Lab instructions

As laboratory work, you should work alone and complete the 4 small labs described in this document. When you are finish with all four, show it for a teacher at a lab session. Be prepared to answer any question the teacher might have about your code.

Each lab below makes use of an HMTL file. You may not change the HTML code written in these files, but feel free to use as much CSS and JavaScript code you want to complete the labs.

Before you start working on the lab you are recommended to:

- View/Take the following videos/tests:
  - Client-side JavaScript & BOM
  - Document Object Model

Lab 1: Generating a table of contents

The file `lab-01-table-of-contents.html` contains some information about Jönköping University from Wikipedia. Your task is to write JavaScript code that generates a table of content for the document (i.e. a list with the title of all the headers).

Some DOM functionalities you might find helpful:

- `document.addEventListener("DOMContentLoaded", aCallbackFunction)` to tell the web browser to call `aCallbackFunction` when it is done parsing the HTML document.
- Calling `querySelector("aCSSSelector")` on `document` or `anHTMLElement` to find the HTML element matching `aCSSSelector`.
- Calling `querySelectorAll("aCSSSelector")` on `document` or `anHTMLElement` to find the HTML elements matching `aCSSSelector`.
- `anHTMLElement.innerText` to read out/change what is written between the start and stop tag of `anHTMLElement`.
- `document.createElement("tagName")` to create a new HTML element of type `tagName`.
- `anHTMLElement.appendChild(aChildElement)` to add `aChildElement` to `anHTMLElement`.

Lab 2: A calculator

The file `lab-02-calculator.html` contains a form through which one should be able to enter two operands and one operation. Your task is to write JavaScript code that displays the result of the selected operation applied on the two entered operands.

Some DOM functionalities you might find helpful:

- `anHTMLElement.addEventListener("eventName", aCallbackFunction)` to tell the web browser to call `aCallbackFunction` when the event `eventName` happens on `anHTMLElement`. You are probably especially interested in the `input` event (for `<input>`,

`<select>`, …) and the `submit` event for (`<form>`). For the `submit` event, also remember to use `event.preventDefault()`.
- `anInputElement.value` to read out/change the entered value in `anInputElement`.
- `aSelectElement.value` to read out/change the value of the selected `<option>` in `aSelectElement`.

Lab 3: Validating input to forms

The file `lab-03-form-validation.html` contains a form the user can use to enter information about a new account. Your task is to write JavaScript code validating the data entered in the form when the user submits it:

- The email field must contain the @ symbol.
- The first password field must contain at least 8 characters.
- The second password field must be equal to the first password field.

If something is wrong, display the error messages on the screen (in the HTML code, it is not OK to use the `alert()` function). Otherwise (if everything is valid), just let the form be submitted.

Some DOM functionalities you might find helpful:

- `aString.length` to get the number of characters in `aString`.
- `aString.includes(aSubString)` to check if `aString` contains `aSubString`.

Lab 4: Tabs

The file `lab-04-tabs.html` contains three forms:

- One form to create a new account.
- One form to sign in to an existing account.
- One form to subscribe to updates.

Although all three forms are useful, displaying all of them at the same time does not make sense since the user is only interested in using one of them. Your task is to write JavaScript code that only displays one form at a time, and by clicking on links the user should be able to switch which form that is shown.

Some DOM functionalities you might find helpful:

- `anHTMLElement.classList` to add/remove/toggle classes `anHTMLElement` has.