Challenge 1

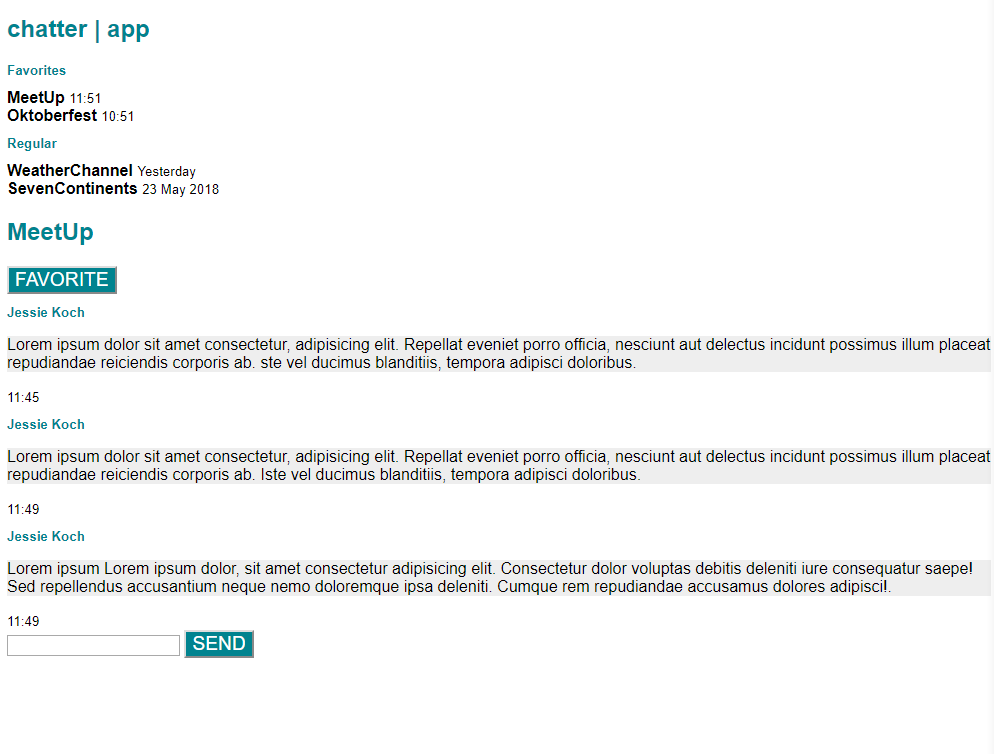
Procedure

As a first step, you will create a list of **channels** and a **chat** window.

You need to submit your code after the **challenge**. Follow the instructions below and **regularly save** your work on your local hard **drive**.

Challenge Goal

You will take small steps towards building a great looking interactive app. In this challenge, you will create an HTML document which looks similar to the sample solution below.



Graded Criteria

You will be graded on how well your solution meets the criteria. Follow the instructions carefully to receive full points. Be creative in finding solutions, however, keep in mind that your peers will grade your work according to the criteria below.

The grading criteria are:

**app bar**: Both app bars must have the same appearance (content and position) as in the sample solution.

* 1 Pt. One bar is at the top and one above above the messages.
* 1 Pt. One of the channel names is used for the second app bar.
* 1 Pt. Both app bars are colored correctly.

**buttons**: The two buttons look like the buttons in the sample solution.

* 1 Pt. Buttons background color is like the sample solution.
* 2 Pt. Buttons text is white and uppercase.
* 1 Pt. Buttons are located correctly.

**channels**: Channel list has the same structure and content as the sample solution.

* 1 Pt. Categories appear correctly.
* 1 Pt. Channel names appear bold.
* 1 Pt. Dates appear next to channel names.
* 1 Pt. Dates appear smaller.

**chat**: Messages have the same structure and content as the sample solution.

* 1 Pt. Message structure is correct, that is, username appears above each message and a date is located below.
* 1 Pt. Message background color is light grey.
* 1 Pt. Username is colored correctly.
* 1 Pt. Username and date appear smaller.

**font-faces**: The font-family has been set to Roboto and a fallback option is used.

* 1 Pt. ‘Roboto’ is used as default font family.
* 1 Pt. A fallback option is used .
* 1 Pt. All fonts look correct.

The **syntax** will be validated by an automated syntax checker. If no mistakes are found, you get one extra point. Overall, a maximum of 19 points can be obtained.

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

Instructions

**1. Structure your HTML file and add a title**

* If you haven’t done already, create a new html file inside a folder called *MYOA MOOC* and name it *index.html*. Declare and structure the HTML document like we’ve done before. The chatter app has two sections within the body, the channel area and the chat area. In a later challenge we will align them side by side. For now, it’s easier to place the chat beneath the channels.
* Add the title “chatter|app” to your document head.

**2. Create channels in the channel area**

* Add the app’s name as a heading 1 to the very top of your app.
* Add the two category names “Favorites” and “Regular” with the help of a h2 tag.
* Add four random channel names and assign them to one of the two categories. Each channel name should appear in bold letters. You can use a special tag for that.

***Tip:****Which HTML Tag would be appropriate? Ask Google or*[*w3schools*](http://www.w3schools.com/html/html_formatting.asp)*for something like: “HTML tag strong”.*

* Now you might be wondering why all channel names in one of the two categories are displayed in one line. There is an easy solution to this problem. Use the <br> tag to add a line break.
* Add some random dates next to the channel names. These dates should appear in smaller letters.

***Tip:****You can use the same ressource as before but this time you search for something like “HTML tag smaller”.*

**3. Create a chat history in the chat area**

* Pick one of the channels you came up with. The conversation in that channel is displayed in your chat area.
* Place the channel name at the top of the chat area with an appropriate heading.

***Tip:****The app will have two app bars at a later stage and both app bars are of equal importance. Thus, their HTML tag should be the same.*

* Add a button that says “Favorite”. This button will allow us to state a channel as favorite later on.
* Write a short, **fictional chat**. Include three random messages but not more than six.
* The message’s headline shows the corresponding username. Feel free to invent creative names for your chat partners and use a <h2> tag for them.
* Add the date when a message was committed below each message. Use a tag that makes elements look smaller.

***Tip****: You already used this tag for the dates in the channel section.*

* The actual **message content** should be displayed beneath the heading. Be creative and come up with your own content. Use the standard <p> tag for that.
* Add a <input> tag to be able to submit messages by yourself later on.
* Finally add a second button that says “Send”.

**4. 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.

**5. Style your app**

* Specify global styles that address all elements and specify the font-family to “Roboto”, “Arial” and “sans-serif” and this particular order.
* Set the color of the <h1> elements to “#00838f”. This is the hex code for a fancy turquoise.

***Tip****: There are several web pages that allow you to get different types of color codes like “RGB” or “Hex” for the colors you want.*

* Your <h2> elements should get the same color as your <h1> elements. In addition, the “font-size” is to become smaller.
* Now we want to style the incoming messages. Address the paragraphs and set the “background-color” to a light grey e.g. “#eeeeee”.
* The last element to style is the button. Make sure its background has the color “#00838f”. The text on the button should be **uppercase** and **white**. The “font-size” should be larger.

***Tip****: Feel free to google for something like “HTML text transform”.*

**6. Check your code**

* Check your code before you move on to the next challenge, especially for nesting and [syntax mistakes](https://validator.w3.org/) and secure an extra point.
* Clean up your code and insert comments for clarity and comprehensibility.

**5. Save your code**

Don’t forget to **save** your work as you will build upon your code in the next challenge.