

## Procedure

---

Follow the instructions below. You can download them as pdf [here](#). Build upon your solution from the previous challenge and upload everything for peer review when you're finished.

There are different exercise types:

- The exercises are important and you should tackle them.
- (\*) These asterisk-marked exercises marked are a bit more difficult and voluntary. They'll improve your skills, tackle them only if you want an extra challenge. However, your app will also work without fulfilling asterisk-marked exercises.

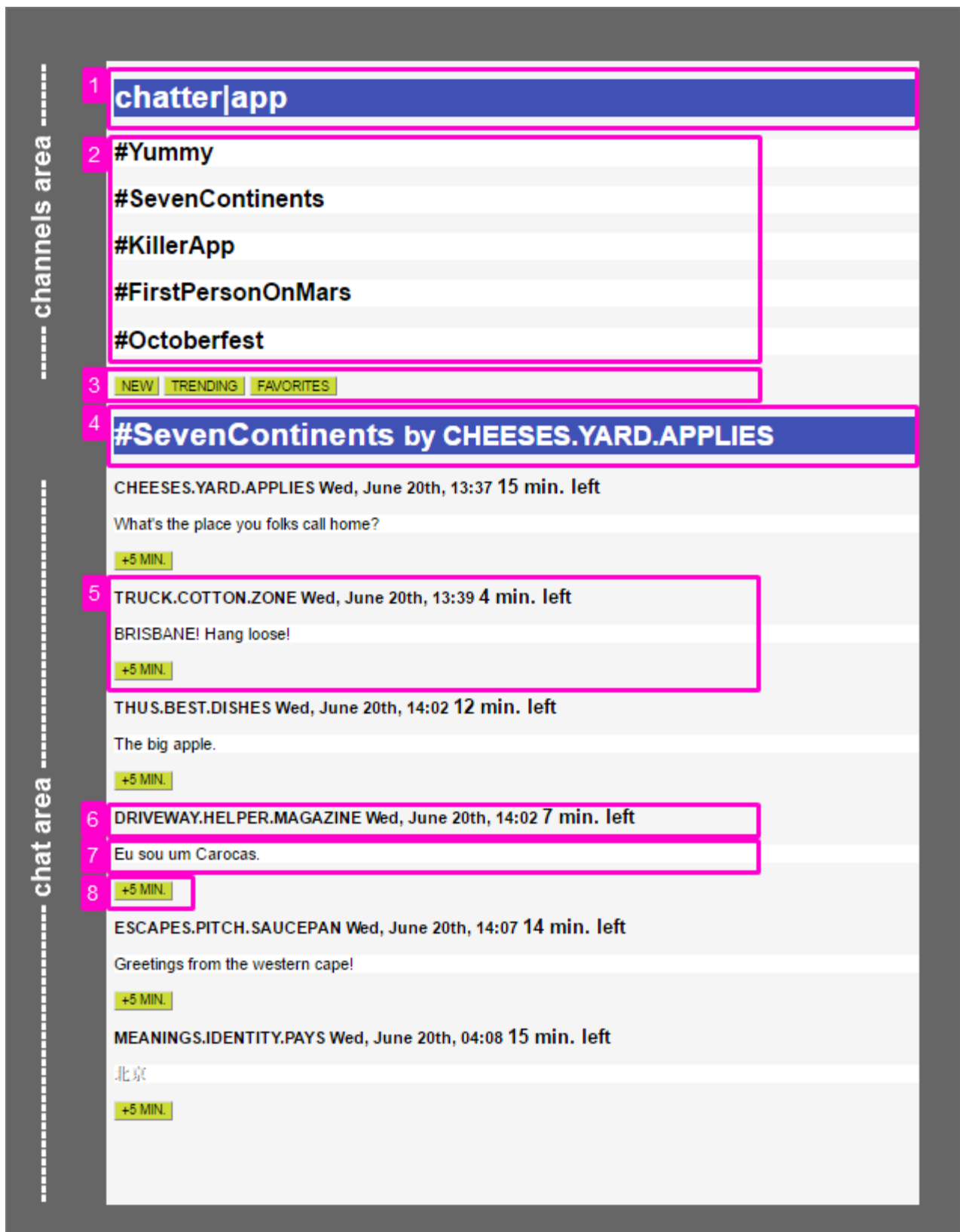
## Challenge Goal

---

In your second assignment you will give your chatter|app a colourful look by using elemental CSS styling tricks you learned in this section, for example...

- how to style HTML elements.
- how to address different parts of a HTML document.
- how to change colors of font and background.
- how to perform text transformations

Your app should be comparable to our sample solution after this challenge. It may look a bit different depending on which colours you picked.



1. App bar - chatter app
2. Channel list
3. Tab buttons
4. App bar - channel name
5. Message + Metadata + button
6. Metadata
7. Message
8. +5 min button

## Graded Criteria

You will be graded on how well your implementation meets the criteria below.

**#colorful:** A color scheme has been picked, applied and commented.

- 1 Pt. Picked color scheme (all six colors) has been documented.
- 1 Pt. Buttons are colored *accent*.
- 1 Pt. Both app bars are colored *primary*.
- 1 Pt. Background colored light grey .
- 1 Pt. Channels & chat text background is white.

**#font-faces:** The font-family has been set to Arial and sans-serif is used as a fallback option.

- 1 Pt. 'Arial' is used as default font family.
- 1 Pt. 'sans-serif' is used as fallback option.
- 1 Pt. All fonts look correct.

**#font:** Required text styles are comparable to sample solution.

- 1 Pt. Button text is in uppercase.
- 1 Pt. Font-size of metadata is medium.
- 1 Pt. Remaining time is larger & non-italic.

The **#syntax** will be validated by an automated syntax checker. If no mistakes are detected you get one extra point. Overall, a maximum of 12 points can be obtained.

## Instructions

---

### 1. Use internal CSS

- There are three ways to apply CSS to an HTML document. For this challenge, we strongly recommend to use internal CSS. It allows you to apply styles for the entire document, without having to embed an external CSS file.

### 2. Color your app

Use the following color palette (similar to [Google's material design guidelines](#)):

Function	Color	Usage
Background	The app's very light grey background	App
Primary text	Dark text on light backgrounds	Chat messages, channel names, ...
Primary color	Main contrast color	App bar, ...
Primary color text	Black or white	High contrast on primary color
Accent color	Color for interactive elements	Floating action button, +5 min. button
Accent color text	Black or white	High contrast on accent color

- Create a color palette and document it with CSS comments. Choose two different colors to improve design, structure and readability of your document. We call them *primary* and *accent* color. Check [materialpalette.com](#) for inspiration.
- Validate whether you should use black or white text for your *primary* and *accent color* with the [contrast ratio validator](#). Pick the one that has a better contrast ratio and use it as *primary color text* . You should have a contrast ratio of at least five. Do the same for your *accent color text*.
- Assign a very light background color. You may apply one of the Google material colors referenced in the table above.
- The two app-bars shall have *primary color* with the according text color.

- Apply a white background to all channel names and chat texts.
- The +5 min buttons in each message container are important interactive elements. Use *accent color* for the background and *accent color text* for text. You can assign the same colors to the channel buttons.

### 3. Make the text more interesting

- Change all fonts to Arial and use sans-serif as a fallback option.
- Change button text to uppercase.
- (\*) Find a way to transform links to w3w locations in **messages** to uppercase. Use a new [element](#) to address locations via CSS.
- Change the font size of the message metadata to medium.
- Make the remaining message time larger and non-italic.  
**Tip:** Do not modify the HTML but rather make changes in CSS.

### 4. Validate your code

- Clean up your code and insert comments for clarity and comprehensibility.
- Use the [W3C Validator](#) to find nesting and syntax mistakes.

### 5. Save your code

Do not forget to **save** your work, push your solution to Github, and **submit the repository link** for peer review.