hogeschool vives

Labo 21 – Verslag JavaScript | Storage

Opleidingsonderdeel: **Web Development 1**

Bachelor in de: **Toegepaste informatica**

Afstudeerrichting: **Applicatieontwikkeling**

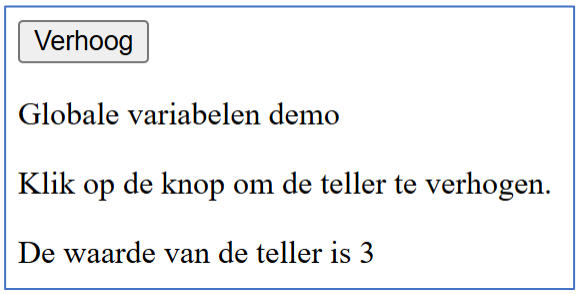
Specialisatie: **Cybersecurity & infrastructure**

Leerling: **Jari Opsomer**

Docent: **Pim Debaere**

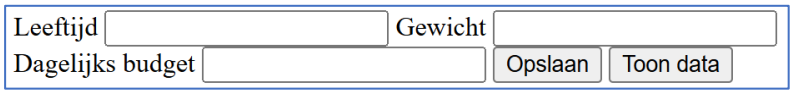
Academiejaar: **2025-2026**

2.1 Opdracht: demo storage

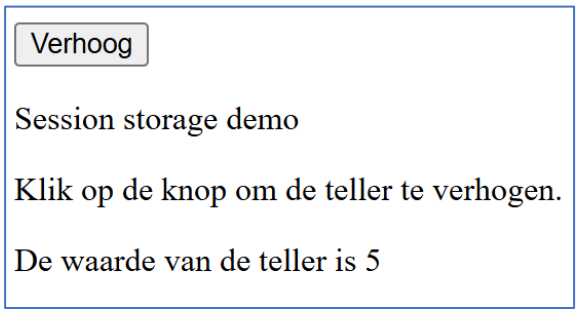


- Blijft de data behouden als je de pagina herlaadt? **Nee**
- Blijft de data behouden als je het tabblad sluit en de pagina opnieuw opent? **Nee**
- Gebruiken twee tabbladen die dezelfde pagina tonen, dezelfde data? **Nee**

3.2.1 Opdracht: demo storage

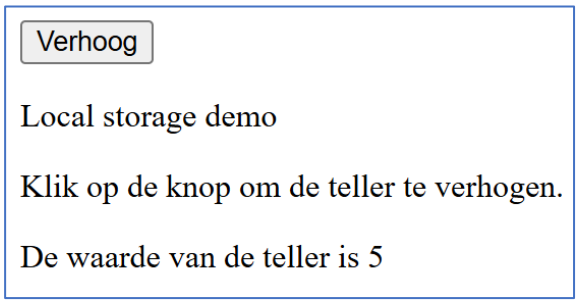


3.2.3 Opdracht: demo session storage



- Blijft de data behouden als je het tabblad sluit en de pagina opnieuw opent? **Nee**
- Blijft de data behouden als je de pagina herlaadt? **Ja**
- Gebruiken twee tabbladen die dezelfde pagina tonen, dezelfde data? **Nee**
- Denk je dat een andere pagina (andere file of url) aan de data geraakt als je ze in hetzelfde tabblad opent? **Nee**

3.2.5 Opdracht: demo local storage



- Blijft de data behouden als je het tabblad sluit en de pagina opnieuw opent? **Ja**
- Blijft de data behouden als je de pagina herlaadt? **Ja**
- Blijft de data behouden als je de browser afsluit, opnieuw start en de pagina weer opent? **Ja**
- Blijft de data behouden als je de pagina in een andere browser opent? **Nee**
- Wordt de data gedeeld door verschillende tabbladen als je de pagina meermaals opent? **Ja**
- Denk je dat een andere pagina (andere file of URL) aan de data geraakt? **Eerder niet (?)**

3.2.6 Opdracht: ColorPicker Pro

HTML-code:

```
<!DOCTYPE html>
<html lang="nl">
<head>
    <title>Color</title>
    <link rel="stylesheet" href="styles/styles.css">
</head>

<body>
    <div class="colorPicker">
       <div class="components">
          <div class="component">
             <span>0<input type="range" id="sldRed" class="slider"
value="128" min="0" max="255">255</span>
             <span>Red</span>
             <span id="lblRed">0</span>
          </div>
          <div class="component">
             <span>0<input type="range" id="sldGreen" class="slider"
value="255" min="0" max="255">255</span>
             <span>Green</span>
             <span id="lblGreen">0</span>
          </div>
          <div class="component">
             <span>0<input type="range" id="sldBlue" class="slider"
value="128" min="0" max="255">255</span>
             <span>Blue</span>
             <span id="lblBlue">0</span>
          </div>
       </div>
       <div class="swatch rounded" id="swatch"></div>
        <input type="button" id="btnSave" value="Save">
    </div>
    <div id="swatchComponents"></div>
    <script src="scripts/code.js"></script>
</body>
</html>
```

CSS-code:

```
.slider {
    vertical-align:middle;
}

.component span:nth-of-type(2) {
    display:inline-block;
    width:3em;
    text-align:right;
}

.components {
    display:inline-block;
    height:100%;
}
```

```css
.colorPicker {
    display:inline-block;
    border:solid 5px lightgrey;
    border-radius:10px;
    margin-bottom:20px;
}

/*

We splitsen het border-radius aspect af in een aparte CSS-klasse
zodat we het overgrote deel kunnen gebruiken voor zowel de swatch
in de colorpicker (met afgerond hoeken) en de nieuwe swatches die
we onderaan toevoegen. De swatch in de colorpicker zal dus twee
CSS-classes moeten krijgen, swatch en rounded

*/

.swatch {
    display:inline-block;
    border:solid 1px black;
    margin:0 0 0 1em;
    height:4em;
    width:5em;
}

.swatch.rounded {
    border-radius:5px;
}

.swatch>input {
    float:right;
}
```

JS-code:

```js
const STORAGE_KEY_SLIDERS = "colorpicker_sliders";
const STORAGE_KEY_SWATCHES = "colorpicker_swatches";

const initialize = () => {
    let btnSave = document.getElementById("btnSave");
    let sliders = document.getElementsByClassName("slider");
    for (let i = 0; i < sliders.length; i++) {
        sliders[i].addEventListener("change", update);
        sliders[i].addEventListener("input", update);
    }

    restoreSliders();
    restoreSwatches();
    update();

    btnSave.addEventListener("click", saveSwatch);
};
// Sliders opslaan & herstellen

const saveSliders = () => {
    const sliderValues = {
        red:   document.getElementById("sldRed").value,
        green: document.getElementById("sldGreen").value,
        blue:  document.getElementById("sldBlue").value
    };
    localStorage.setItem(STORAGE_KEY_SLIDERS, JSON.stringify(sliderValues));
};
```

```
const restoreSliders = () => {
    const stored = localStorage.getItem(STORAGE_KEY_SLIDERS);
    if (!stored) return;
    const { red, green, blue } = JSON.parse(stored);
    document.getElementById("sldRed").value   = red;
    document.getElementById("sldGreen").value = green;
    document.getElementById("sldBlue").value  = blue;
};

// Swatches opslaan & herstellen

const loadSwatchData = () => {
    const stored = localStorage.getItem(STORAGE_KEY_SWATCHES);
    return stored ? JSON.parse(stored) : [];
};

const persistSwatches = () => {
    // Lees de data-attributen van alle bestaande swatches uit de DOM
    const swatchDivs = document.getElementById("swatchComponents").children;
    const data = [];
    for (let div of swatchDivs) {
        data.push({
            red:   div.getAttribute("data-red"),
            green: div.getAttribute("data-green"),
            blue:  div.getAttribute("data-blue")
        });
    }
    localStorage.setItem(STORAGE_KEY_SWATCHES, JSON.stringify(data));
};

const restoreSwatches = () => {
    const swatchData = loadSwatchData();
    const swatchComponents = document.getElementById("swatchComponents");
    swatchData.forEach(({ red, green, blue }) => {
        const swatch = buildSwatchComponent(red, green, blue);
        swatchComponents.appendChild(swatch);
    });
};

// Swatch aanmaken & beheren

const saveSwatch = () => {
    const swatchComponents = document.getElementById("swatchComponents");
    const swatch = buildSwatchComponent(
        document.getElementById("sldRed").value,
        document.getElementById("sldGreen").value,
        document.getElementById("sldBlue").value
    );
    swatchComponents.appendChild(swatch);
    persistSwatches(); // opslaan na toevoeging
};

const configureSwatch = (swatch, red, green, blue) => {
    swatch.setAttribute("data-red",   red);
    swatch.setAttribute("data-green", green);
    swatch.setAttribute("data-blue",  blue);
    swatch.style.background = `rgb(${red},${green},${blue})`;
};

const buildSwatchComponent = (red, green, blue) => {
    let swatch    = document.createElement("div");
    let btnDelete = document.createElement("input");

    swatch.className = "swatch";
    configureSwatch(swatch, red, green, blue);
    swatch.addEventListener("click", setColorPickerFromSwatch);

    btnDelete.setAttribute("type",  "button");
```

```
    btnDelete.setAttribute("value", "X");
    btnDelete.addEventListener("click", deleteSwatch);

    swatch.appendChild(btnDelete);
    return swatch;
};

const setColorPickerFromSwatch = (event) => {
    let swatch = event.target;
    document.getElementById("sldRed").value   = swatch.getAttribute("data-red");
    document.getElementById("sldGreen").value = swatch.getAttribute("data-green");
    document.getElementById("sldBlue").value  = swatch.getAttribute("data-blue");
    update();
};

const deleteSwatch = (event) => {
    let swatchComponents = document.getElementById("swatchComponents");
    let swatch = event.target.parentNode;
    swatchComponents.removeChild(swatch);
    persistSwatches(); // opslaan na verwijdering
    event.stopPropagation();
};

// UI bijwerken

const update = () => {
    let red   = document.getElementById("sldRed").value;
    let green = document.getElementById("sldGreen").value;
    let blue  = document.getElementById("sldBlue").value;

    document.getElementById("lblRed").innerHTML   = red;
    document.getElementById("lblGreen").innerHTML = green;
    document.getElementById("lblBlue").innerHTML  = blue;

    document.getElementById("swatch").style.background =
`rgb(${red},${green},${blue})`;

    saveSliders(); // slider stand opslaan bij elke wijziging
};

window.addEventListener("load", initialize);
```