

Procedure

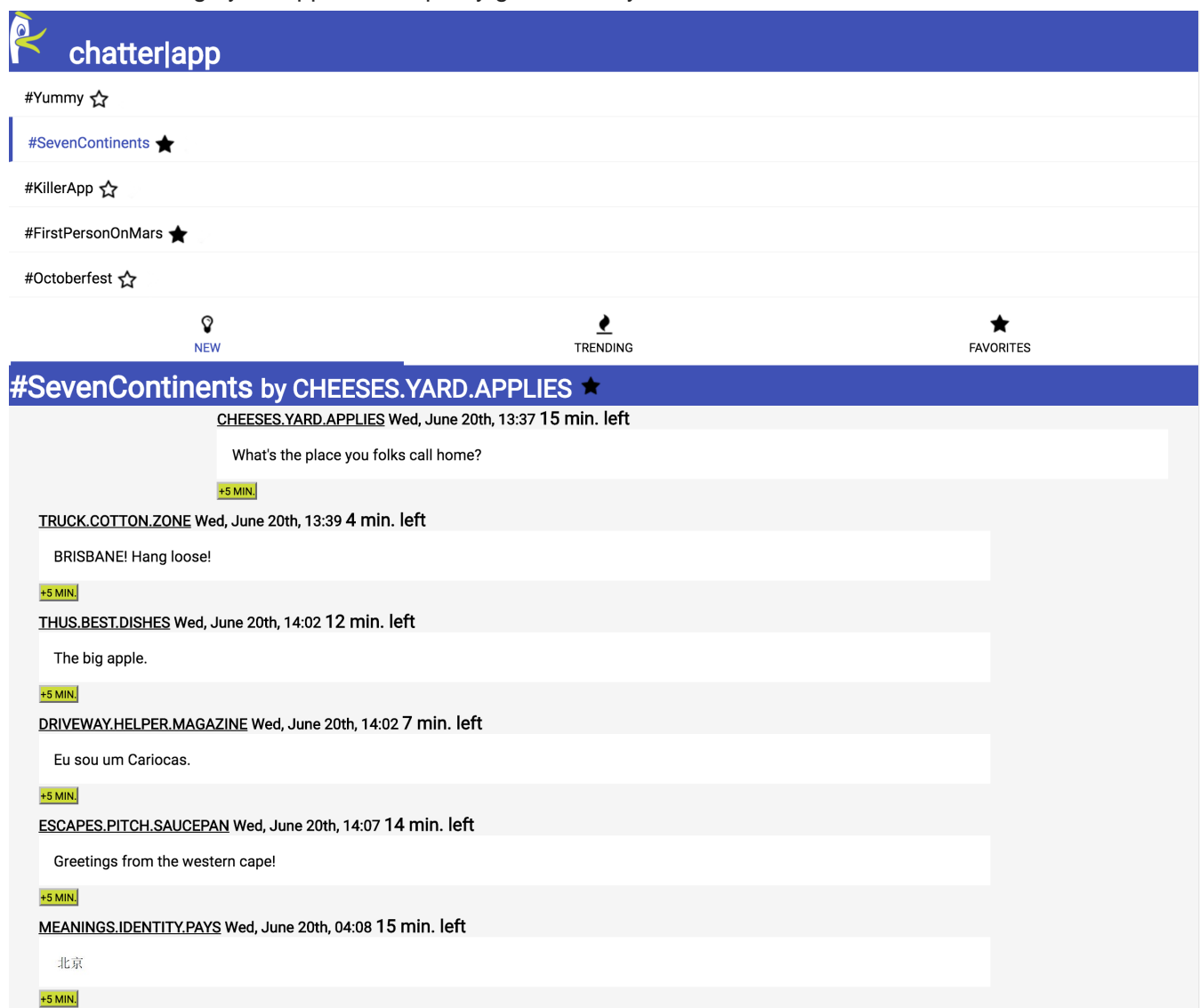
Follow the instructions below. You can download them as pdf [here](#). Build upon your solution from the previous challenge and upload everything for peer review when you're finished.

There are different exercise types:

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

Challenge Goal

After this challenge your app will look pretty good already:



Grading Criteria

#channel: Channel list comparable to solution & current channel highlighted.

- 1 Pt. Current channel is highlighted.
- 1 Pt. Channel list is comparable to solution. Compare tile height, font-size, all tiles should be same height.

- 1 Pt. Tiles separated by thin light-grey lines.

#tab-bar: Tab-bar buttons look comparable to solution.

- 1 Pt. Space between content (icon/text) and button border comparable to solution.
- 1 Pt. Button height comparable to solution.
- 1 Pt. Buttons distributed equally.
- 1 Pt. Tab 'New' highlighted.

#messages: Following requirements (size/ position) of Message containers are comparable to solution:

- 1 Pt. Vertical distance between message containers.
- 1 Pt. Width of message containers.
- 1 Pt. Alignment of 'all' messages.
- 1 Pt. Alignment of .own messages.
- 1 Pt. Space between chat text and paragraph border.

Your **#syntax** will be graded automatically. Overall, 12 Points (Pt.) can be achieved.

Instructions

1. Outsource your CSS

- Create an empty external style.css file in a CSS subfolder and link to it in your index.html. Divide your CSS file in five different "components" or "sections" by eye-catching CSS comments:
 - Global
 - App bar
 - Channels list
 - Tab bar
 - Messages
- Start with the global styles: copy&paste the app's background, capitalized buttons and the font from index.html.
- Remove ugly spacing between all components (document, headlines, ...). To do so, select **all elements** and reset their margins and paddings to 0. How do you select all elements? It's normal to forget, everyone does. Ask Google: 'select all elements'. And check out already recommended sources.
- Like your buttons, also links to w3w locations need to be uppercase. Select both elements simultaneously (remove redundancy...)!
- Continue with the app bars' CSS. You have two of them, an app-bar in each HTML file. Select them both by an element selector, like in the previous challenge and you will do just fine.
- Repeat the copy and paste procedure for your channels list styles.

Note that element selectors will no longer be distinctive for all of your app's components: The tab bar as well as all messages are wrapped in 's.

- To identify and style them individually, assign an id to the tab-bar. Copy&Paste the relevant styles from index.html and use the id selector. Use CSS to center the text inside the buttons.

As the tab bar turns up only once in your app it is appropriate to address it via an id selector. However, if you have a component which is used repeatedly, like messages in the next step, its more efficient to address it via one class.

2. Get your messages across

- Assign a class named message to all chat message containers (the one wrapped around metadata, text and button).

- Use all the message formatting from index.html to style your messages. Here is a reminder on the styles you have to apply, in order to make your message look like our sample solution. Below are tips on how to get there. However, please **do not copy&paste** these directly, it is not CSS syntax!
 - Message container (width 80%, margin-bottom 8px)
 - Message metadata (font-size medium)
 - Message text (background white, padding 16 px)
 - Message button (background CDDC39; black text color; transform text to uppercase)
 - Message emphasised information (font size larger; font style normal)

Tip: It will be easier if you distinguish between styles which apply to every message part and those styles which are relevant for specific parts of the message i.e. links. Start with styles, which apply to the entire message and insert them in the *class selector* you have just created.

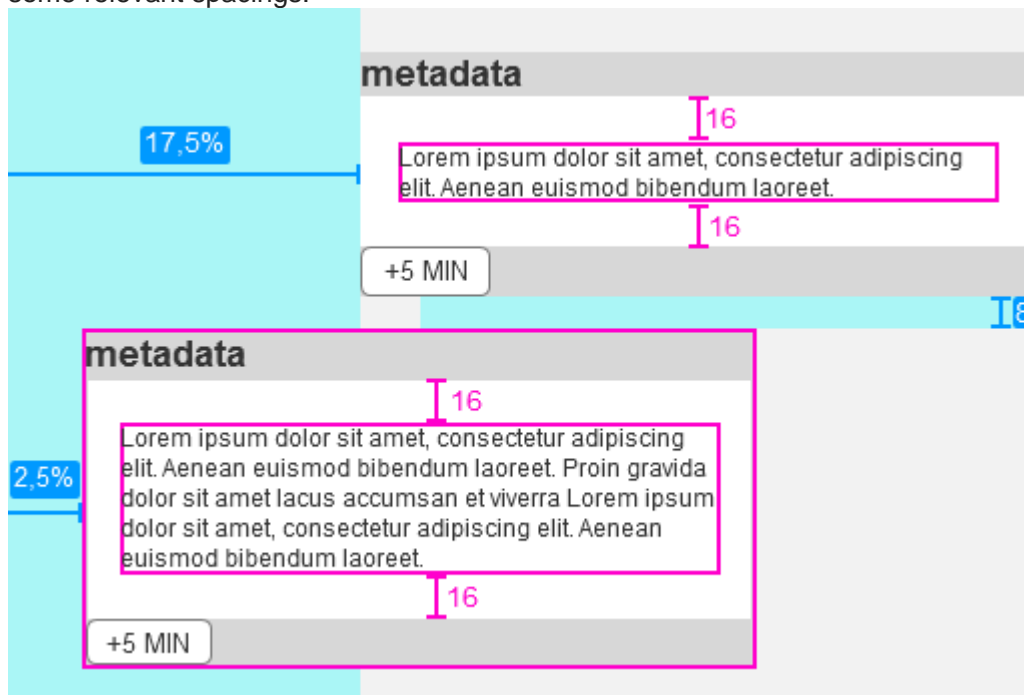
Now our message containers have several children:

, , ... and so on. You want to select those by their hierarchy within their parent element that is 'message'. For instance, there are w3w-links inside h3-headlines, which are nested inside the overall divs. So you tell CSS: Give .message h3 a {black text color}. And the link color should turn from default *blue* for links, to our primary text color *black*.

Also you need to understand the box-model. To really wrap your mind around what it does can be tricky. [Youtube](#) is as always a great source for information.

3. Box-model your messages

Now, let's check out the nitty gritty and work more on that box-model. Here is a principal layout with some relevant spacings.



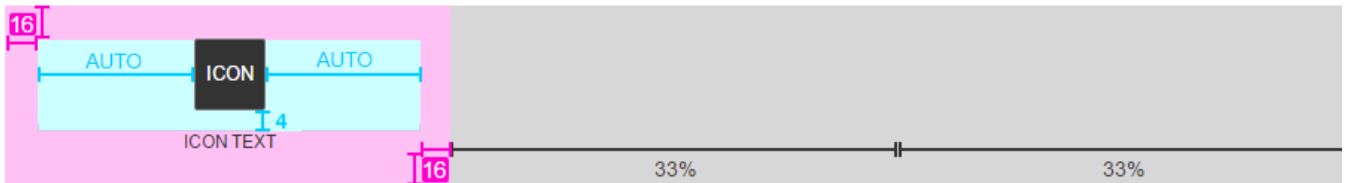
As you can see in the picture above your own messages are positioned further to the right, than messages of other users, as in Whats App for instance.

- Left-align *all* messages by using margins. Apply the spacings indicated in the picture above. You have to use a second class to align your "own" chat message.

4. Design the tab bar

- Let's have a look at the tab bar. We have to extract the styles for it as well. Select only the three tab bar buttons, which are nested inside the tab bar. So far they only had text-transform: uppercase; as style. Please add:
 - a white background
 - no border
 - a 13 px font size
- To make them look right tab bar's buttons should

- each claim one-third of the available width
- be 72 pixels high
- have a 16-pixels distance between its content, that is icon and text, and its own border.
- (*) The icons inside the tab bar buttons are currently inline. However, they should appear above the text. Use CSS to let them create a line-break. Google 'margin:auto' and use it to center the image horizontally. Spacing between icon and text is 4 px.



Compare your results in the browser with the image above. Use the inspector (WIN: F12 or Ctrl+Shift+I and MAC: Cmd+Opt+I). In the 'Style' Tab you see the box-model of each element in comparison to your HTML file in the 'Elements' Tab.

- The tab 'New' should be highlighted by default. The highlighting should change when someone clicks on another tab. We will do that later, with a bit of interaction. Right now assign a selected class to the button, for highlighting. Address the selected button in CSS. We have several buttons in our code, so select the tab-bar first, its buttons within and then narrow it down by adding .selected. Now add a border at the bottom and use primary color for highlighting. The border is 4-pixels-high.

5. Design the channel list

- highlight the color and left border of the currently selected channel using primary color, as you did previously in the tab bar.
- Each tile of the channels list should have a 16 pixels padding and a 16 pixels-high font. Also add the overall tile-height (to compensate the varying icon heights). You can calculate that by adding the padding- and font-measurements. Consider the box-sizing property.
- Separate the tiles by thin background-colored lines. You can do it, you have used that style before!
- (*) Currently, the (inline) icons are aligned to the text's baseline. For more information go to [MDN](#). Correct it to middle.

6. Try a new font

- (*) Embed and use Roboto ([click to select this font](#)) as your font. Generate a link that you can use in your document's head. [Google helps](#) to get started with the font. If you read a bit in the material design guidelines you will have noticed that they use the dp as value. We always converted it 1:1 to pixels.

7. Clean up

- Check your code & syntax. Using [W3C Validator](#) helps.
- Make useful sections and comments.

8. Save your work

Do not forget to **save** your work, **push** it to Github, and paste the URL in the submission mask. If you have questions concerning Github, consider the discussion board or look it up in the earlier sections.