challenge06.md

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 assessment when you're finished.

There are different exercise types:

- These exercises are important and you should tackle them
- (*) These asterisk-marked exercises marked 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

Graded Criteria

#channels: Function switchChannel() changes app bar's channel name and location (as a link) to selected channel and logs the changes.

- 1 Pt. Click on channel name in list sets name of selected channel in app bar above chat area.
- 1 Pt. Click on channel name in list also sets a location for the channel in app bar.
- 1 Pt. Updated channel location link works.
- 1 Pt. New channel name is logged to console.

#star: Star in app bar turns to 'star-o' when channel changes and to 'star' when star is clicked.

- 1 Pt. Onload star in app bar is filled.
- 1 Pt. Click on channel sets star in app bar to 'star-o.png'.
- 1 Pt. Click on star in app bar changes the star from empty to filled.
- 1 Pt. jQuery was used to create the function.

#highlighting: Function removes highlighting from tabs and highlights clicked tab.

- 1 Pt. Click on tab removes highlighting from all (other) tabs.
- 1 Pt. Click on tab adds highlighting to that tab.
- 1 Pt. Clicked tab is logged to console.
- 1 Pt. jQuery has been used to create the function.

#emojis: Emoji box is hidden onload and toggles onclick.

- 1 Pt. Emoji menu is hidden onload.
- 1 Pt. Menu appears and disappears when smiley button is clicked.
- 1 Pt. jQuery was used to create function.

Your **#syntax** will be graded automatically. Overall, 15 points can be obtained.

Instructions

1. Start the action

- Create an external script.js file in a "js" subfolder and make sure it's embedded in your index.html. The first thing you do here is use console.log(''). What you write inside the parenthesis will be displayed in the console of your browser, when you execute this bit of JavaScript. Tell your console "app is alive", to find out if your JavaScript is working!

2. Switch channels

If the user clicks on a different channel, a function shall change the name in the right app bar and log the new channelName to the console. If you want to use your own approach to change the channel name, go ahead. We are always exited to see new ideas.
The next two steps explain how we did it. We start to build the function and explain afterwards how it will be invoked and thus executed.

- Write a switch function switchChannel(). Use channelName as parameter. You need to do two things:
    - Print 'Tuning into channel', channelName to the console.
    - Use document.getElementById('...').innerHTML=...; to change the channel name inside the right app bar. Only the name, not the entire app bar content should change. To get there you have to use an appropriate selector.
- We will pass an argument when calling our new function. Make sure the function switchChannel() is called if someone clicks on a channel in your list. Depending on the selected channel name, it shall exchange our parameter channelName with the actual channel name (i.e. 'Yummy'). **Thus, you pass the channel name as your functions argument** and call the function by using an onclick event.
- Upgrade the function to also change the app bar's location. After clicking on a channel the location should change to **'upgrading.never.helps'** . Use the same logic as in the previous step. Make certain that the location is a hyperlink and someone who clicks on the new location is directed to the corresponding w3w site. Ensure that the CSS styles that apply to the initial location also apply to the new one. Do not forget to insert 'by' before the location as well as a space to separate the words.
- Fix the link color in CSS to primary-text.
- Test the function in your console by calling it.

3. Favor channels

- Add a filled to the very right (16px from the edge) of your channel's app bar. Use HTML and CSS.
- Now embed jQuery. jquery can be embedded using various ways, right now we recommend to just directly link to the jQuery CDN. Be careful to use the latest and minimized version (min.3.1.1.).
- Next step: write a function that changes the app bar's star when clicking on a channel back to .
Tip: Use jQuery to select the star and change its source attribute, whenever a channel list-item is clicked. You will need to use the jQuery equivalent of document.getElementByID('...') and the attr() method. How should you invoke that function? You already have an onclick listener attached to each channel!
- Add a separate function to turn the **app bar's star** from unfilled back to filled when you click on it.

Test your functions: If you load the page, the app bar's star is filled. Click on a channel, the app bar's star is unfilled. Click on the star in the app bar, it should now turn to a filled star again. We will work on toggling the star and saving what the user liked in the next challenge.

4. Tap tabs

- We need a new function to switch between the tab buttons. Call it selectTab().

The function shall add and remove the class *selected* every time it's invoked. You can use the approach described below, or tackle the task on your own.
Tip: You can reuse the logic from the switchingChannel() task. So you need: An onclick listener with arguments. For your function a parameter (we suggest *tabId*) which takes a passed argument. Also appropriate jQuery methods, which will use the argument as a $("selector") to add and remove the class *selected*, come in handy.

- Use a nested selector to remove the class *.selected* from all buttons in the tab bar. You will need jQuery's removeClass() method. Check jQuery's documentation if necessary.
- After you removed *.selected* from all buttons, you add it to the one the user clicked on. You select the element with addClass(). But how to tell jQuery which button the user just clicked on? Well, every button has an ID and if you click on a button the ID should be passed as an argument to selectTab() and recognized by the jQuery selector. So you should give each tab button an **appropriate ID**.

- To test your function log the passed argument. Print 'Changing to tab', tabId to the console.

5. Toggle emoji box onclick

- Add a piece of code that hides the emoji box initially.
  Tip: A CSS property which specifies if an element is displayed or not can be handy here.
- Use **jQuery** to show the emoji box on click.
- The box should not only appear onclick, but also vanish onclick again. With other words, it should toggle when a user clicks the button. Again jQuery provides the functionality you're looking for.

6. Switch channels pt. II

- (*) You are already able to highlight tabs when switching them. We shouldn't use IDs for channel switching (because there will be many, and names could overlap with our ids). We'll use a more efficient way later on, but for now, you could apply a jQuery content filter selector. The selector searches, within the selected element, for whatever is inside the brackets. You can also filter for variables or parameters, as we would like to. However, you should be careful not to handle them as you would regular strings, for example:

  See how the parameter is not used as a string but added. It is called concatenation. Do not simply copy the code! It is just example. Just give it a try and modify your switchChannel() to highlight only the selected channel, without modifying the HTML...

7. Clean up

- Ensure that you have good structure and that you used sufficient comments.
- Check code and syntax. Using W3C Validator or another validator - there are many more out there - helps.

8. Save your code

Do not forget to **save** your work, **push** it to Github, and paste the repo URL in the submission mask.