Code & Go

JavaScript

Day 04 - 02

Teaching managers

pedago-coding@epitech.eu

Table of contents

Foreword

> **In all the following exercises, the instructions must be executed after the full loading of the page.**

Remember that you're coding in JQuery, so you must use the JQuery alternative to do something, and not pure JS.

Exercise 01 (3Pts)

File to hand in: JavaScript_D04_02/ex_01/

Create an input allowing someone to enter a text. When the text **is considered valid** add it to a bulleted list displayed under the input.

Exercise 02 (3Pts)

File to hand in: JavaScript_D04_02/ex_02/

Define a few CSS classes:

- A "note" class, putting the border of the element in blue
- An "email" class, putting the border of the element in green
- A "todo" class, putting the border of the element in red

It should now be possible to precise one of this 3 types (note, email, todo) when validating the input, you're free on the way you implement this and you can add more style to your css however if an input of type "email" is created the element displayed with the content must possess the class CSS "email". It will work the same way for the 2 other types.

Note: You may need to add some different check to validate the inputs depending on the type ☺.

Exercise 03 (3Pts)

File to hand in: JavaScript_D04_02/ex_03/

You will now implement a search in the elements created.

As of now you only need to handle a search by types. You will implement a search by words in the following exercise.

If I search for "email", only the elements being "email" must stay displayed. The others **must not be deleted** and must be shown again when the search is over (Once again you're free for the implementation).

Exercise 04 (4Pts)

File to hand in: JavaScript_D04_02/ex_04/

You will upgrade your search possibilities. It should now be possible to search by a word or a part of word contained in the elements.

For example if I search for "rick" and I have an email "patrick@something.com" and a "todo" with the value "Send a mail to Rick", they both must be displayed.

Moreover, you should add the possibility to combine your search with words and types, so in the previous example if I also choose to display only "todo" only the "Send a mail to Rick" should be displayed in the end.

Once again, there should be no element destroyed, and it should be possible to switch from a search to another without the need to reset everything. (For example if I selected "todo" and "rick" in my search but now decided I want all the types I should not need to erase everything and re-do my selection).

Exercise 05 (2Pts)

File to hand in: JavaScript_D04_02/ex_05/

Let's upgrade your input a little. Make it possible to add tags to an element. The tags must also be displayed. It should be possible to add multiple tag to an element, even after its creation.

Exercise 06 (5Pts)

File to hand in: JavaScript_D04_02/ex_06/

Let's upgrade our search engine a little more. It should now be possible to search for "tags", but also for two types of elements at the same times (email and todo for example).

Moreover, you will add the possibilities to search for multiples tags at the same time. It should also be possible to precise if the tags must all be included or not.

Example:

I have a "todo" with the following tags: "work", "tired", "dailyLife".

An "email" with the following tags: "work", "boss".

And a "note" with the followings tags: "boss", "money".

If my search for tags say "work", the "todo" and "email" must be displayed.

If I know search for "work" and "boss" (and both must be included), only the "email" will be displayed at the end.

And finally, if I search for "dailyLife" or "money" (one or the other), the "todo" and "note" must be displayed.

We do not ask you to handle cases like this: ("work" && "boss") || ("boss" && "money")

And all the more complicate cases you may imagine, however if you do handle those cases, you may be awarded bonus points.

Bonus

Here is a little idea for a Bonus.

Add a bit of PHP for the backend, and make it possible to save all the elements created with all the fields in a database by calling the PHP.

It would also now be possible to leave the page and come again to see all the elements having already been saved be displayed again thanks to the data being now durable.