

Procedure

As a first step, you will create a list of **channels** and a **chat** window. We recommend to install the following software to build the chatter|app:

- [Google Chrome](#) as a web browser,
- [Brackets](#) for editing, and
- [GitHub Desktop](#) for pushing code to your repository.

You can download the [instructions as pdf](#).

Tip: Copy & Paste the instructions as into your HTML documents.

After the second challenge you need to push (upload) your solution to your personal GitHub repository and submit the repository link. Follow the instructions below and regularly **save** your work on your local hard drive. You will build upon your code in the second implementation challenge.

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.

chatter|app

#Yummy

#SevenContinents

#KillerApp

#FirstPersonOnMars

#Octoberfest

New Trending Favorites

#SevenContinents by cheeses.yard.applies

cheeses.yard.applies Wed, June 20th, 13:37 15 min. left

What's the place you folks call home?

+5 min.

truck.cotton.zone Wed, June 20th, 13:39 4 min. left

BRISBANE! Hang loose!

+5 min.

thus.best.dishes Wed, June 20th, 14:02 12 min. left

The big apple.

+5 min.

driveway.helper.magazine Wed, June 20th, 14:02 7 min. left

Eu sou um Carocas.

+5 min.

escapes.pitch.saucepan Wed, June 20th, 14:07 14 min. left

Greetings from the western cape!

+5 min.

meanings.identity.pays Wed, June 20th, 04:08 15 min. left

.北京

+5 min.

Graded Criteria

You will be graded on how well your solution meets the criteria below. Follow the instructions carefully in order to receive all the points. Be creative in finding solutions, however, keep in mind that your peers will grade your work according to the criteria below. **If your solution doesn't meet the criteria, you won't receive full points.**

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 the messages.
- 1 Pt. A three word location, generated by <http://what3words.com/>, is used.

#buttons: The three buttons (new, trending, and favorites) look like the buttons in the sample solution.

- 1 Pt. Buttons are placed side by side.

- 1 Pt. Buttons are located below the channel list.

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

- 1 Pt. Message structure is correct, that is, metadata appears above each message and a +5min button is located below.
- 1 Pt. Metadata headline includes what3words location, weekday, date, time, time-to-live.
- 1 Pt. Time-to-life is emphasised.

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

Instructions

1. Structure your HTML file

- Your Github repository contains an empty HTML file (index.html). Insert the HTML document structure you have learned so far in this course. The chatter app has two sections, 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 also as a heading to the very top of your app. You will change it into an [app bar](#) later on.
- Add five random thread topics (channels).
- Add three buttons: **new**, **trending** and **favorites**.
Tip: If you're curious how to make your buttons look like those in the sample solution, either google 'HTML buttons' or visit one of our favorite websites [W3Schools](#).

3. Create a chat history in the chat area

- To keep the channels and the chat area apart, insert a horizontal rule ([hr tag](#)) or other content that separates them.
- 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. There will be an app bar at a later point, too.
Tip: The app will have two app bars and both app bars are of equal importance. Thus, their html tag should be the same.
- Add a **random location** (e.g., "by cheeses.yard.applies") next to the channel name, inside the future app bar. This three-word location shows where the channel was created. You generate a three-word location of your choice by using [what3words](#). Make the location *smaller* by using an appropriate tag.

Tip: Which HTML Tag would be appropriate? We haven't covered it yet. Ask Google or [w3schools.com](#) for something like: "HTML tag smaller".

- Write a short, **fictional chat**. Include three random messages but not more than six.
- The message's headline is composed of its metadata (location, weekday, date, and time). Your chat messages should be posted from at least two different locations.
- Add a **time to live** (ttl) to the heading. The time indicates when the message will be deleted. We consider this information as more important than location, date and time, so please *emphasize* it by using an [appropriate tag](#).
Tip: Use a heading tag for the whole line and nest the **ttl** with the appropriate *emphasize* tag.
- The actual **message content** should be displayed beneath the heading. Be creative and come up with your own content.
- Finally, attach a **+5 min** button to each message in a new line. The button's purpose will be to extend the message's time to live by five minutes later in the course.

4. Check your code

- Check your code before you move on to the next challenge, especially for nesting and [syntax mistakes](#) 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.