Web Technologies I

Practical class:
JS II

Find the slides @ E-studijas (JavaScript section)

Darja Solodovņikova – darja.solodovnikova@lu.lv
Maksims Ivanovs – m.ivanovs@lu.lv
Uldis Bojārs – uldis.bojars@lu.lv

⌄ JavaScript

JavaScript scripting language - rest of the course
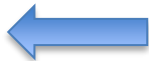
JavaScript language

Code Examples - Javascript

4th Practical Class - JavaScript

5th Practical Class JavaScript ⟵ This description and upload of the task solution

Working with DOM

Resources

- Slides (e-studijas):
  - JavaScript language
  - Working with DOM
  - Form validation

- W3Schools Array (check *filter* and *sort* array methods)- https://www.w3schools.com/jsref/jsref_obj_array.asp
- JavaScript (MDN) - https://developer.mozilla.org/en-US/docs/Web/JavaScript
- W3Schools - https://www.w3schools.com/js/

- Code examples
  - links in lecture slides

Tools

- Visual Studio Code / NetBeans / Notepad++
- Browser (Chrome, Firefox, …)

- Developer tools
  – Chrome: Ctrl+Shift+I
  – Firefox: F12

Task

- Create an interactive image gallery. The gallery should allow users to add images via a form, validate the input, and interact with the gallery by filtering images based on categories and sorting them by various criteria.

- Use the provided index.html and script.js files as a basis. Change them both if necessary.

- index.html already contains a form to add new images to the gallery with fields: Image URL, Image Name, Category

- script.js contains a mock-up of functions that can be implemented to solve the task. You can use the code mock-up or modify script.js as you wish.

Task cont.

- Form Validation:
  - Implement client-side form validation:
    - Ensure the Image URL points to an image file (e.g., ends with .jpg, .png, .gif, or .svg).
    - Ensure the Image Name is not empty.
    - Ensure the Category is selected.
  - Display error messages under each field if validation fails.
- Gallery Display:
  - Once the form is successfully submitted, the image should be added to a dynamic gallery below the form.
  - Each image in the gallery should display:
    - The image itself.
    - The image's name.
    - The category of the image.
    - The date and time when the image was added.
  - Each image should have a Remove button to delete the image from the gallery.

Task cont. II

- Filtering:
  - Add buttons above the gallery to filter images by their category (e.g., All, Nature, City, Abstract).
  - When a filter button is clicked, the gallery should only display images that belong to that category.
  - Clicking the All button should display all images regardless of category.
- Sorting: Add a dropdown to allow users to sort the images by:
  - Name (A-Z): Sort images alphabetically by their name in ascending order.
  - Name (Z-A): Sort images alphabetically by their name in descending order.
  - Date (Ascending): Sort images by the date they were added, from the earliest to the latest.
  - Date (Descending): Sort images by the date they were added, from the latest to the earliest.