**W4-S3** PRACTICE

*CSS FLEXBOX WITH REAL CASES*

## *At the end of this practice, you should be able to…*

* Apply a **top to** **bottom approach** to define your **CSS** **strategy**
* From mockups, **write your CSS code**
* From a customer need, define your **mockup** and your **HTML/CSS** **code**

## *How to work?*

* Download **the start code** from the LMS
* Open the folder in VS Code

## *How to submit?*

* **Push** your final codeon **your GitHub repository**
* Then **attach the GitHub path** to the MS Team assignment and **turn it in**

## *Are you lost?*

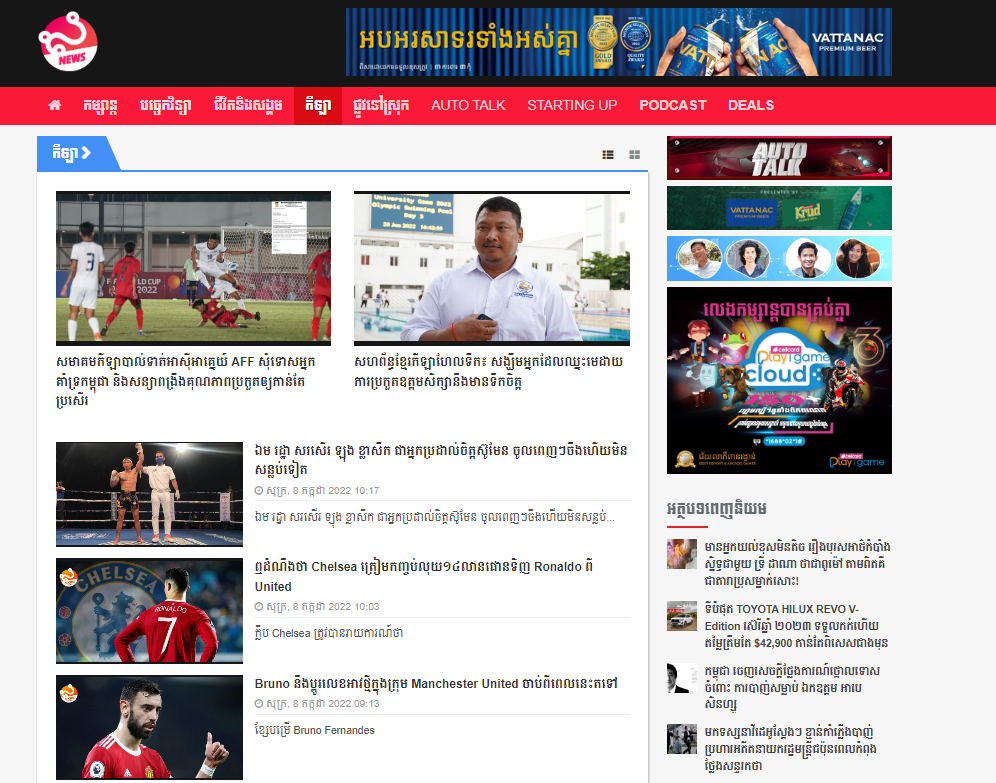
*You can read the following documentation to be ready for this practice:*

* [Guide - WC3 School](https://www.w3schools.com/css/css3_flexbox.asp)
* [Guide - Flexbox properties](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)
* [Video – Flex in 10 minutes!](https://www.youtube.com/watch?v=GteJWhCikCk&ab_channel=BroCode)

## *What is a top to bottom approach in designing CSS?*

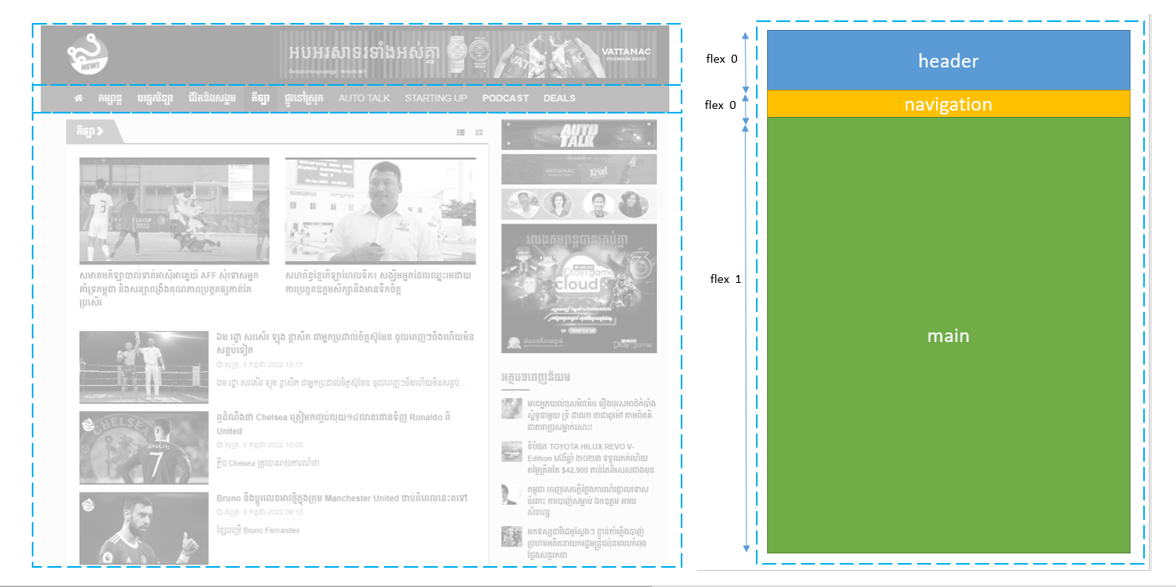
A Top-to-bottom approach in designing CSS is a method where you start styling from **the most general elements** and move towards the more **specific elements**.

😊 This approach helps in **dividing your layout tasks in smaller tasks!**

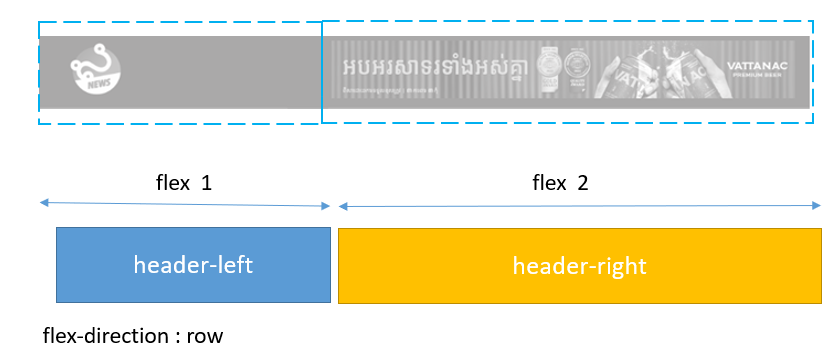


*Our example*

As example on the above page, we focus first on the main elements:

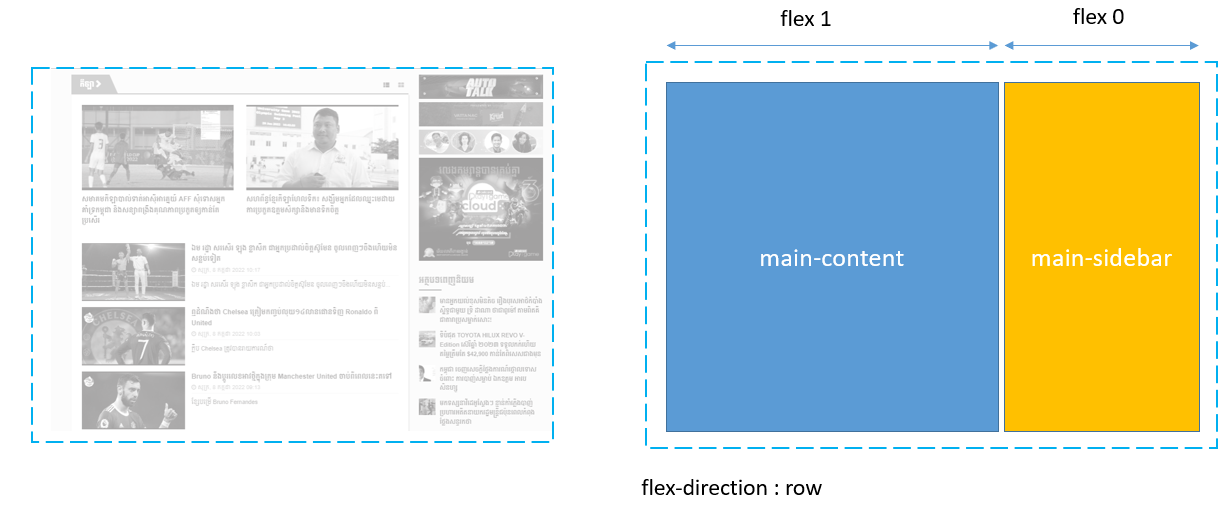


We then identify the sub containers:

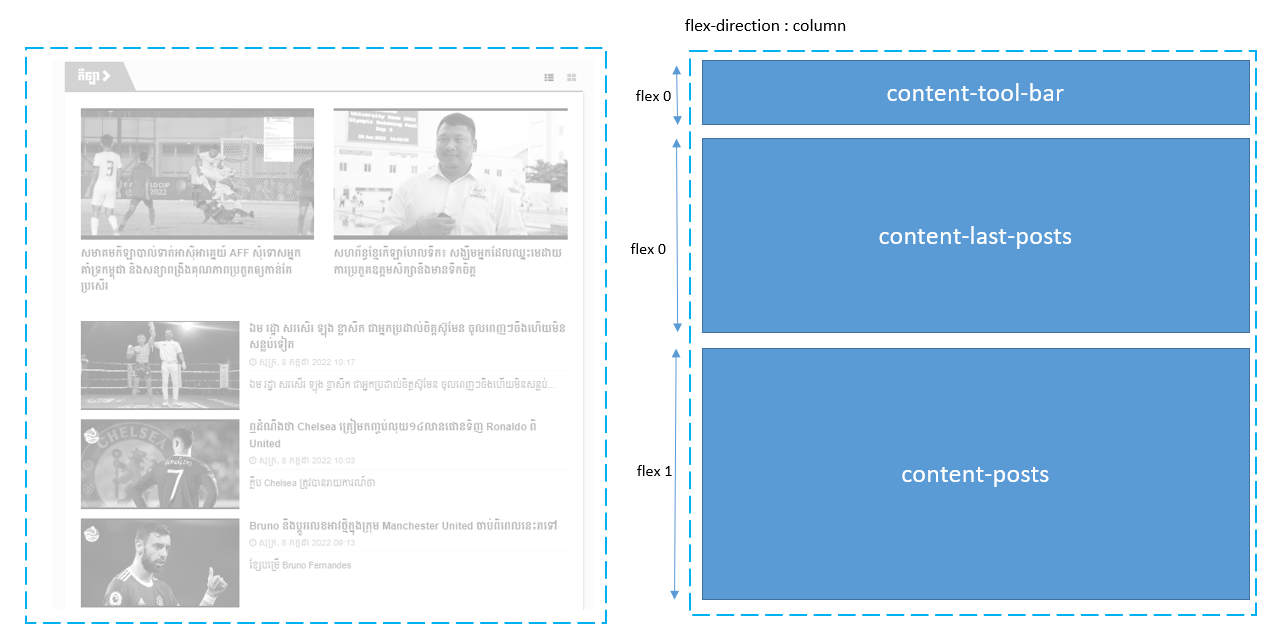


We also make sure to choose **appropriate container names**:

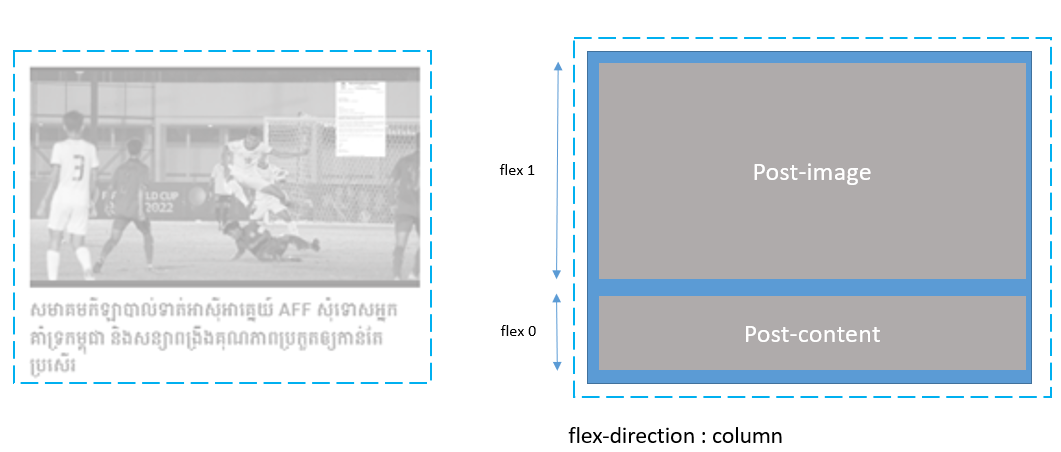
As example bellow, the main container is composed of the content and the sidebar containers



The content container itself if composed of 3 parts, as follows:



Finally, we define the deeper containers, such as the post item:



# EXERCISE 1

Identify the needed flex containers to implement the bellow mockup.

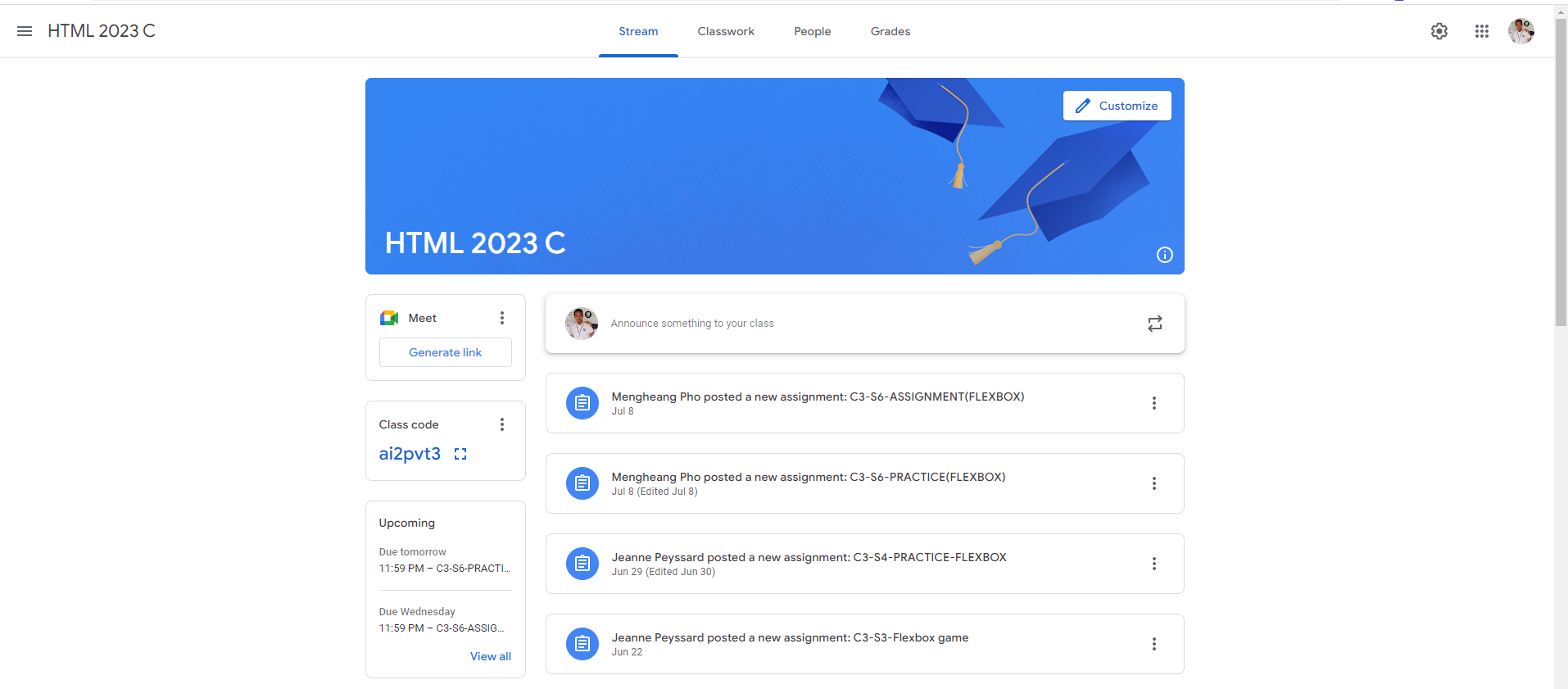
* Start with the **top containers first**, and then the deeper containers
* Use **flexbox layout** only
* **Name** each of your CSS container appropriately

Note

* No coding required for this exercise; we need you to **reflect on your CSS flex containers**

To start

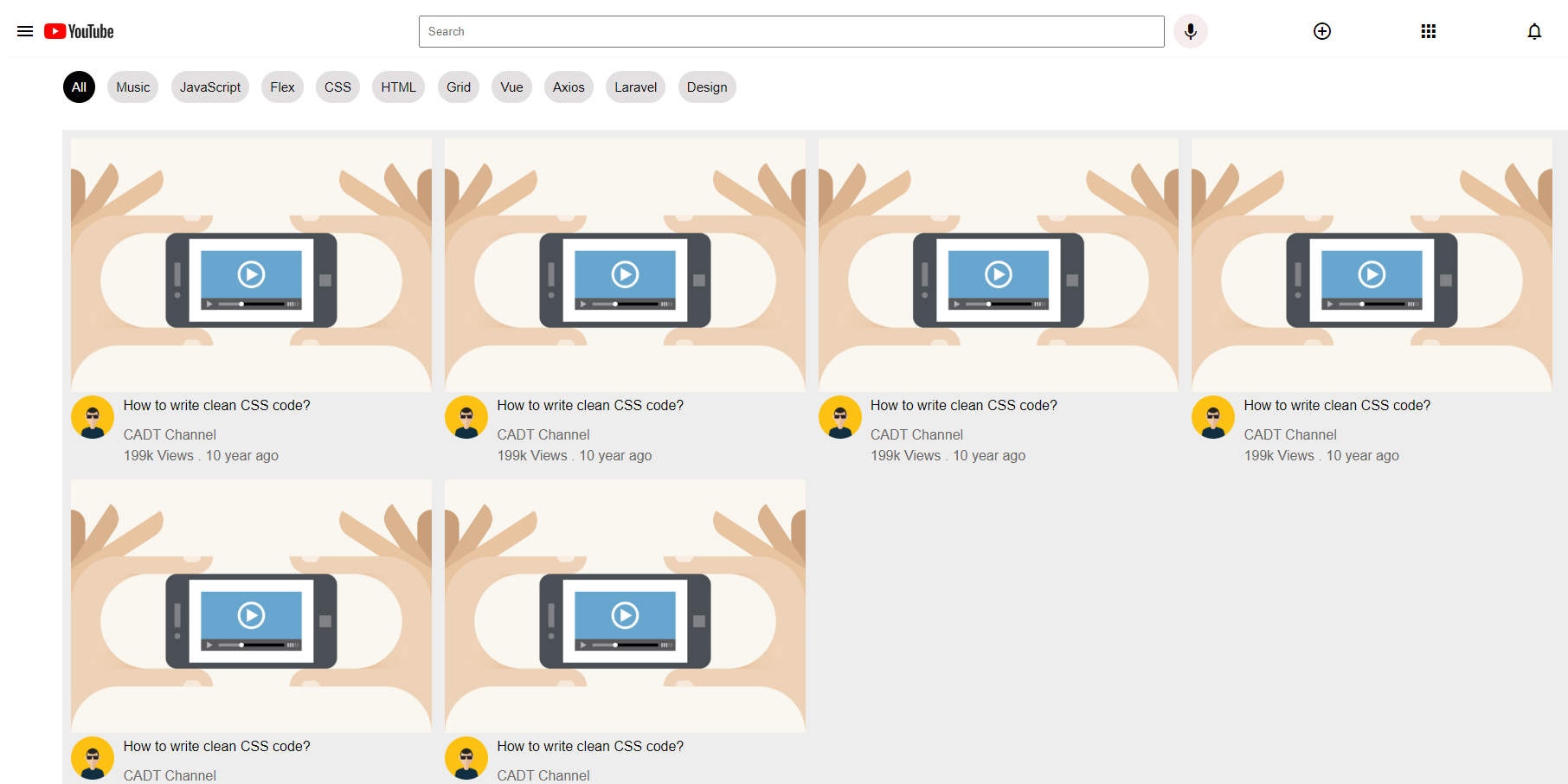
* You can start from the given **power point slides**



# EXERCISE 2

You need to update an existing code to comply with the below mock up

* Use **flexbox layout** only

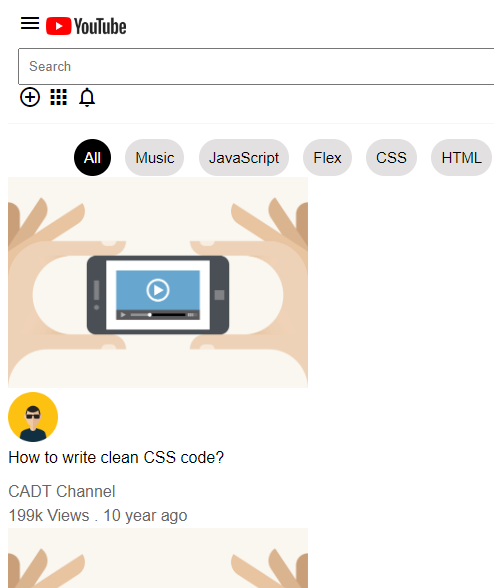
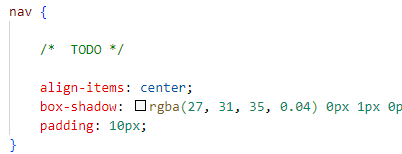


Hints

Before starting to code, **identify your needed flex containers** to match with the mockup

To start

* You start from the given code
* Nothing to change on the HTML code
* You only need to complete the CSS with Flex rules, **on every TODO section**

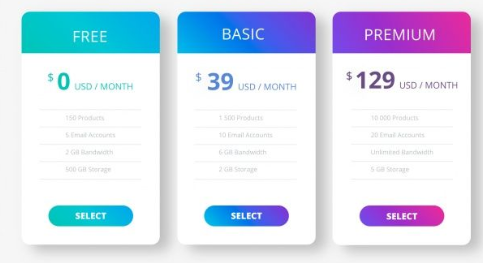
*Just complete the CSSS with missing CSS rules*

# EXERCISE 3

We're going to use everything we learned about Flexbox to **build a pricing table.**

*You've probably seen pricing table out on the Internet before.*

It's usually three categories small, medium, large, and it shows the customers which features they'll get based on which tier they're going to subscribe to.



You need to build a **pricing table with three items** (Basic, Standard, Premium) using Flexbox.

Constraints

* Use must use **flexbox** to layout your page
* Make it **responsive**: when the window goes below a certain size(smartphone), it actually stacks the items vertically rather than having them horizontally.
* Be **creative**!! You can style it however you like!

Hints

Responsiveness works with **media queries**!

* *what can you do, using media query and Flexbox attributes to achieve our needs?*

To start

* You can start from the **given index.html** start code, or make you own