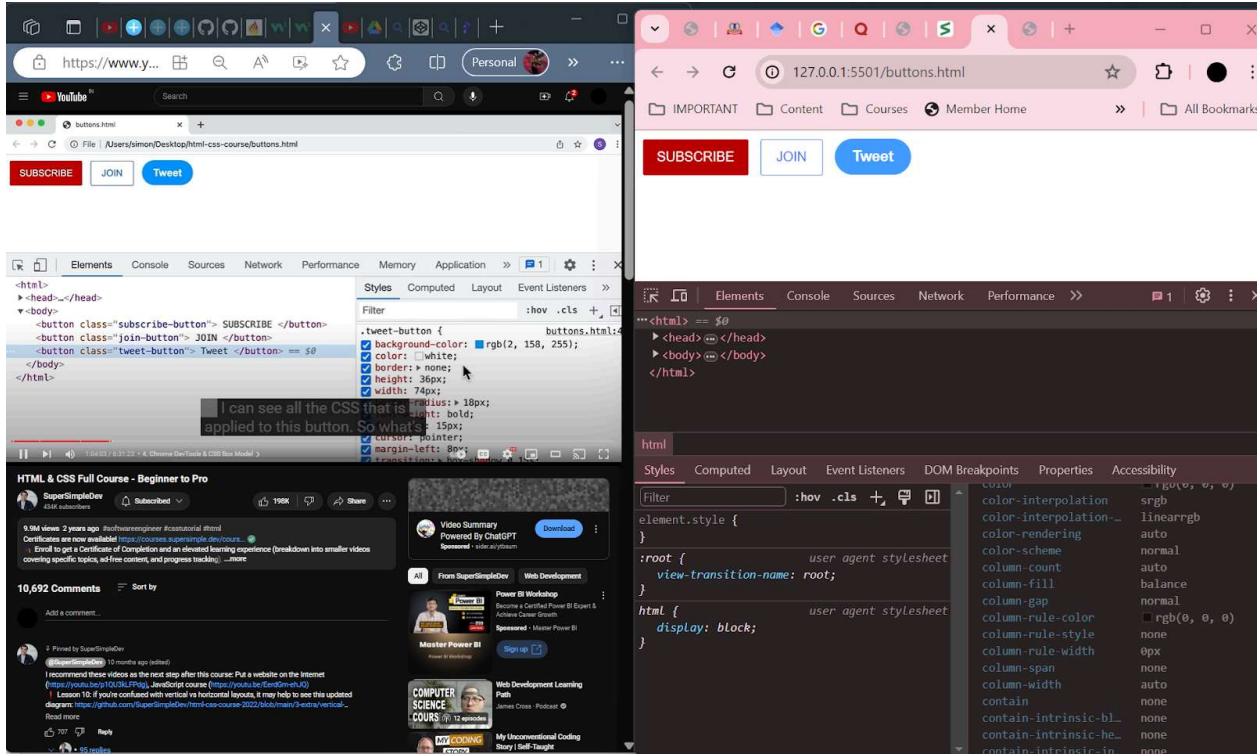
A screenshot of a Windows desktop environment. On the left is a code editor window titled "buttons.html" showing CSS and HTML code. On the right is a web browser window titled "127.0.0.1:5501/buttons.html" displaying three buttons: a red "SUBSCRIBE" button, a blue "JOIN" button, and a light blue "Tweet" button.

```
<style>
    .join-button {
        border-radius: 2px;
        cursor: pointer;
        transition: background-color 0.15s, color 0.15s;
    }
    .join-button:hover {
        background-color: #rgb(2, 130, 255);
        color: #white;
    }
    .join-button:active {
        opacity: 0.7;
    }
    .tweet-button {
        background-color: #rgb(2, 158, 255);
        color: #white;
        border: none;
        height: 36px;
        width: 76px;
        border-radius: 18px;
        font-weight: bold;
        font-size: 15px;
        cursor: pointer;
        margin-left: 8px;
        transition: box-shadow 0.15s;
    }
    .tweet-button:hover {
        box-shadow: 5px 5px 10px #rgba(0, 0, 0, 0.151);
    }
</style>
<button class="sub-button">SUBSCRIBE</button>
<button class="join-button">JOIN</button>
```

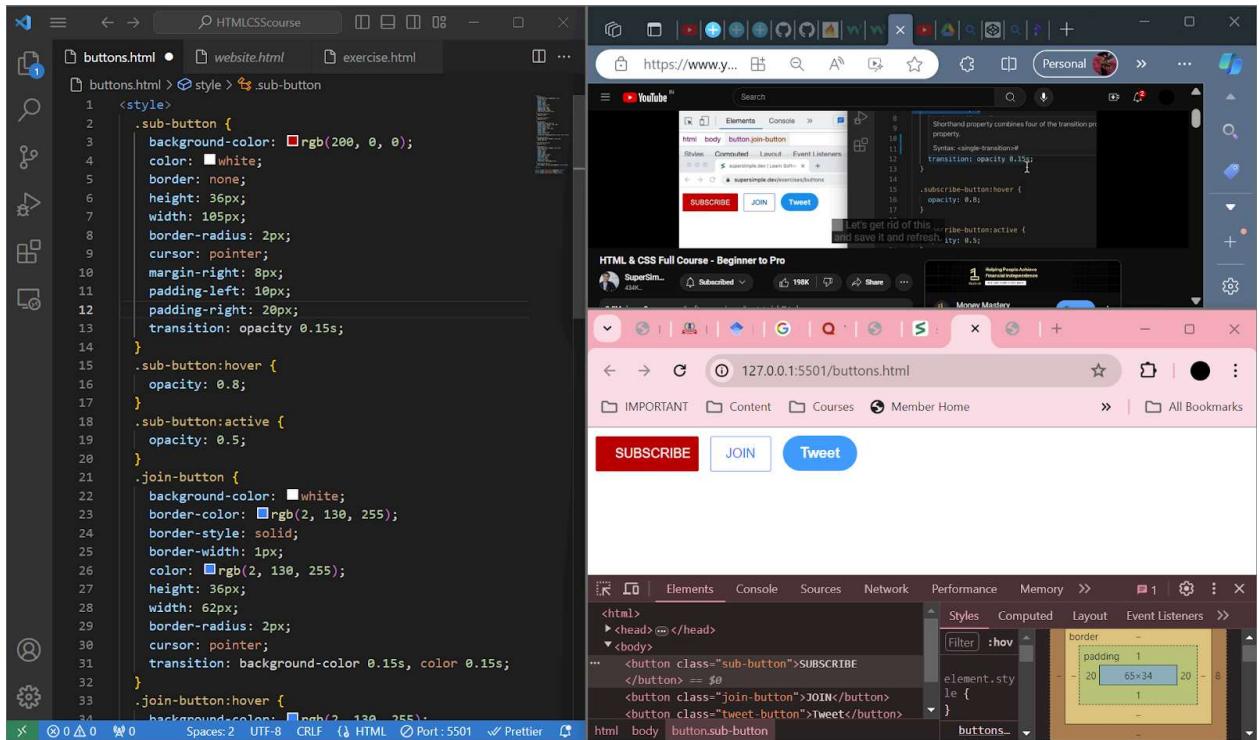
A second screenshot of a Windows desktop environment, identical to the first one. It shows the same code editor with the "buttons.html" file open and the same web browser displaying the three buttons.

```
<style>
    .join-button {
        border-radius: 2px;
        cursor: pointer;
        transition: background-color 0.15s, color 0.15s;
    }
    .join-button:hover {
        background-color: #rgb(2, 130, 255);
        color: #white;
    }
    .join-button:active {
        opacity: 0.7;
    }
    .tweet-button {
        background-color: #rgb(2, 158, 255);
        color: #white;
        border: none;
        height: 36px;
        width: 76px;
        border-radius: 18px;
        font-weight: bold;
        font-size: 15px;
        cursor: pointer;
        margin-left: 8px;
        transition: box-shadow 0.15s;
    }
    .tweet-button:hover {
        box-shadow: 5px 5px 10px #rgba(0, 0, 0, 0.151);
    }
</style>
<button class="sub-button">SUBSCRIBE</button>
<button class="join-button">JOIN</button>
```

How to use dev tools to see style elements. We can see the exact colors and styling used on the page. The HTML portion shows how the page is structured.



## Padding, border and Margins - Box model.



Added space between buttons using margins.

The screenshot shows a browser window with the URL <https://www.youtube.com/watch?v=...>. On the left, the DevTools sidebar shows the file structure: buttons.html > style > .join-button, .join-button:hover, .join-button:active, .tweet-button, .tweet-button:hover. The right pane shows the rendered HTML with three buttons: 'SUBSCRIBE' (red), 'JOIN' (blue), and 'Tweet' (blue). A tooltip over the 'JOIN' button displays its CSS properties, including background-color: #007bff, border-color: #007bff, border-style: solid, border-width: 1px, color: #fff, cursor: pointer, font-size: 15px, font-weight: bold, height: 36px, width: 76px, border-radius: 2px, and padding-left: 16px; padding-right: 16px;. The 'Tweet' button has a similar structure but with a different color scheme. The bottom of the browser window shows the YouTube channel 'SuperSimpleDev' with 198K subscribers.

```
<style>
.join-button {
    padding-bottom: 10px;
    padding-left: 16px;
    padding-right: 16px;
}
.join-button:hover {
    background-color: #007bff;
    color: #fff;
}
.join-button:active {
    opacity: 0.7;
}
.tweet-button {
    background-color: #007bff;
    color: #fff;
    border: none;
    height: 36px;
    width: 76px;
    border-radius: 18px;
    font-weight: bold;
    font-size: 15px;
    cursor: pointer;
    margin-left: 8px;
    transition: box-shadow 0.15s;
    vertical-align: top;
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 16px;
    padding-right: 16px;
}
.tweet-button:hover {
    box-shadow: 5px 5px 10px #007bff;
}

```

Did the basic exercise 4h.

The screenshot shows a browser window with the URL <https://www.youtube.com/watch?v=...>. On the left, the DevTools sidebar shows the file structure: exercise1.html > style > button. The right pane shows the rendered HTML with a navigation bar: 'Back' (disabled), '1', '2', '3', '4', '5', 'Next'. A tooltip over the 'Stretch' button displays its CSS properties, including background-color: #007bff, border-color: #007bff, border-style: solid, border-width: 1px, color: #fff, cursor: pointer, font-size: 15px, font-weight: bold, height: 36px, width: 76px, border-radius: 2px, and padding-left: 16px; padding-right: 16px;. Below the navigation bar, there are two more sections: '4j. 3D Click' (with a green 'Shadow' button) and '4k. Margin and padding together' (with three green buttons labeled 'One', 'Two', and 'Three'). The bottom of the browser window shows the YouTube channel 'SuperSimpleDev' with 198K subscribers.

```
<style>
button {
    padding: 5px;
    padding-left: 7px;
    padding-right: 7px;
    margin: 8px;
}
a {
    margin: 3px;
}

```

```
<button class="back-button">Back</button>
<a href="#">1</a>
<a href="#">2</a>
<a href="#">3</a>
<a href="#">4</a>
<a href="#">5</a>
<button class="next-button">Next</button>
```

- Text styles - font
- Experimenting with different fonts, sizes and weights.

```

1 <style>
2   .video-title {
3     font-family: Arial;
4     font-size: 18px;
5     font-weight: bold;
6     width: 300px;
7     line-height: 24px;
8     margin-bottom: 5px;
9   }
10
11 .vid-stats {
12   font-family: arial;
13   font-size: 14px;
14   color: #rgb(96,96,96);
15   margin-top: 0;
16 }
17
18 </style>
19
20
21 <p class="video-title">Talking
22 Tech and AI with Google CEO
23 Sundar Pichai!</p>
24
25 <p class="vid-stats">
26 | 3.4M views &#183; 6 months ago
27 </p>

```

## Styling Paragraphs - width, line height and font styles.

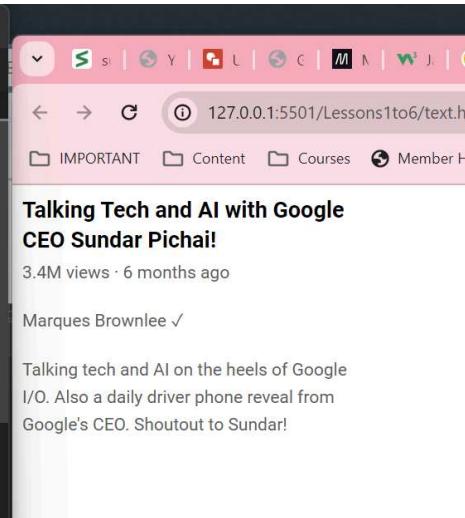
```

1 <style>
2   .vid-stats {
3     font-family: arial;
4     font-size: 14px;
5     color: #rgb(96,96,96);
6     margin-top: 0;
7     margin-bottom: 20px;
8   }
9
10 .vid-author {
11   font-family: arial;
12   font-size: 14px;
13   color: #rgb(96,96,96);
14   margin-top: 0;
15   margin-bottom: 20px;
16 }
17
18 .desc {
19   font-family: arial;
20   font-size: 14px;
21   color: #rgb(96,96,96);
22   width: 280px;
23   line-height: 22px;
24   margin-top: 0;
25 }
26
27 </style>
28
29
30 <p class="video-title">Talking
31 Tech and AI with Google CEO
32 Sundar Pichai!</p>
33
34 <p class="desc">
35 Talking tech and AI on the heels of Google
36 I/O. Also a daily driver phone reveal from
37 Google's CEO. Shoutout to Sundar!
38
39
40 <p class="vid-stats">
41 | 3.4M views &#183; 6 months ago
42 </p>

```

CSS specificity - how the more specific styles are applied and other less specific code is overridden.

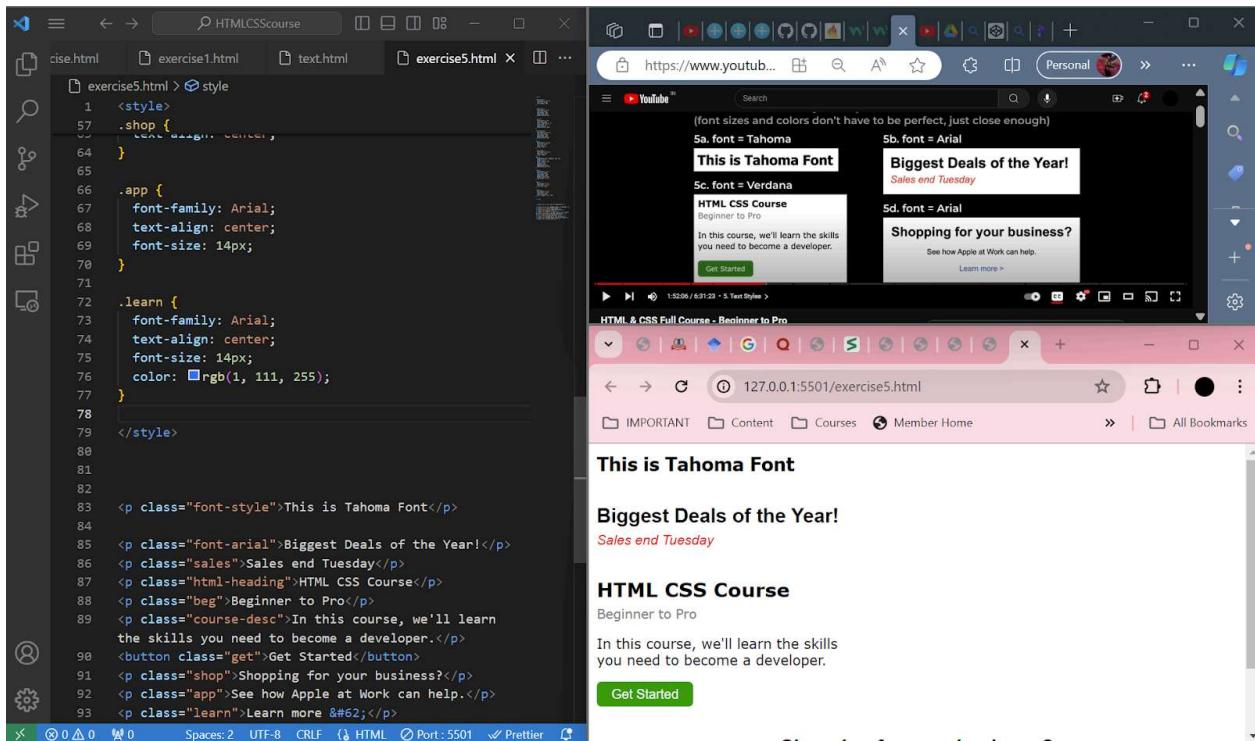
```
1 p {
2   font-family: Roboto;
3   margin-top: 0;
4   margin-bottom: 0;
5 }
6
7 .video-title {
8   font-size: 18px;
9   font-weight: bold;
10  width: 280px;
11  line-height: 24px;
12  margin-bottom: 5px;
13 }
14
15 .vid-stats {
16  font-size: 14px;
17  color: □rgb(96,96,96);
18  margin-bottom: 20px;
19 }
20
21
22 .vid-author {
23  font-size: 14px;
24  color: □rgb(96,96,96);
25  margin-bottom: 20px;
26 }
27
28 .desc {
29  font-size: 14px;
30  color: □rgb(96,96,96);
31  width: 280px;
32  line-height: 22px;
33  margin-bottom: 100px;
34 }
```



The screenshot shows a browser window displaying a video player interface. The video title is "Talking Tech and AI with Google CEO Sundar Pichai!". Below the title, it says "3.4M views · 6 months ago". The video description reads: "Talking tech and AI on the heels of Google I/O. Also a daily driver phone reveal from Google's CEO. Shoutout to Sundar!"

On the left side of the image, there is a file explorer window from VS Code. It shows a project structure with folders like "Channel-pics", "Lessons1to6", and "Exercises". Inside "Lessons1to6/Exercises", there are files named "exer9.html", "exercise.html", "exercise1.html", "exercise5.html", "exercise7.html", and "exercise8.html". The "text.css" file is open in the editor, showing the CSS code provided at the top of the page.

Exercises - was able to do styling by looking at sample images.



```
<style>
  .shop {
    text-align: center;
  }
  .app {
    font-family: Arial;
    text-align: center;
    font-size: 14px;
  }
  .learn {
    font-family: Arial;
    text-align: center;
    font-size: 14px;
    color: #rgb(1, 111, 255);
  }
</style>



This is Tahoma Font



Biggest Deals of the Year!



Sales end Tuesday



HTML CSS Course



Beginner to Pro



In this course, we'll learn the skills you need to become a developer.



Get Started



Shopping for your business?



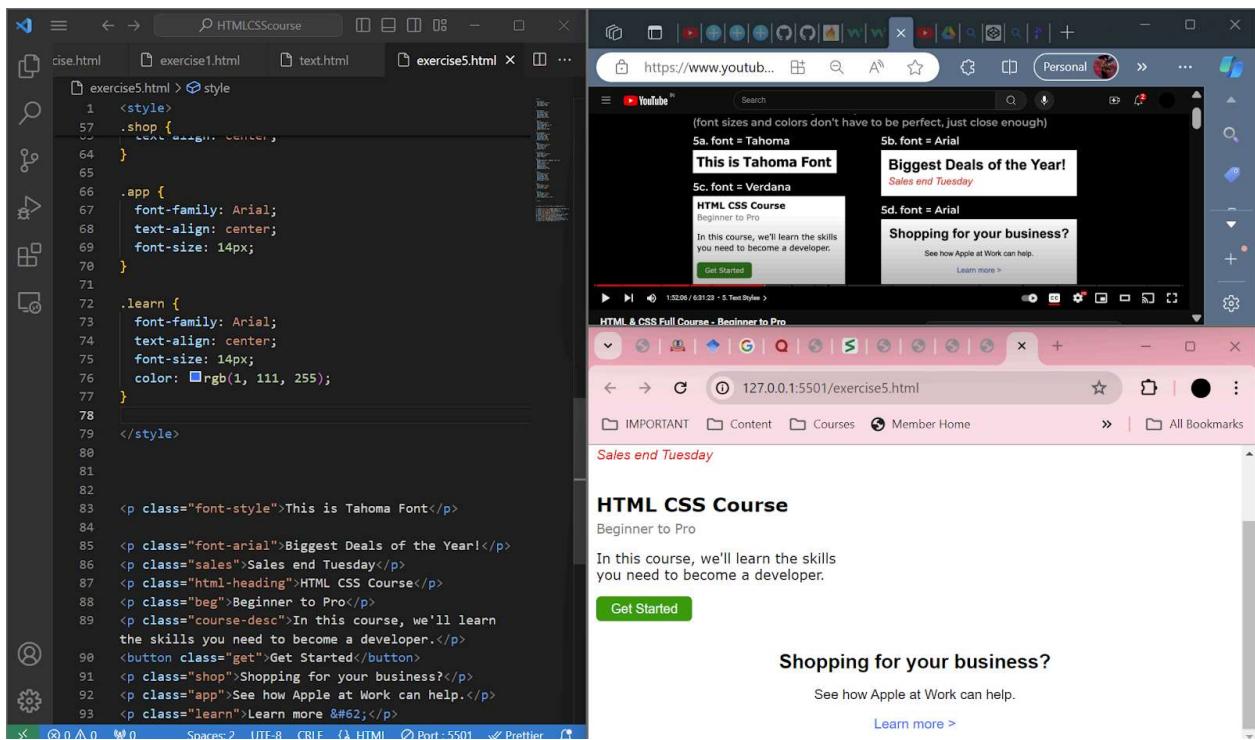
See how Apple at Work can help.



Learn more &#62;


```

Was able to style with some ease.



```
<style>
  .shop {
    text-align: center;
  }
  .app {
    font-family: Arial;
    text-align: center;
    font-size: 14px;
  }
  .learn {
    font-family: Arial;
    text-align: center;
    font-size: 14px;
    color: #rgb(1, 111, 255);
  }
</style>



This is Tahoma Font



Biggest Deals of the Year!



Sales end Tuesday



HTML CSS Course



Beginner to Pro



In this course, we'll learn the skills you need to become a developer.



Get Started



Shopping for your business?



See how Apple at Work can help.



Learn more &#62;


```

## HTML structure, Images and Text Boxes ( 12PM - 2 PM)

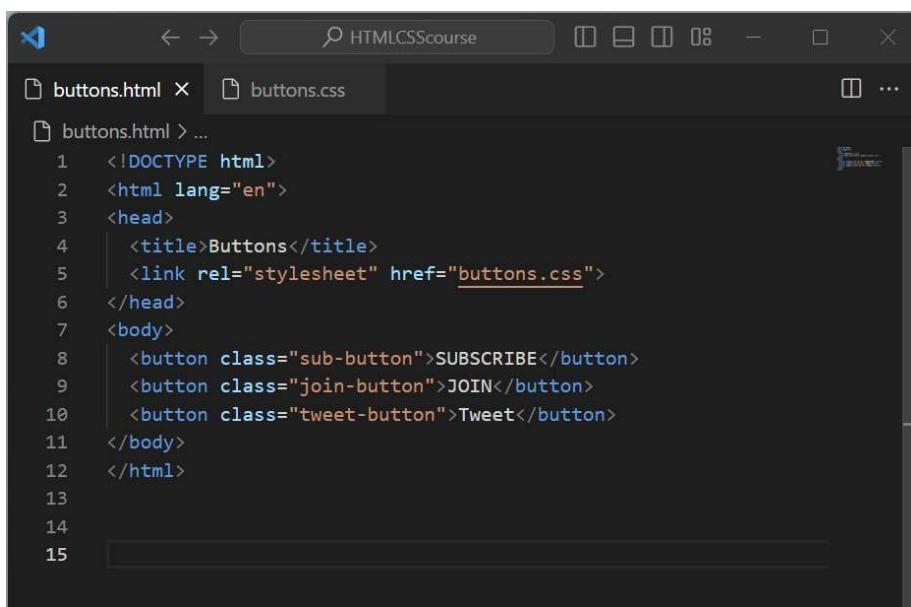
Html, head and body tags. ‘a’ tag and adding ‘target’ attribute.

```
website.html > html > body
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  | <title>First Website</title>
5  </head>
6  <body>
7  | <button>Hello</button>
8  | <p>paragraph of text</p>
9  | <a href="https://www.youtube.com/" target="_blank">Link
10 | to YouTube</a>
11 | <p></p>
12 | <a href="https://www.google.com/" target="_blank">Search
13 | with Google</a>
14 |
15 | <p></p>
16 | <a href="https://www.amazon.com/">Back to Amazon</a>
17 | <p>Nike Black Running Shoes</p>
18 | <p></p>
19 | <p>$39 - in stock.</p>
20 | <p></p>
21 | <p>Free delivery tomorrow.</p>
22 | <p></p>
23 | <button>Add to Cart</button>
24 | <button>Buy now</button>
25 | </body>
26 | </html>
```

Moving CSS code to a separate file to make html code look clean and make it easier to read.

Link tag.

Void elements - link tag, img tag, etc.



The screenshot shows a code editor window with two tabs: 'buttons.html' and 'buttons.css'. The 'buttons.html' tab is active, displaying the following HTML code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Buttons</title>
<link rel="stylesheet" href="buttons.css">
</head>
<body>
<button class="sub-button">SUBSCRIBE</button>
<button class="join-button">JOIN</button>
<button class="tweet-button">Tweet</button>
</body>
</html>
```

The 'buttons.css' tab is visible in the background. The code editor has a dark theme and includes standard window controls (minimize, maximize, close) at the top.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Text Styling</title>
    <link rel="stylesheet" href="./styles/text.css" />
  </head>
  <body>
    <p class="video-title">Talking Tech and AI with Google CEO Sundar Pichai!</p>
    <p class="vid-stats">3.4M views &#183; 6 months ago</p>
    <p class="vid-author">Marques Brownlee &#10003;</p>
    <p class="desc">Talking tech and AI on the heels of Google I/O. Also a daily driver phone reveal from Google's CEO. Shoutout to Sundar!</p>
    <p class="apple-text">Shop early for the best selection of holiday favourites. <span class="shop">Shop now &#62;</span></p>
  </body>
</html>

```

Downloading fonts from google fonts and Adding them to our code. “Roboto” font family applied successfully.

Talking Tech and AI with Google  
CEO Sundar Pichai!

3.4M views · 6 months ago

Marques Brownlee ✓

Talking tech and AI on the heels of Google I/O. Also a daily driver phone reveal from Google's CEO. Shoutout to Sundar!

<script></script>

Style	Computed
font-family	Roboto
font-feature-settings	normal
font-kerning	auto
font-optical-sizing	auto
font-palette	normal
font-size	18px

The tutorial is structured in a way where we get familiar with basics and build up to advance concepts by building a clone of youtube.com

Images - img tag and its properties. They maintain their original size and hence, width is used to contain the image.

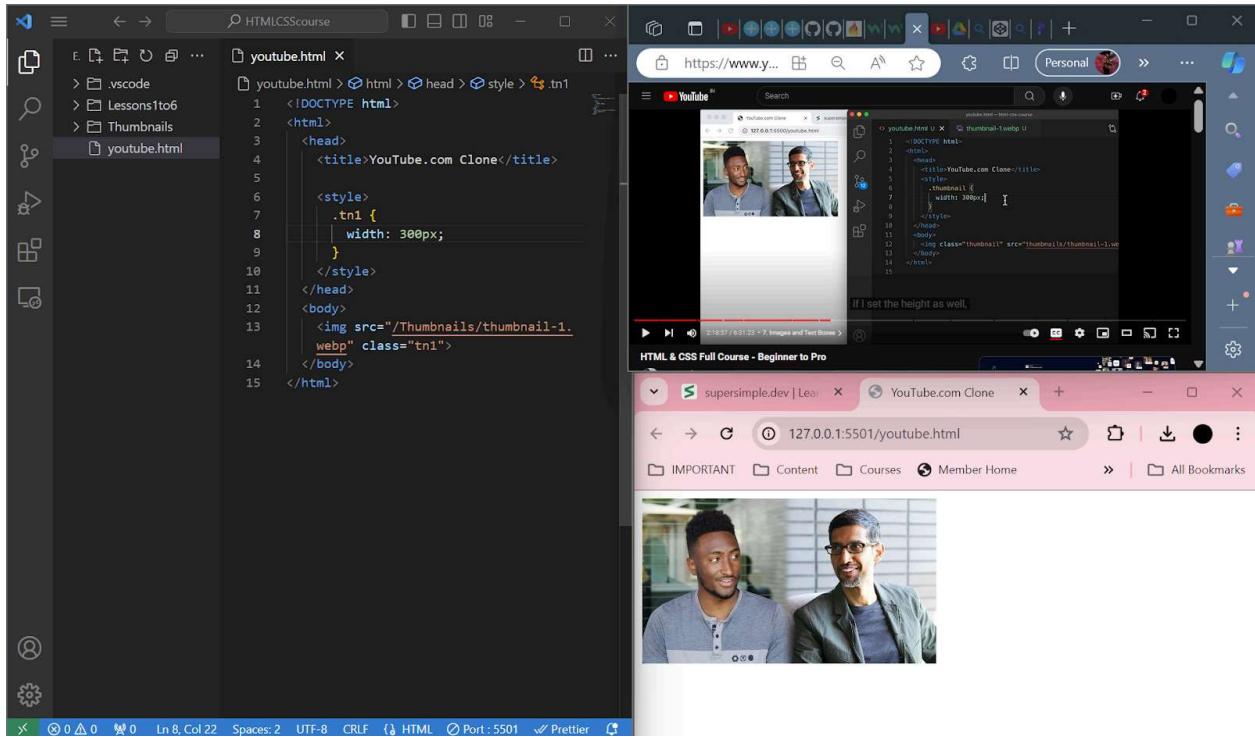


Image attributes - width, height, object-fit, object-position, and border

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the file `youtube.html` with the following content:

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>YouTube.com Clone</title>
5
6          <style>
7              .tn1 {
8                  width: 300px;
9                  height: 300px;
10                 object-fit: contain;
11                 object-position: top;
12                 border-color: red;
13                 border-width: 2px;
14                 border-style: solid;
15             }
16         </style>
17     </head>
18     <body>
19         
20     </body>
21 </html>

```

The browser window shows the rendered output of the code, displaying a video thumbnail of two men in a frame with a red border.

## Added a search bar - Text Box (Input tag)

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the file `youtube.html` with the following content:

```

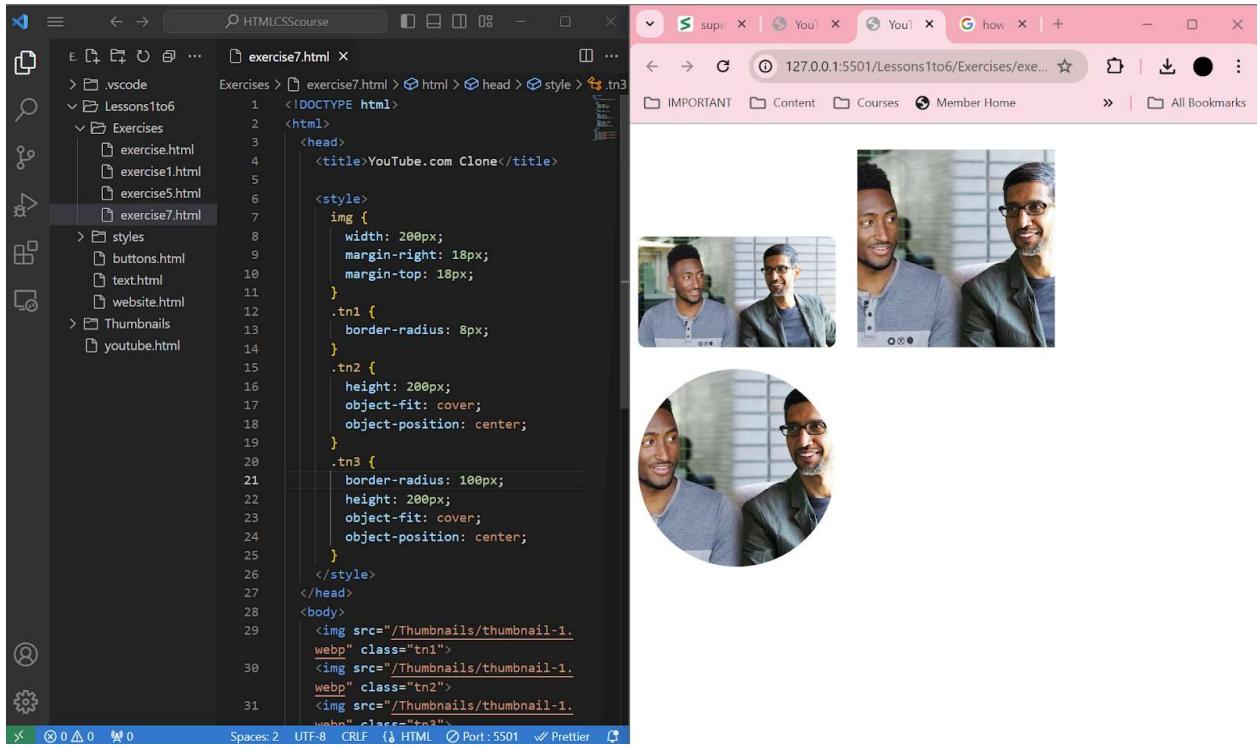
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>YouTube.com Clone</title>
5
6          <style>
7              .tn1 {
8                  width: 300px;
9              }
10             .search-box {
11                 font-size: 20px;
12                 margin-left: 18px;
13             }
14         </style>
15     </head>
16     <body>
17         
18         <input class="search-box" type="text" placeholder="Search">
19     </body>
20 </html>

```

The browser window shows the rendered output of the code, displaying a video thumbnail of two men with a search bar at the bottom labeled "Search". A tooltip in the browser indicates that the search bar is a text input element.

## Exercises

Different shaped images using object fit, width and border radius.



The screenshot shows a VS Code interface with an open file named 'exercise7.html'. The code defines three CSS classes for images:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>YouTube.com Clone</title>
5   </head>
6   <body>
7     
8     
9     
10
11   </body>
12 </html>
```

The browser preview on the right displays three images corresponding to these classes. The first image is a square thumbnail. The second image is a rounded rectangular thumbnail. The third image is a circular thumbnail.

Styling the text box - search bar to look like google search bar

```
29   .search-box {
30     font-size: 16px;
31     margin-left: 100px;
32     width: 450px;
33     padding: 10px;
34     padding-left: 15px;
35     box-shadow: 1px 1px 5px grey;
36     border-radius: 20px;
37     border: none;
38   }
39   </style>
40 </head>
41 <body>
42   
43   
44   
45   <input class="search-box" type="text"
46     placeholder="Search Google or type a URL">
47 </html>
```

Results:

---



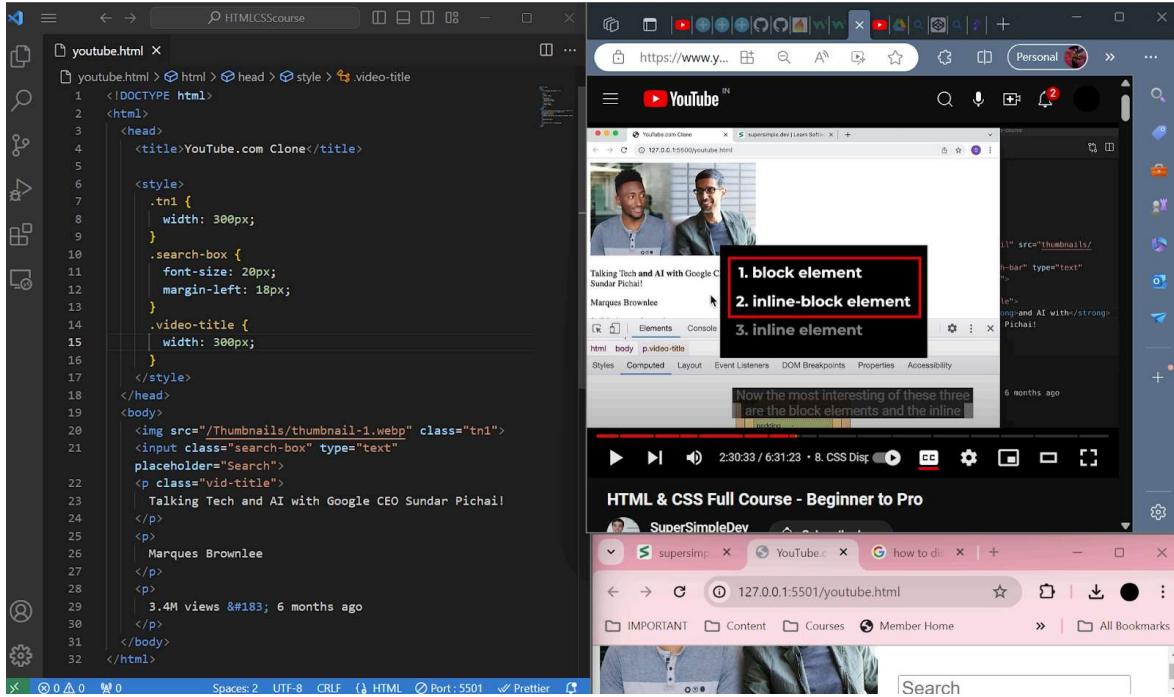
Search Google or type a URL

Lunch (2PM - 3PM)

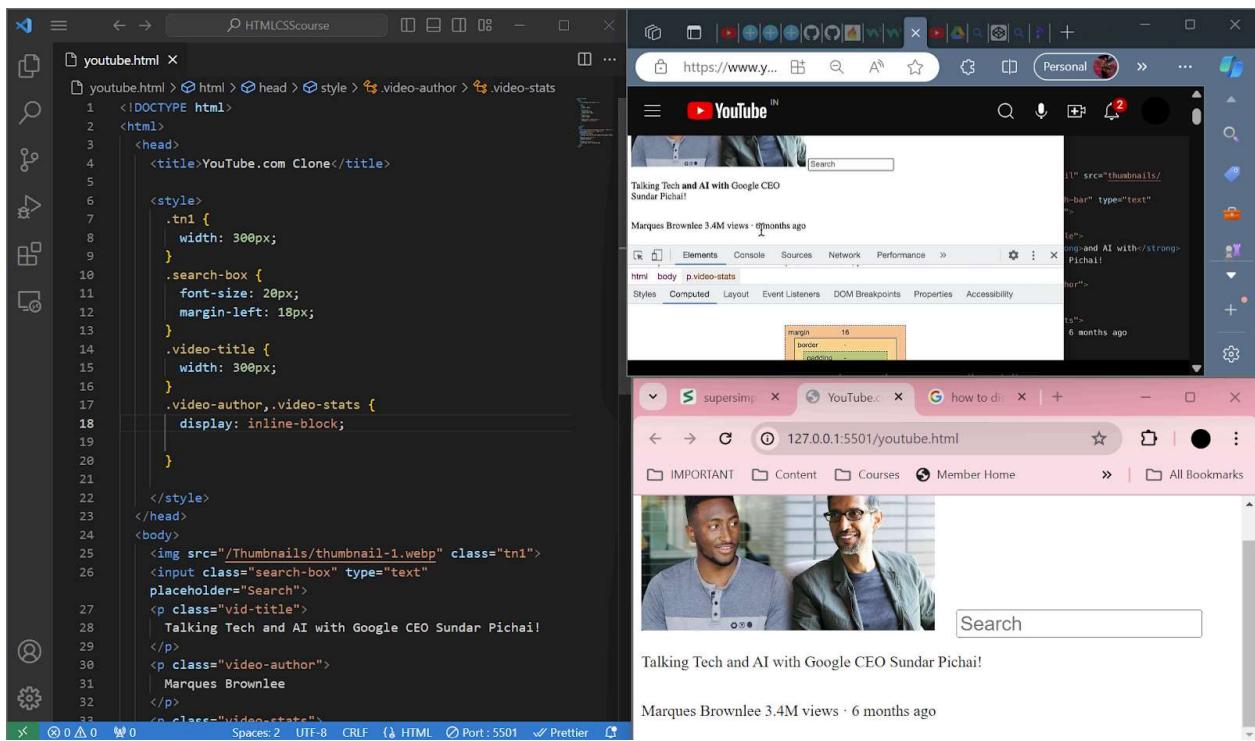
**Advanced HTML & CSS (3PM - 7PM)**

CSS display property

1. Block elements
2. Inline elements
3. Inline block elements

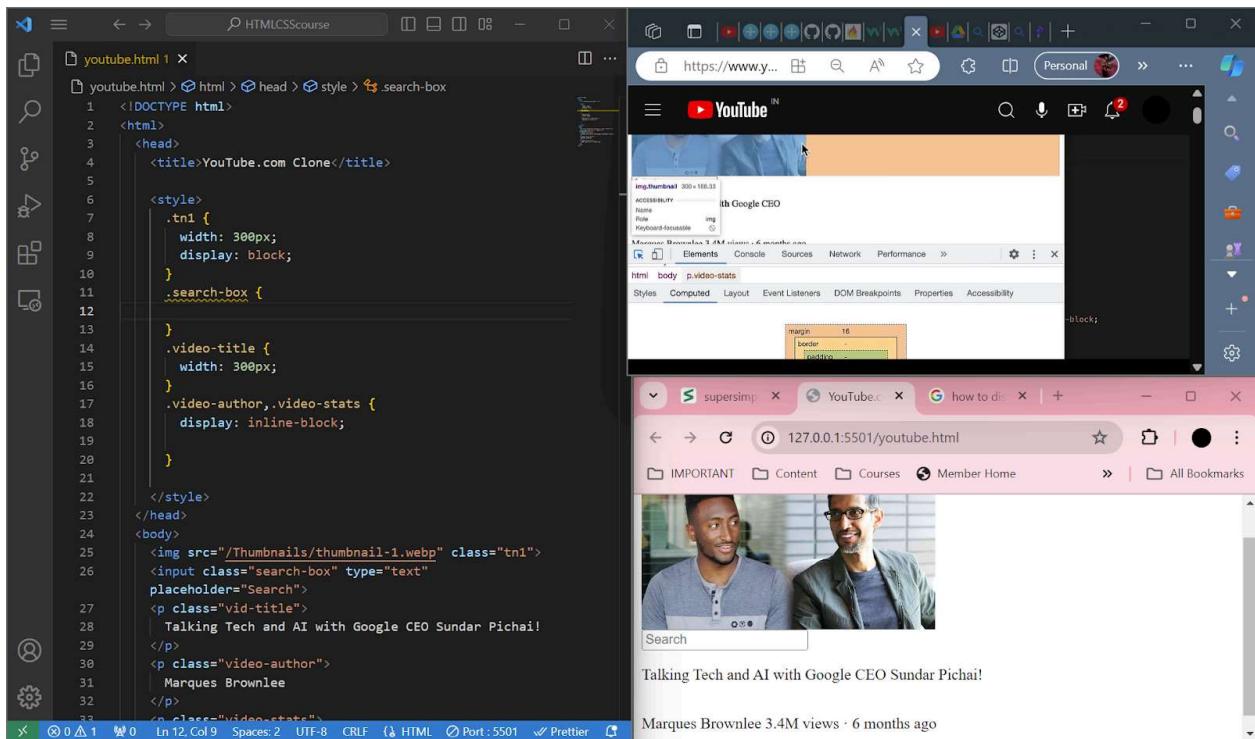


Inline element styling. Inline display property allows multiple elements in the same line. Image and search bar in the same line.



Block elements - take up the whole block or line.  
(display: block;) image and search bar on different lines.

Inline block elements - behave as block elements and take up as much width as available and elements are on the same line (no line break) - video author and video stats in the below image are inline elements.



## Exercises

Got good practice by doing the exercises.

Styled google logo and search bar.

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the file `exercise8.html` with the following content:

```

<html lang="en">
  <head>
    <style>
      .sum-text {
        vertical-align: middle;
        margin-bottom: 50px;
      }
      img {
        display: block;
        width: 200px;
        margin-left: 200px;
        margin-bottom: 20px;
      }
      .gg {
        border: none;
        margin-left: 50px;
        padding-top: 12px;
        padding-bottom: 12px;
        padding-left: 16px;
        padding-right: 16px;
        font-size: 16px;
        width: 50px;
        border-radius: 30px;
        box-shadow: 0 1px 6px rgba(0, 0, 0, 0.2);
      }
    </style>
  </head>
  <body>
    <input type="text" class="name" placeholder="Name">
  </body>

```

The browser window shows a survey form with fields for Name and Email. Below the form, a message says "Thanks for chatting with our customer support. Would you like to take our quick survey? Yes No". The Google logo is at the bottom.

## DIV element

Tutorial showed how div elements can be used to create blocks and can be used to style each div differently.

The screenshot shows a code editor on the left and a browser window on the right. The code editor displays the file `youtube.html` with the following content:

```

<html>
  <body>
    <div>
      <style>
        .video-title {
          width: 300px;
        }
      </style>
    </div>
    <input class="search-box" type="text" placeholder="Search">
    
    <p class="vid-title">
      Talking Tech and AI with Google CEO Sundar Pichai!
    </p>
    <p class="video-author">
      Marques Brownlee
    </p>
    <p class="video-stats">
      3.4M views 6 months ago
    </p>
    <div>
    </div>
  </body>
</html>

```

The browser window shows a YouTube video player for a video titled "Talking Tech and AI with Google CEO Sundar Pichai!". The video has 3.4M views and was posted 6 months ago. A tooltip on the right side of the video player area says "`<div>` = division = just a box".

Created a div at the bottom of the screen.

The screenshot shows a development environment with two main windows. On the left is a code editor in VS Code displaying the file `youtube.html`. The code includes a new `<div>` element at the bottom of the page. On the right is a browser window showing the rendered page, which displays the text "this is a div" at the bottom. Below the browser is a developer tools panel showing the element tree, highlighting the newly created `<div>` node.

```
<html>
  <head>
    <style>
      .video-title {
        width: 300px;
      }
    </style>
  </head>
  <body>
    <input class="search-box" type="text" placeholder="Search">
    

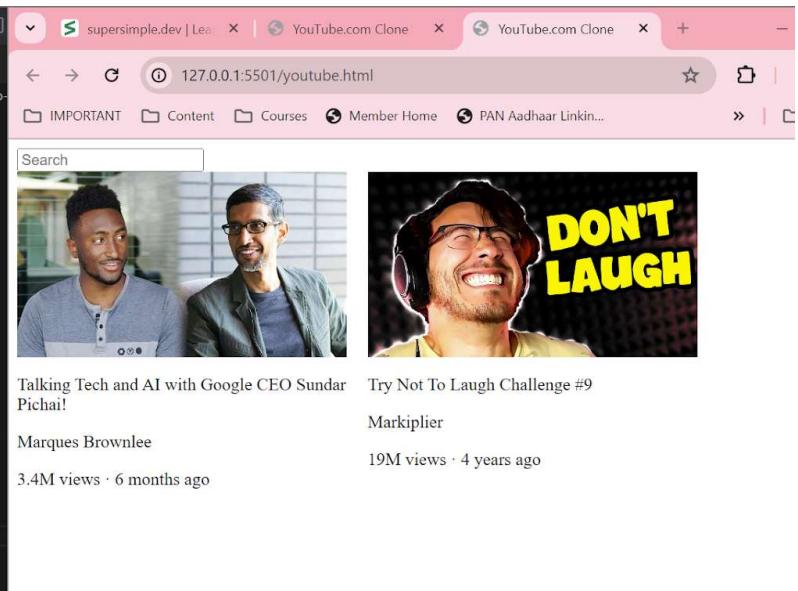
    <p class="vid-title">
      Talking Tech and AI with Google CEO Sundar Pichai!
    </p>
    <p class="video-author">
      Marques Brownlee
    </p>
    <p class="video-stats">
      3.4M views · 6 months ago
    </p>
    <div>
      this is a div
    </div>
  </body>
</html>
```

Duplicated the div to create two video previews (build a youtube home screen)

The screenshot shows a development environment with two main windows. On the left is a code editor in VS Code displaying the file `youtube.html`. The code now contains two `<div>` elements under the `<body>` tag, each containing a video thumbnail and associated text. On the right is a browser window showing the rendered page, which displays two video preview cards side-by-side. The developer tools panel below the browser shows the element tree with two `<div>` nodes highlighted.

```
<html>
  <head>
    <style>
      .video-preview {
        
      }
    </style>
  </head>
  <body>
    <div class="video-preview">
      
      <p class="vid-title">
        Talking Tech and AI with Google CEO Sundar Pichai!
      </p>
      <p class="video-author">
        Marques Brownlee
      </p>
      <p class="video-stats">
        3.4M views · 6 months ago
      </p>
    </div>
    <div class="video-preview">
      
      <p class="vid-title">
        Talking Tech and AI with Google CEO Sundar Pichai!
      </p>
      <p class="video-author">
        Marques Brownlee
      </p>
      <p class="video-stats">
        3.4M views · 6 months ago
      </p>
    </div>
  </body>
</html>
```

Added the second preview details.



youtube.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>YouTube.com Clone</title>
  </head>
  <style>
    .tn1 {
      width: 300px;
      display: block;
    }

    .video-title {
      width: 300px;
    }

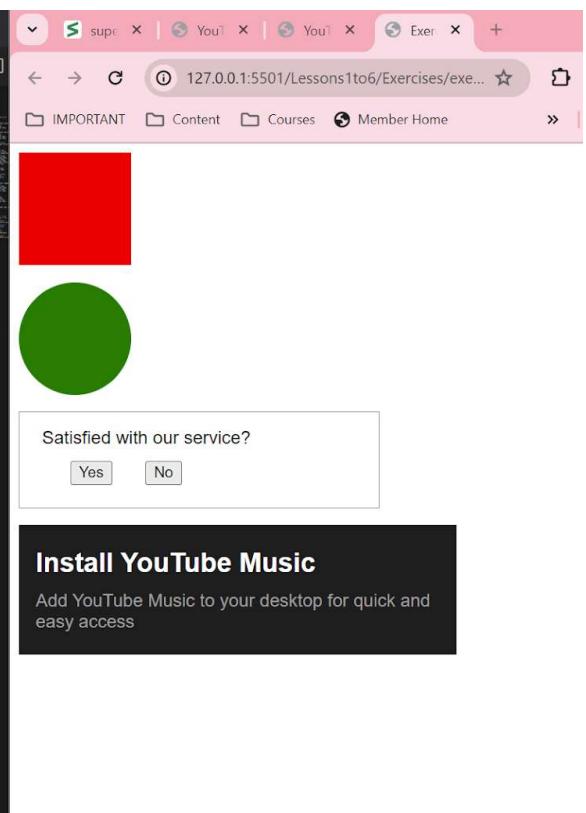
    .search-box {
      display: block;
    }

    .video-preview {
      width: 300px;
      display: inline-block;
      vertical-align: top;
      margin-right: 15px;
    }
  </style>

```

Exercises:

Executed the exercises.



exer9.html

```
<html lang="en">
<head>
  <style>
    .green {
      height: 100px;
      border-radius: 50px;
      margin-bottom: 15px;
    }

    .question {
      font-family: Arial;
      vertical-align: middle;
    }

    .sur {
      border-width: 1px;
      border-style: solid;
      border-color: #rgb(150, 150, 150);
      width: 300px;
      padding-left: 20px;
      padding-bottom: 20px;
      padding-top: 20px;
      margin-bottom: 15px;
      vertical-align: middle;
      line-height: 5px;
    }

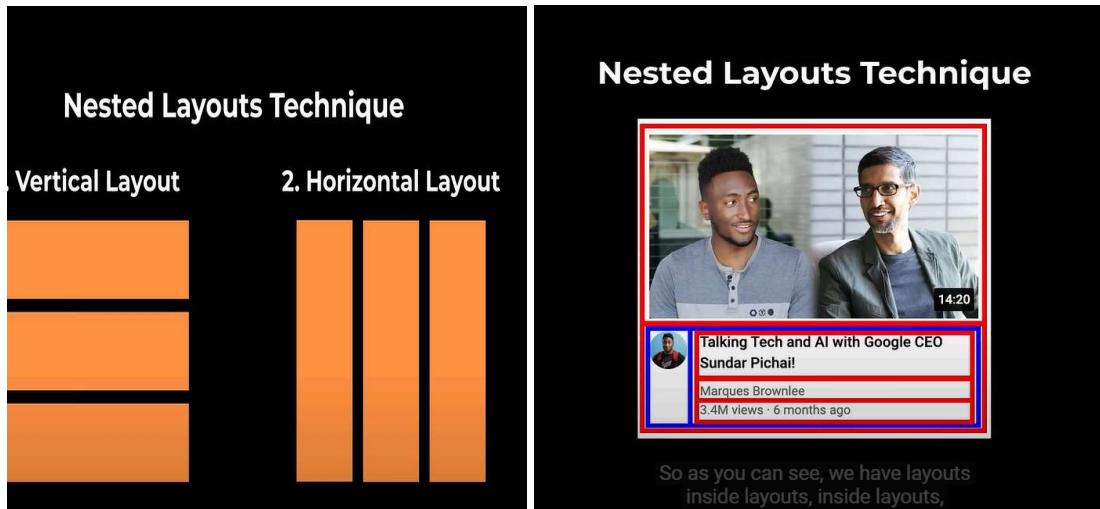
    button {
      margin-top: 18px;
      margin-left: 25px;
    }

    p {
      font-family: Arial;
      margin-top: 0;
      margin-bottom: 0;
    }
  </style>

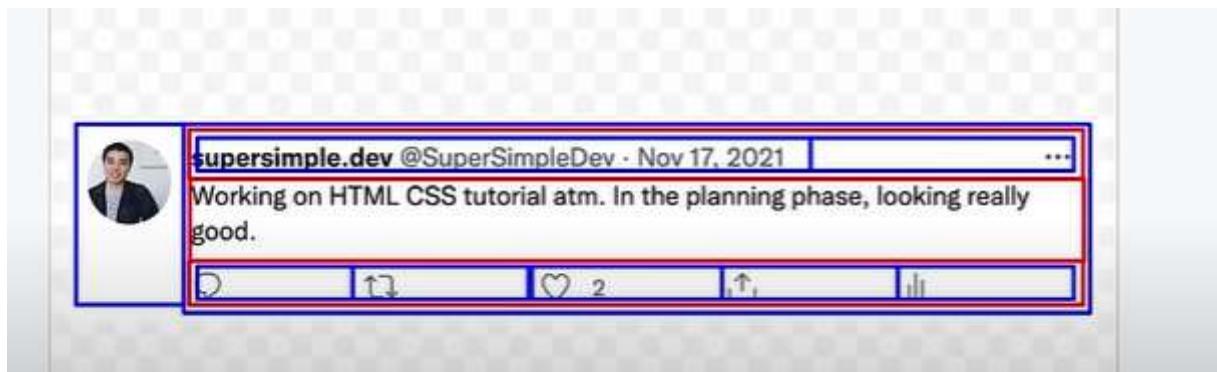
```

## Nested Layouts Technique

Nested layouts have vertical layouts and horizontal layouts that can be used to design and style each element within a block.



Marked the nested layouts technique using google drawings.



Adding the channel profile pic. Also added a few div elements to divide the first video preview into nested layouts.

The screenshot shows a browser window with two tabs: 'supersimple.dev | Learn' and 'YouTube.com Clone'. The main content area displays a video thumbnail for 'Try Not To Laugh' by 'Markiplier', featuring Sundar Pichai and Marques Brownlee. Below the thumbnail, the video title is 'Talking Tech and AI with Google CEO Sundar Pichai!', the views are 19M, and the upload date is 4 years ago. The bottom section shows a comment from 'Marques Brownlee' with 3.4M views and a timestamp of 6 months ago. To the right, the browser's developer tools are open, showing the HTML code for the 'youtube.html' file. The code includes the structure of the video preview, the search box, and the video statistics.

```
<html>
  <head>
    <style>
      .search-box { ... }
      .video-preview { ... }
      .tn1 { ... }
      .vid-title { ... }
      .video-author { ... }
      .video-stats { ... }
    </style>
  </head>
  <body>
    <input class="search-box" type="text" placeholder="Search">

    <div class="video-preview">
      <div>
        
      </div>
      <div>
        
        <p class="vid-title">Talking Tech and AI with Google CEO Sundar Pichai!</p>
        <p class="video-author">Markiplier</p>
        <p>19M views · 4 years ago</p>
        <p class="video-stats">3.4M views 6 months ago</p>
      </div>
    </div>
  </body>
</html>
```

Styling the profile picture.

The screenshot shows a browser window with two tabs: "supersimple.dev | Learn" and "YouTube.com Clone". The main content area displays a video thumbnail for "Talking Tech and AI with Google CEO Sundar Pichai" by Marques Brownlee, with 19M views and posted 6 months ago. To the right is a code editor showing the HTML and CSS for the "youtube.html" file. The code includes styles for channel pictures, video titles, search boxes, and video previews.

```
<!DOCTYPE html>
<html> == $0
  > <head> == </head>
  > <body> == </body>
</html>
```

```
youtube.html x text.html
youtube.html > html > head > link
rel="stylesheet"

<style>
  p {
    font-family: Roboto, Arial;
  }
  .tn1 {
    width: 300px;
    display: block;
  }

  .vid-title {
    margin-top: 0;
    font-size: 14px;
    font-weight: 500;
  }

  .search-box {
    display: block;
  }

  .video-preview {
    width: 300px;
    display: inline-block;
    vertical-align: top;
    margin-right: 15px;
  }

  .channel-pic {
    display: inline-block;
    width: 50px;
    vertical-align: top;
  }
```

Refined the styling of the first video preview by changing the text font, size and color using nested layouts.

The screenshot shows a browser window with two video preview cards. The first card features two men and is titled "Talking Tech and AI with Google CEO Sundar Pichai!". The second card features a man laughing and is titled "Try Not To Laugh Challenge #9". On the right, the browser's developer tools are open, displaying the CSS code for the "video-preview" class. The code includes styles for the channel picture, title, author, and stats.

```
video-preview {
    position: relative;
    width: 100px;
    height: 100px;
}

video-preview::before {
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    width: 100px;
    height: 100px;
    background-color: black;
    opacity: 0.5;
}

video-preview img {
    width: 100px;
    height: 100px;
    object-fit: cover;
}

video-preview .channel-pic {
    width: 100px;
    height: 100px;
    border-radius: 50%;
    overflow: hidden;
}

video-preview .channel-pic img {
    width: 100px;
    height: 100px;
    object-fit: cover;
}

video-preview .vid-info {
    margin-top: 5px;
    font-size: 0.8em;
    color: #ccc;
}

video-preview .vid-title {
    margin-top: 5px;
    font-size: 1em;
    font-weight: bold;
    color: #000;
}

video-preview .vid-author {
    margin-top: 5px;
    font-size: 0.8em;
    color: #ccc;
}

video-preview .vid-stats {
    margin-top: 5px;
    font-size: 0.8em;
    color: #ccc;
}
```

Duplicated all the elements and styling for the 2nd video preview.

The screenshot shows a browser window with two video preview cards. The first card features two men and is titled "Talking Tech and AI with Google CEO Sundar Pichai!". The second card features a man laughing and is titled "Try Not To Laugh Challenge #9". On the right, the browser's developer tools are open, displaying the CSS code for the "video-preview" class. A new "margin" style has been added to the code.

```
video-preview {
    position: relative;
    width: 100px;
    height: 100px;
}

video-preview::before {
    content: '';
    position: absolute;
    top: 0;
    left: 0;
    width: 100px;
    height: 100px;
    background-color: black;
    opacity: 0.5;
}

video-preview img {
    width: 100px;
    height: 100px;
    object-fit: cover;
}

video-preview .channel-pic {
    width: 100px;
    height: 100px;
    border-radius: 50%;
    overflow: hidden;
}

video-preview .channel-pic img {
    width: 100px;
    height: 100px;
    object-fit: cover;
}

video-preview .vid-info {
    margin-top: 5px;
    font-size: 0.8em;
    color: #ccc;
}

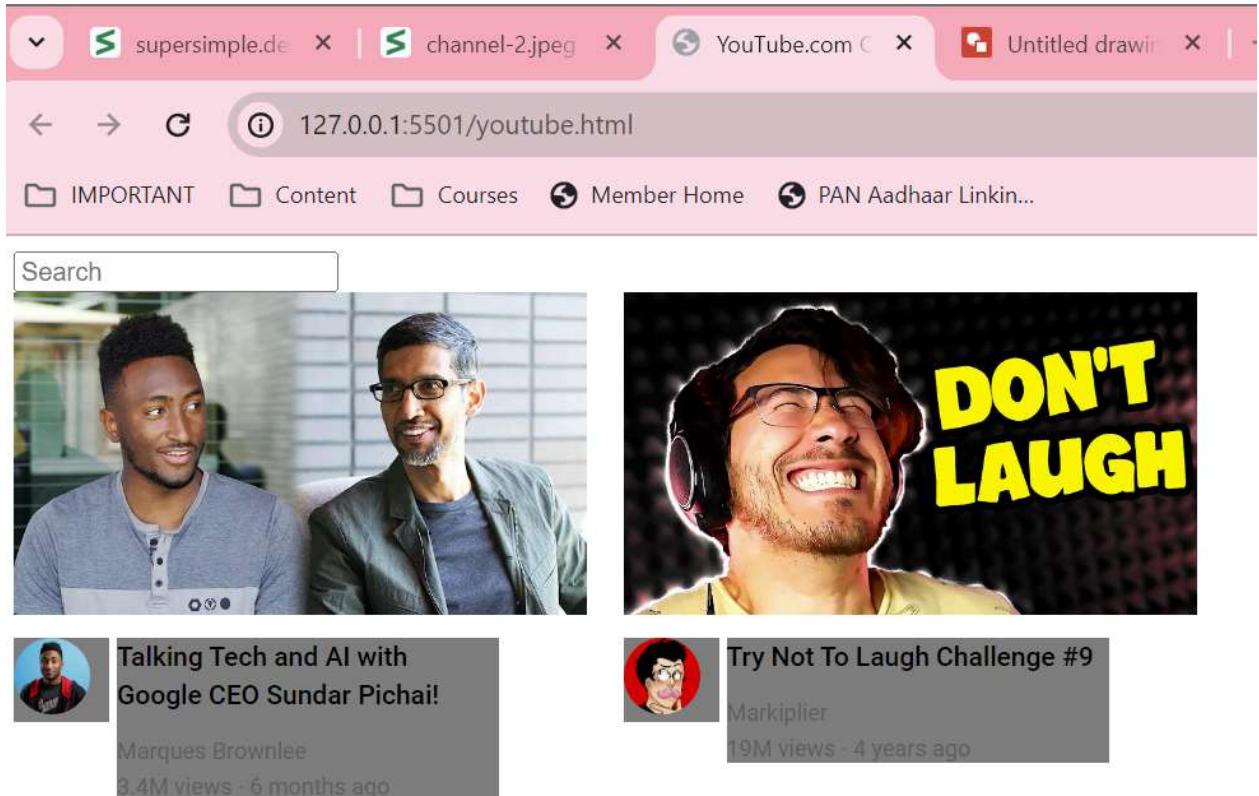
video-preview .vid-title {
    margin-top: 5px;
    font-size: 1em;
    font-weight: bold;
    color: #000;
}

video-preview .vid-author {
    margin-top: 5px;
    font-size: 0.8em;
    color: #ccc;
}

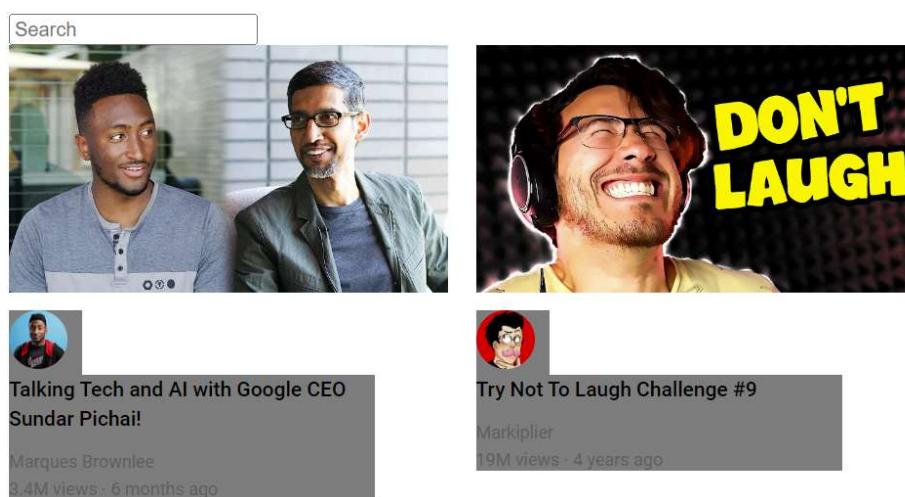
video-preview .vid-stats {
    margin-top: 5px;
    font-size: 0.8em;
    color: #ccc;
}
```

## CSS Grid

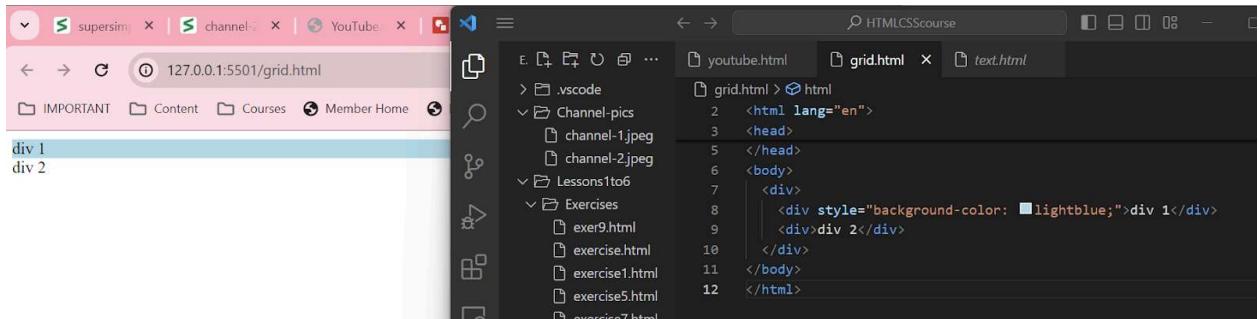
Inline-block elements cause some alignment issues. Instead we can use the CSS grid. Added background to visualize.



Alignment issues caused by increasing width.



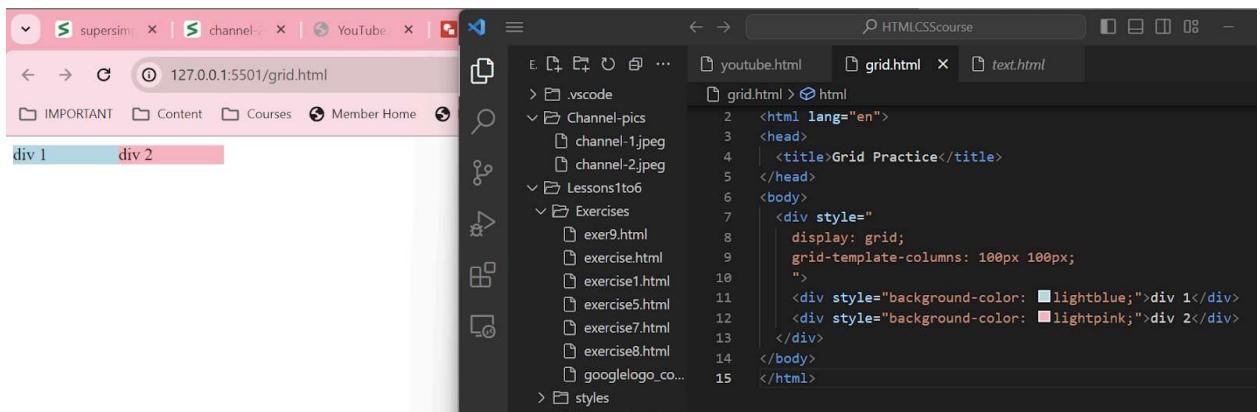
Inline style for practice.



A screenshot of a browser window titled "HTMLCSScourse". The address bar shows "127.0.0.1:5501/grid.html". The page content displays two div elements: "div 1" (lightblue background) and "div 2" (pink background). To the right of the browser is a code editor window showing the HTML code. The relevant part of the code is:

```
<html lang="en">
<head>
</head>
<body>
<div style="background-color: lightblue;">div 1</div>
<div style="background-color: pink;">div 2</div>
</body>
</html>
```

Creating 1x2 CSS grid

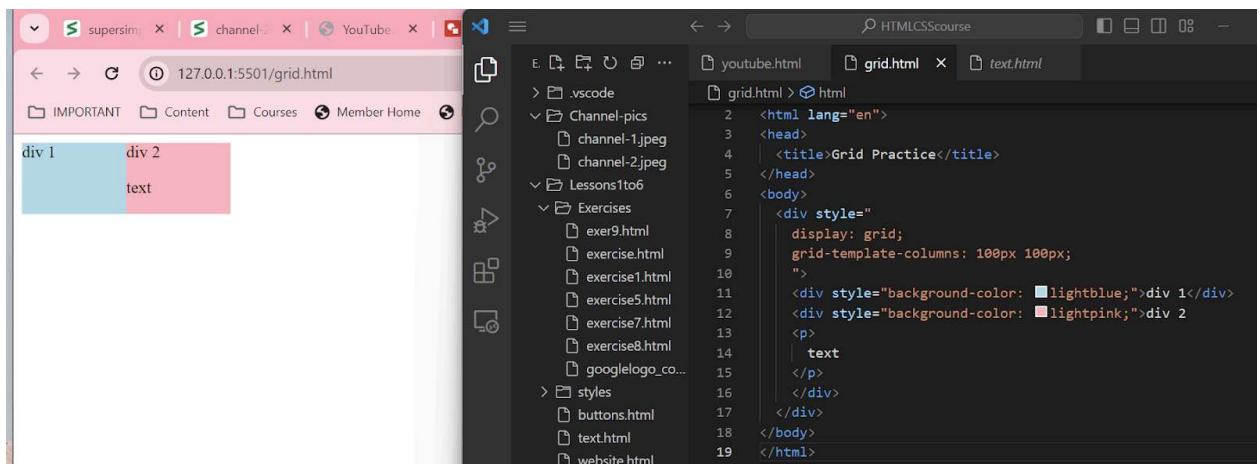


A screenshot of a browser window titled "HTMLCSScourse". The address bar shows "127.0.0.1:5501/grid.html". The page content displays two div elements: "div 1" (lightblue background) and "div 2" (pink background). To the right of the browser is a code editor window showing the HTML code. The relevant part of the code is:

```
<html lang="en">
<head>
<title>Grid Practice</title>
</head>
<body>
<div style="display: grid; grid-template-columns: 100px 100px;">
<div style="background-color: lightblue;">div 1</div>
<div style="background-color: pink;">div 2</div>

```

Maintains alignment even after adding paragraph tag.



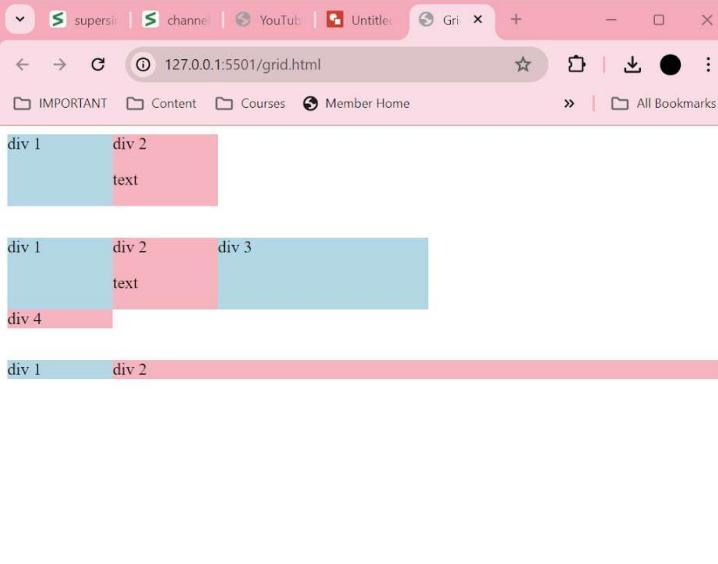
A screenshot of a browser window titled "HTMLCSScourse". The address bar shows "127.0.0.1:5501/grid.html". The page content displays two div elements: "div 1" (lightblue background) and "div 2" (pink background), which contains a paragraph element "text". To the right of the browser is a code editor window showing the HTML code. The relevant part of the code is:

```
<html lang="en">
<head>
<title>Grid Practice</title>
</head>
<body>
<div style="display: grid; grid-template-columns: 100px 100px;">
<div style="background-color: lightblue;">div 1</div>
<div style="background-color: pink;">div 2
<p>text</p>
</div>

```

Fr property - grid template columns attribute.

On the last grid in the below image, the fr property allows the 2nd column to use as much width as available(in the remaining space).



The screenshot shows a browser window with three examples of grid layout:

- Example 1:** A single row with two columns. The first column is light blue and contains "div 1". The second column is light pink and contains "text".
- Example 2:** A single row with three columns. The first column is light blue and contains "div 1". The second column is light pink and contains "div 2". The third column is light blue and contains "div 3". Below this row is a light pink box containing "text" and "div 4".
- Example 3:** A single row with two columns. The first column is light blue and contains "div 1". The second column is light pink and contains "div 2".

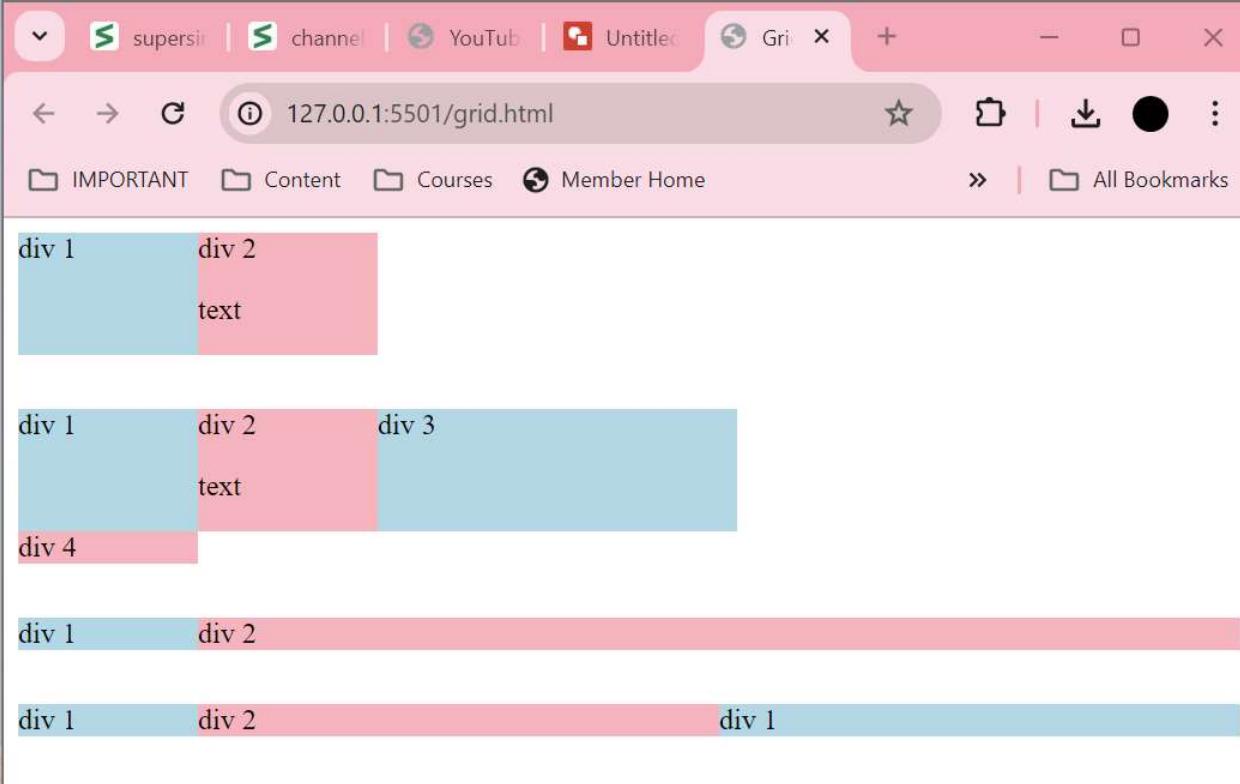
**Code:**

```

id.html > html > body > div
<html lang="en">
<body>
  <div style="display: grid; grid-template-columns: 100px 100px 200px; margin-top: 30px;">
    <div style="background-color: #lightblue;">div 1</div>
    <div style="background-color: #lightpink;">div 2<br/>text</div>
    <div style="background-color: #lightblue;">div 3</div>
    <div style="background-color: #lightpink;">div 4</div>
  </div>
  <div style="display: grid; grid-template-columns: 100px 1fr; margin-top: 30px;">
    <div style="background-color: #lightblue;">div 1</div>
    <div style="background-color: #lightpink;">div 2</div>
  </div>
</body>
</html>

```

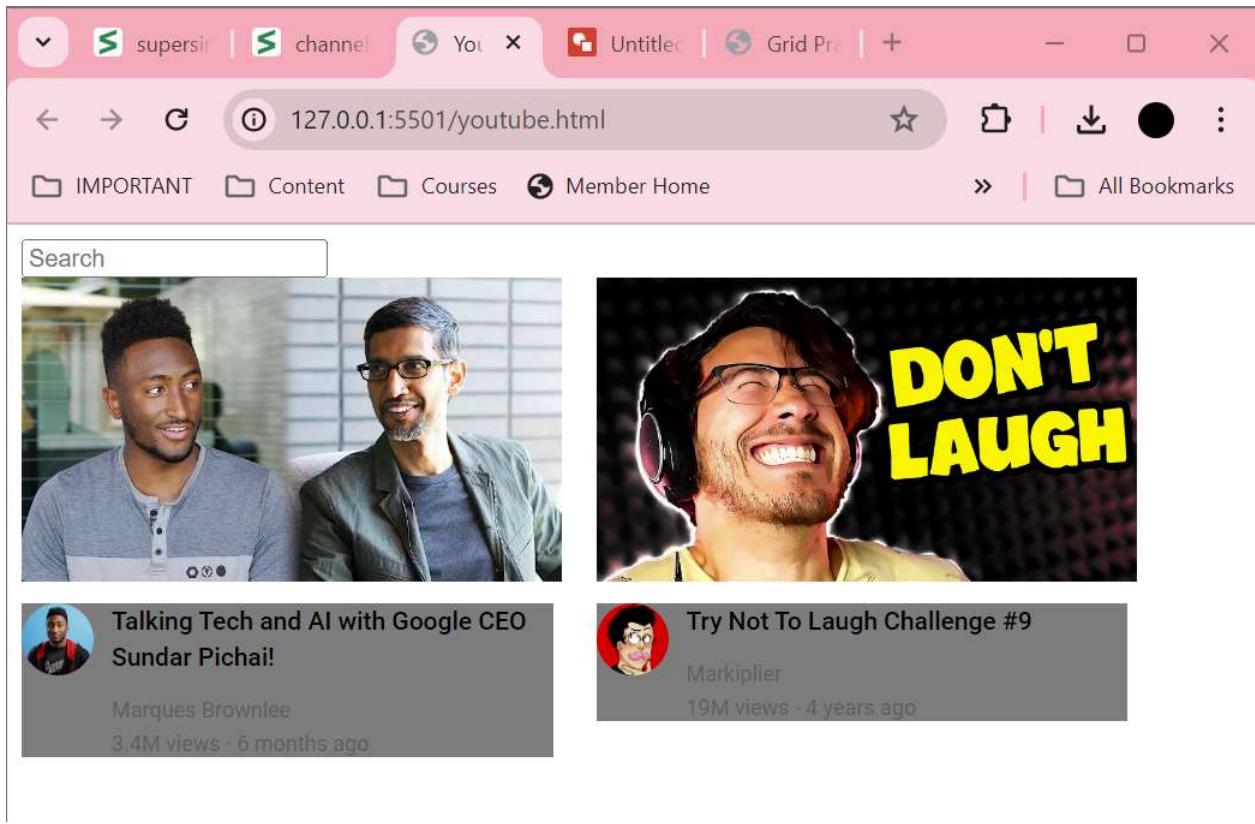
When 2nd and 3rd columns are assigned 1fr - the columns take up Equal amount of remaining space.



The screenshot shows a browser window with three examples of grid layout:

- Example 1:** A single row with two columns. The first column is light blue and contains "div 1". The second column is light pink and contains "text".
- Example 2:** A single row with three columns. The first column is light blue and contains "div 1". The second column is light pink and contains "div 2". The third column is light blue and contains "div 3". Below this row is a light pink box containing "text" and "div 4".
- Example 3:** A single row with three columns. The first column is light blue and contains "div 1". The second column is light pink and contains "div 2". The third column is light blue and contains "div 1".

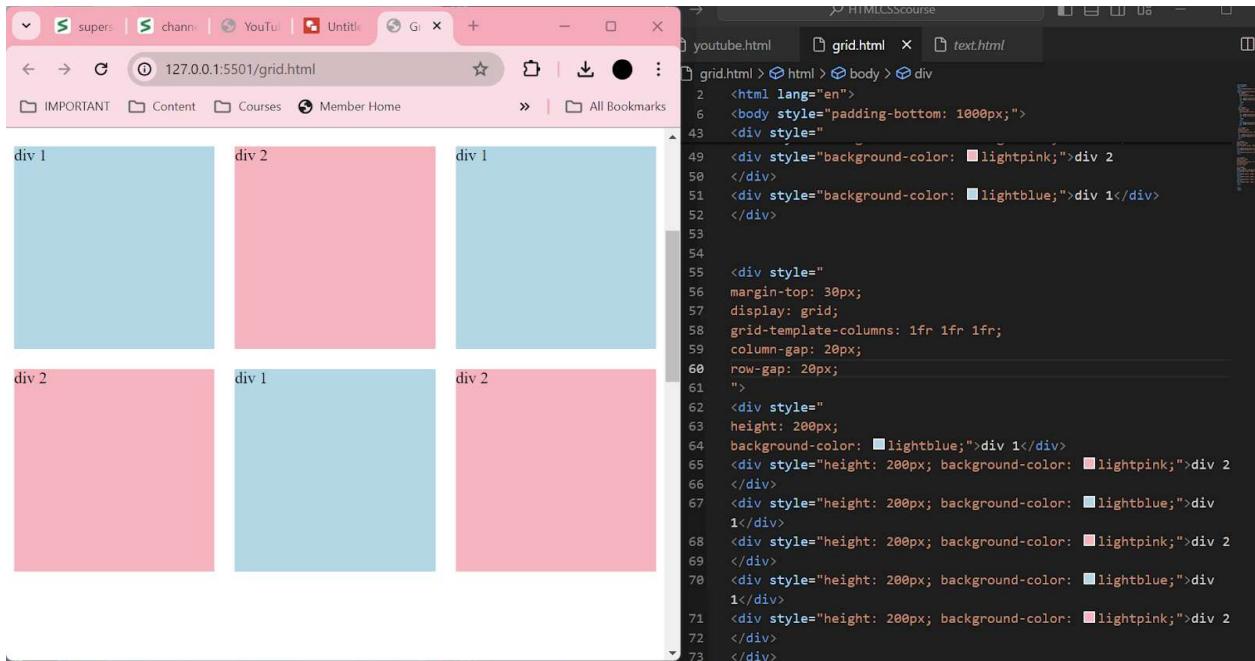
Using grids to align the text and profile pic in the video previews.



CSS grid 2x3 created for practice. And added column gap.

```
grid.html > html > body > div
2   <html lang="en">
3     <body style="padding-bottom: 1000px;">
4       <div style="
5         display: grid;
6         grid-template-columns: 1fr 1fr 1fr;
7         column-gap: 20px;
8       ">
9         <div style="
10           margin-top: 30px;
11           height: 200px;
12           background-color: #lightblue;">div 1</div>
13         <div style="height: 200px; background-color: #lightpink;">div 2</div>
14         <div style="height: 200px; background-color: #lightblue;">div 1</div>
15         <div style="height: 200px; background-color: #lightpink;">div 2</div>
16         <div style="height: 200px; background-color: #lightblue;">div 1</div>
17         <div style="height: 200px; background-color: #lightpink;">div 2</div>
18       </div>
19     </body>
20   </html>
```

Added row gap.



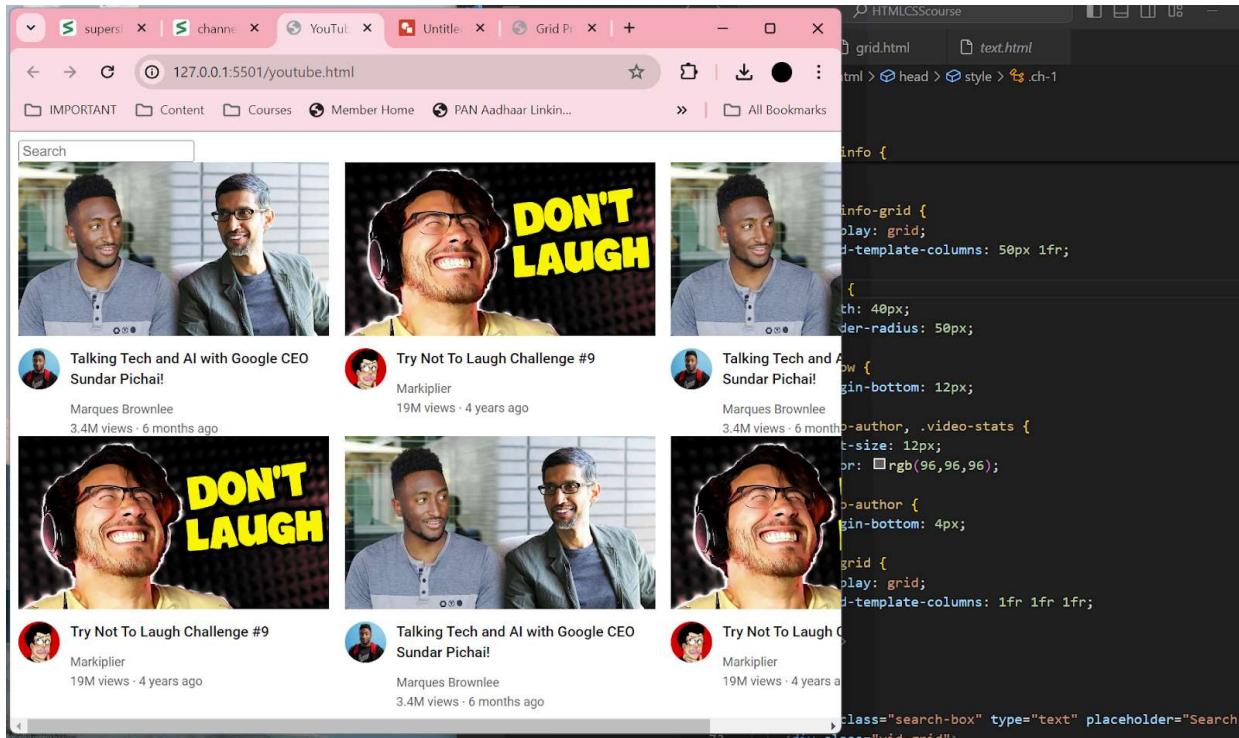
```
grid.html > html > body > div
2   <html lang="en">
6     <body style="padding-bottom: 1000px;">
43       <div style="background-color: #lightblue;">div 1</div>
49       <div style="background-color: #lightpink;">div 2</div>
50     </div>
51       <div style="background-color: #lightblue;">div 1</div>
52     </div>
53
54
55       <div style="margin-top: 30px; display: grid; grid-template-columns: 1fr 1fr 1fr; column-gap: 20px; row-gap: 20px;">
61         <div style="height: 200px; background-color: #lightblue;">div 1</div>
64         <div style="height: 200px; background-color: #lightpink;">div 2</div>
66         <div style="height: 200px; background-color: #lightblue;">div 1</div>
68         <div style="height: 200px; background-color: #lightpink;">div 2</div>
70         <div style="height: 200px; background-color: #lightblue;">div 1</div>
71         <div style="height: 200px; background-color: #lightpink;">div 2</div>
72       </div>
73
```

Applying these concepts to our youtube clone to have 6 video previews.

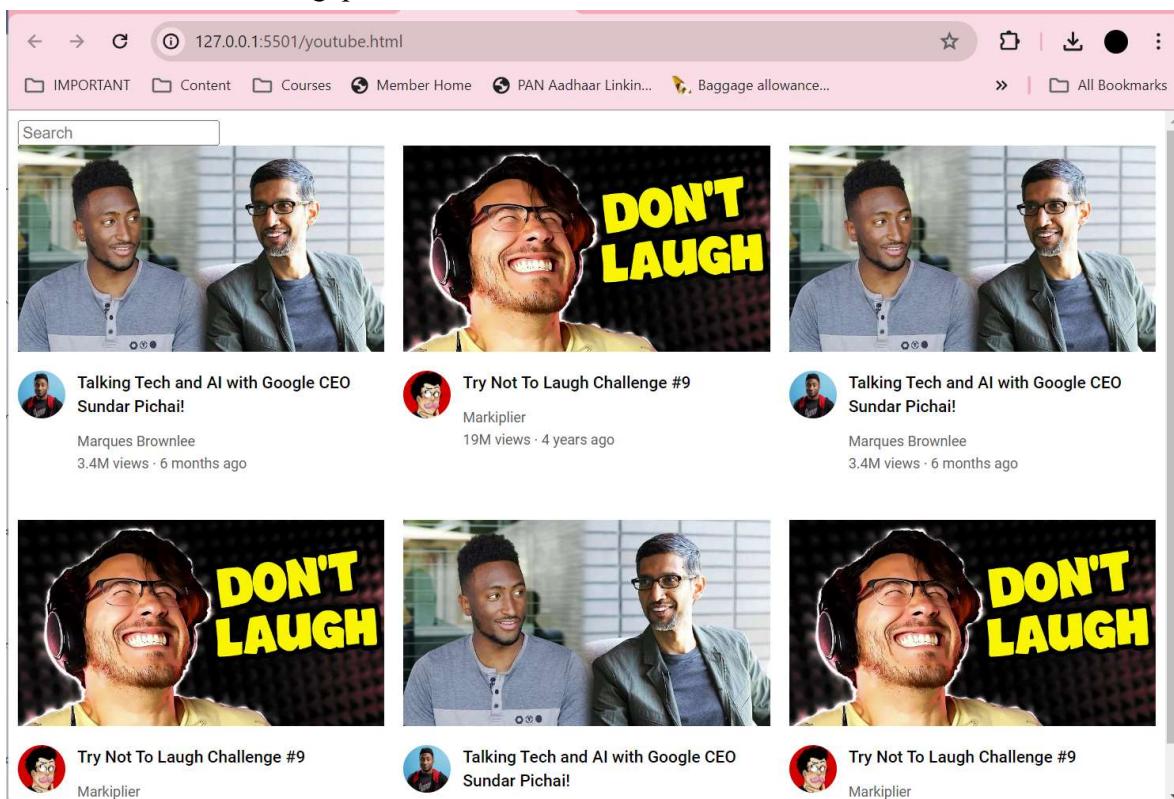
duplicated the existing previews for 2x3 grid



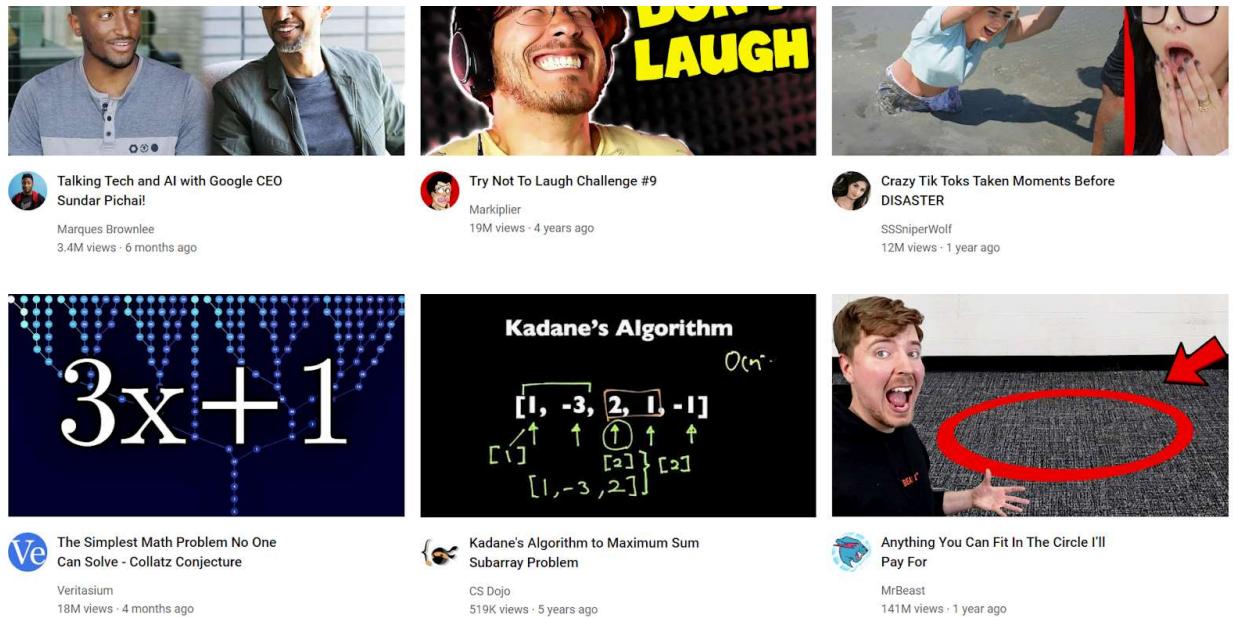
Created the grid and Assigned 1fr 1fr 1fr for the 3 columns to have equal amount of space in every column.



Added row and column gaps.



Added all the other 4 video titles, profile pictures, video authors and video stats.



**Next:** Finishing rest of the HTML and CSS course/tutorial. Continuing the Javascript course/tutorial.

## Appendix

### May 16, 2024 - Notes

- Questionnaire builder should have several different components/elements.
- Each element has potential answers, meaning validation might be required for most elements.
- First outline how the questionnaire should be structured and add basic elements and then, build on that.
- Use triple equals when comparing values (for example in if-else loop) in order to compare values of the same data type!!
- indexOf() gets the index of an element in an array.
- Break the project into small, manageable tasks and tackle them one at a time.
- Read Documentation- Utilize MDN Web Docs (Mozilla Developer Network) for JavaScript documentation and examples.
- Read up on the HTML5 Drag-and-Drop API to understand the events and methods that can be used.

- Key words:

Keyword	Description
var	Declares a variable
let	Declares a block variable
const	Declares a block constant
if	Marks a block of statements to be executed on a condition
switch	Marks a block of statements to be executed in different cases
for	Marks a block of statements to be executed in a loop
function	Declares a function
return	Exits a function
try	Implements error handling to a block of statements

### May 18, 2024 - Notes

- `rgb(255, 255, 255)` -> white
- `rgb(0,0,0)` -> black