Web Dev Bootcamp 2024/25

JavaScript - Lab 08

23.11.2024

Agenda

1. Q&A with Jan
2. Diskusi tentang tugas - tugas minggu lalu
3. JavaScript localStorage
4. Prep & To-Do untuk next session
5. Organisasi dan feedback

Q&A with Jan

- Lead industrial & structural engineer
- Last year led a project for building a microLED plant/factory in Malaysia
- Experienced in semiconductor manufacturing
- Background in civil engineering and software development

📍 Germany

Diskusi tentang tugas 2 minggu yang lalu

- **Build a Palindrome Checker** - Sari's solution

| Input Validation | ```function handleCheckPalindrome() { // Input validation
    const inputValue = userInput.value.trim();

    if (inputValue === '') {
        alert('Please input a value');
        return;
    }

    displayResult(inputValue, isPalindrome(inputValue));
    userInput.value = ''; // Clear input field
}``` |
|---|---|
| Conditions, Ternary operators | ```function displayResult(input, isPalindrome) {
    resultDiv.textContent = ''; // Clear previous result

    const message = document.createElement('p');
    message.className = 'user-input';
    message.textContent = `"${input}" ${isPalindrome ? 'is' : 'is not'} a palindrome.`;
    resultDiv.appendChild(message);

    resultDiv.classList.remove('hidden');
}``` |

Diskusi tentang tugas minggu lalu

- **Build a Roman Numeral Converter** - Sari's solution

Input Validations	Recursions
| ```javascript
function isValidNumber(inputStr, num) {
    if (!inputStr || inputStr.match(/[e.]/g)) {
        displayError('Please enter a valid number.');
        return false;
    }
    if (num < 1) {
        displayError('Please enter a number greater than or equal to 1.');
        return false;
    }
    if (num > 3999) {
        displayError('Please enter a number less than or equal to 3999.');
        return false;
    }
    return true;
}
``` | ```javascript
function convertToRoman(num) {
    const romanNumeralsMap = [
        ['M', 1000],
        ['CM', 900],
        ['D', 500],
        ['CD', 400],
        ['C', 100],
        ['XC', 90],
        ['L', 50],
        ['XL', 40],
        ['X', 10],
        ['IX', 9],
        ['V', 5],
        ['IV', 4],
        ['I', 1]
    ];
    let result = '';

    romanNumeralsMap.forEach(([roman, value]) => { // Looping the input number to the array above
        while (num >= value) {
            result += roman;
            num -= value;
        }
    });

    return result;
}
``` |

5

10 mins. break :)

JavaScript localStorage

- Kenapa tema MDN Web Docs aku di Chrome beda dengan di Edge?

- Walaupun aku gak sign-in, kenapa freeCodeCamp bisa simpan codes aku?

- Sewaktu aku mengisi formulir online, kadang ada feature 'Save for later'.

- Gimana caranya Google ingat recent searches?

- WhatsApp Web kok bisa ingat emoji-emoji yang barusan aku pake?

- Kok website - website nggak menanyakan cookie consent tiap aku refresh page nya?

Local Storage!

JavaScript localStorage

localStorage merupakan native API dari JavaScript

- Lumayan baru, release sekitar tahun 2018-an

- Semua modern browsers support localStorage

- Memiliki limit up to 5 mb

- Sifatnya persistent (data tidak terhapus walaupun browser ditutup atau PC di restart)

- Menyimpan data dalam format key-value objects

JavaScript localStorage

localStorage Features	Pros	Cons
Merupakan bagian dari `window` object property.	Bisa diakses oleh browser dari tab manapun.	Bisa menjadi celah untuk cross-site scripting (XSS) vulnerabilities.
Bisa menyimpan key-value strings.	Akses ke data yang lumayan cepat dan efisien.	Tidak direkomendasikan untuk menyimpan objects yang kompleks.
Data persistence spesifik untuk suatu website.	Caching.	Bisa menyebabkan data leakage.

9

JavaScript localStorage

```
localStorage.setItem("myFirstCat", "Nanuk");         // Di sini kita memasukkan data ke localStorage
localStorage.setItem("mySecondCat", "Balu");         // Data di localStorage selalu dalam bentuk key-value pairs
localStorage.setItem("myNeighborCat", "Mimi");       // Key-Value pairs ini selalu dalam tipe String (UTF-16)


const cat = localStorage.getItem("myFirstCat");
console.log(cat);                                    // Hasilnya adalah "Nanuk"

localStorage.removeItem("myNeighborCat");            // Menghapus key-value "myNeighborCat", "Mimi" dari localStorage

localStorage.clear();                                // Hapus semua data dari localStorage
```

More on localStorage:

- MDN Web Docs: localStorage
- freeCodeCamp: How to Use localStorage in JavaScript
- localStorage Complete Guide
- JavaScript localStorage In-depth Tutorial

- OWASP: HTML5 Security - Local Storage
- Save for Later Feature in Forms Using localStorage
- Storing and Retrieving JavaScript Objects in localStorage
- Please Stop Using localStorage

10

JavaScript localStorage

	localStorage	sessionStorage	Cookies
Ukuran	5 mb	5 mb	4 kb
Persistence	✔	✘	✘
Same-Origin rules	✔	✔	✔
Client-side only	✔	✘	✘
JavaScript API	✔	✔	✘
Akses	window	tab/page	tergantung Server
Legacy-Browsers	✘	✘	✔

To-Do untuk next session

5. When you click on the `#check-btn` element without entering a value into the `#user-input` element, an alert should appear with the text `"Please provide a phone number"`.

6. When you click on the `#clear-btn` element, the content within the `#results-div` element should be removed.

7. When the `#user-input` element contains `1 555-555-5555` and the `#check-btn` element is clicked, the `#results-div` element should contain the text `"Valid US number: 1 555-555-5555"`.

8. When the `#user-input` element contains `1 (555) 555-5555` and the `#check-btn` element is clicked, the `#results-div` element should contain the text `"Valid US number: 1 (555) 555-5555"`.

9. When the `#user-input` element contains `5555555555` and the `#check-btn` element is clicked, the `#results-div` element should contain the text `"Valid US number: 5555555555"`.

10. When the `#user-input` element contains `555-555-5555` and the `#check-btn` element is clicked, the `#results-div` element should contain the text `"Valid US number: 555-555-5555"`.

Source: https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures-v8/build-a-telephone-number-validator-project/build-a-telephone-number-validator

- **#1 s.d. #4**: HTML & CSS

- **#5**: Input validation

- **#6**: Event listener

- **#7 s.d. #34**: Conditions, membandingkan input String dengan regex patterns

12

To-Do untuk next session

09	**JavaScript Algo. & Data Structures**	● JavaScript localStorage	Sabtu, **23 Nov. 2024** 19:00 - 21:00 WIB	Pre-read: ● localStorage Complete Guide ● Using localStorage in JavaScript **Homework**: 1. **freeCodeCamp: Build a Telephone Number Validator**
10	**JavaScript Algo. & Data Structures**	● Sorting Algorithms ● Object-Oriented vs Functional Programming	Sabtu, **30 Nov. 2024** 19:00 - 21:00 WIB	**Pre-read**: ● Sorting Algorithms Explained ● What is Functional Programming? ● OOP in JavaScripts for Beginners ● (Optional) JavaScript OOP Tutorial **Homework**: 1. freeCodeCamp: Build a Cash Register

Organisasi dan feedback

1. Kita sekarang sudah separoh jalan di chapter JavaScript 🤗
   a. Siapa aja yang udah coba VMs dari Cloud Kilat?
   b. Gimana pace nya?
   c. Ada bagian/materi yang perlu diulangin lagi?
   d. Ada uneg-uneg?

2. Feedback: **https://forms.gle/zp8EYzFhATMasHGq6**