

Labo 20 – Verslag JavaScript | Deel 8

Opleidingsonderdeel: **Web Development 1**

Bachelor in de: **Toegepaste informatica**

Afstudeerrichting: **Applicatieontwikkeling**

Specialisatie: **Cybersecurity & infrastructure**

Leerling: **Jari Opsomer**

Docent: **Pim Debaere**

Academiejaar: **2025-2026**

2.2 Opdracht: dagen op de werelddbol

HTML-code:

```
<!DOCTYPE html>
<html lang="nl">
<head>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles/style.css">
  <title>Title</title>
</head>
<body>

<p></p>

<script type="text/javascript" charset="utf-8"
src="scripts/code.js"></script>
</body>
</html>
```

JS-code:

```
const setup = () => {
  const geboortedatum = new Date("2007-05-04T00:00:00.000Z");
  let verschil = new Date() - geboortedatum;
  let verschilInDagen = verschil / 1000 / 60 / 60 / 24;
  let string = `${Math.floor(verschilInDagen)} dagen (${Math.floor(verschilInDagen / 364)} jaar)`;
  console.log(string);
  document.querySelector("p").textContent = string;
}

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

Resultaat:

6908 dagen (18 jaar)

2.6 Opdracht: JSON serialisatie en deserialisatie

JS-code:

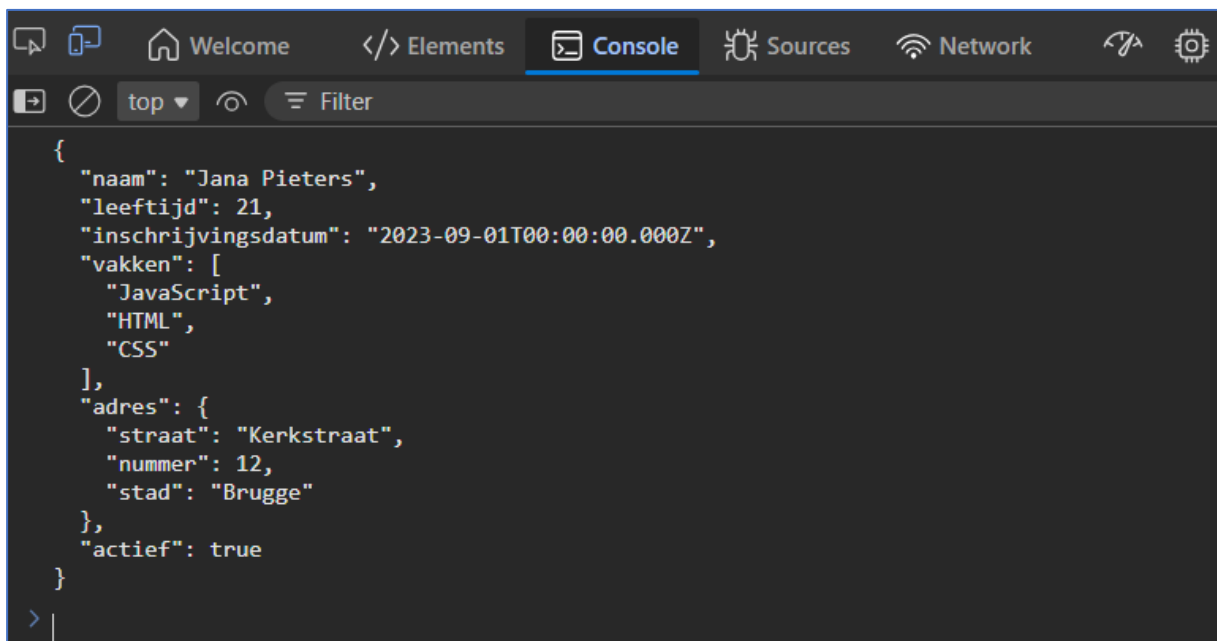
```
// Programma 1: object definiëren en omzetten naar JSON

const adres = { straat: "Kerkstraat", nummer: 12, stad: "Brugge" };

const student1 = {
  naam: "Jana Pieters",
  leeftijd: 21,
  inschrijvingsdatum: new Date("2023-09-01"),
  vakken: ["JavaScript", "HTML", "CSS"],
  adres: adres,
  actief: true
};

const jsonString = JSON.stringify(student1, null, 2);
console.log(jsonString);
```

Console:



```
{
  "naam": "Jana Pieters",
  "leeftijd": 21,
  "inschrijvingsdatum": "2023-09-01T00:00:00.000Z",
  "vakken": [
    "JavaScript",
    "HTML",
    "CSS"
  ],
  "adres": {
    "straat": "Kerkstraat",
    "nummer": 12,
    "stad": "Brugge"
  },
  "actief": true
}
```

JS-code:

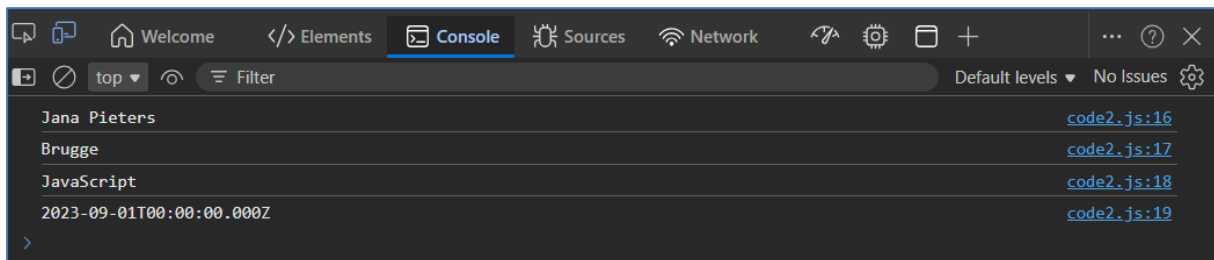
```
// Programma 2: JSON string terug omzetten naar object

const jsonString = `{
  "naam": "Jana Pieters",
  "leeftijd": 21,
  "inschrijvingsdatum": "2023-09-01T00:00:00.000Z",
  "vakken": ["JavaScript", "HTML", "CSS"],
  "adres": { "straat": "Kerkstraat", "nummer": 12, "stad": "Brugge" },
  "actief": true
}`;

const student2 = JSON.parse(jsonString);

console.log(student2.naam); // "Jana Pieters"
console.log(student2.adres.stad); // "Brugge"
console.log(student2.vakken[0]); // "JavaScript"
console.log(student2.inschrijvingsdatum); // "2023-09-01T00:00:00.000Z" ←
string, geen Date!
```

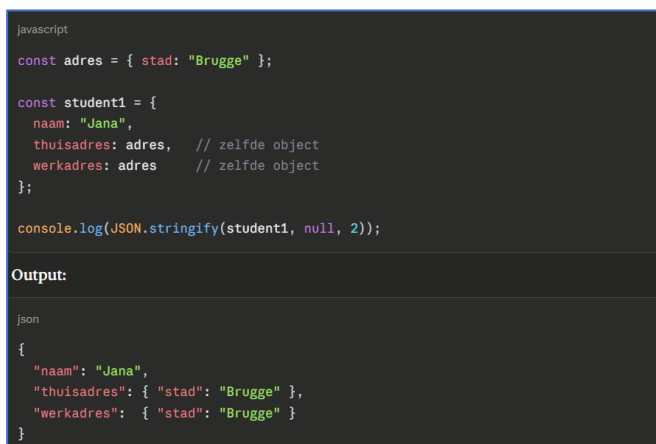
Console:



Vraag 1 – Welk formaat gebruikt JSON voor een `Date` object?

`JSON.stringify()` converteert een `Date` object automatisch naar een **ISO 8601 string**: "2023-09-01T00:00:00.000Z"

Vraag 2 – Wat als hetzelfde object meerdere keren voorkomt?



2.7 Opdracht: contactmanager

Data:

```
let personen=[
  {
    voornaam: 'Jan',
    familienaaam: 'Janssens',
    geboorteDatum: new Date('2010-10-10'),
    email: 'jan@example.com',
    aantalKinderen: 0
  },
  {
    voornaam: 'Mieke',
    familienaaam: 'Mickelsen',
    geboorteDatum: new Date('1980-01-01'),
    email: 'mieke@example.com',
    aantalKinderen: 1
  },
  {
    voornaam: 'Piet',
    familienaaam: 'Pieters',
    geboorteDatum: new Date('1970-12-31'),
    email: 'piet@example.com',
    aantalKinderen: 2
  }
];
```

HTML-code:

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

<body>
  <div>
    <select id="lstPersonen" size="10">
    </select>
    <input type="button" id="btnNieuw" value="Nieuw">
  </div>
  <form>
    <div class="line">
      <label for="txtVoornaam">Voornaam</label>
      <input type="text" id="txtVoornaam" value="">
      <span class="errorMessage" id="errVoornaam"></span>
    </div>
    <div class="line">
      <label for="txtFamilienaam">Familienaam</label>
      <input type="text" id="txtFamilienaam" value="">
      <span class="errorMessage" id="errFamilienaam"></span>
    </div>
    <div class="line">
      <label for="txtGeboorteDatum">Geboortedatum</label>
      <input type="text" id="txtGeboorteDatum" value="">
      <span class="errorMessage" id="errGeboorteDatum"></span>
    </div>
  </form>
</body>
```

```

    </div>
    <div class="line">
      <label for="txtEmail">Email</label>
      <input type="text" id="txtEmail" value="">
      <span class="errorMessage" id="errEmail"></span>
    </div>
    <div class="line">
      <label for="txtAantalKinderen">Aantal kinderen</label>
      <input type="text" id="txtAantalKinderen" value="">
      <span class="errorMessage" id="errAantalKinderen"></span>
    </div>
    <input type="button" id="btnBewaar" value="Bewaar">
  </form>

  <script src="scripts/validatie.js"></script>
  <script src="scripts/code.js"></script>
</body>
</html>

```

CSS-code:

```

.errorMessage {
  color: red;
}

input.invalid {
  border: solid 1px red;
}

form label {
  display: inline-block;
  width: 6em;
}

#lstPersonen {
  width: 400px;
}

body {
  font-family: Arial, sans-serif;
  padding: 1em;
}

div {
  margin-bottom: 0.5em;
}

.line {
  margin-bottom: 0.4em;
}

input[type="button"] {
  margin-top: 0.5em;
  margin-right: 0.5em;
  padding: 0.3em 0.8em;
  cursor: pointer;
}

select {
  display: block;
  margin-bottom: 0.5em;
}

```

JS-code | validatie.js:

```
// Dit is dezelfde valideringscode als in de form opgave.
// Verwijderde functionaliteit :
// - load event listener
// - proficiat popup

const valideer = () => {
  valideerVoornaam();
  valideerFamilienaam();
  valideerGeboorteDatum();
  valideerEmail();
  valideerAantalKinderen();
};

const valideerVoornaam = () => {
  let txtVoornaam = document.getElementById("txtVoornaam");
  let voornaam = txtVoornaam.value.trim();
  if (voornaam.length > 30) {
    reportError(txtVoornaam, "max. 30 karakters");
  } else {
    clearError(txtVoornaam);
  }
};

const valideerFamilienaam = () => {
  let txtFamilienaam = document.getElementById("txtFamilienaam");
  let familienaam = txtFamilienaam.value.trim();
  if (familienaam.length == 0) {
    reportError(txtFamilienaam, "verplicht veld");
  } else if (familienaam.length > 50) {
    reportError(txtFamilienaam, "max. 50 karakters");
  } else {
    clearError(txtFamilienaam);
  }
};

const valideerGeboorteDatum = () => {
  let txtGeboorteDatum = document.getElementById("txtGeboorteDatum");
  let geboorteDatum = txtGeboorteDatum.value.trim();
  // correcte lengte?
  if (geboorteDatum.length != 10) {
    reportError(txtGeboorteDatum, "verplicht veld");
  } else {
    let formatCorrect=true;
    // streepjes op juiste plaats?
    if (formatCorrect && !(geboorteDatum.charAt(4)=='-' &&
geboorteDatum.charAt(7)=='-') ) {
      formatCorrect=false;
    }
    // jaar gedeelte een getal?
    if (formatCorrect) {
      // year
      let yearText=geboorteDatum.substring(0,4);
      if (!isPositiveNonZeroNumber(yearText)) {
        formatCorrect=false;
      }
    }
    // maand gedeelte een getal?
    if (formatCorrect) {
      // month
      let monthText=geboorteDatum.substring(5,7);
      if (!isPositiveNonZeroNumber(monthText)) {
        formatCorrect=false;
      }
    }
    // dag gedeelte een getal?
    if (formatCorrect) {
```

```

        // day
        let dayText=geboorteDatum.substring(8,11);
        if (!isPositiveNonZeroNumber(dayText)) {
            formatCorrect=false;
        }
    }

    if (formatCorrect) {
        clearError(txtGeboorteDatum);
    } else {
        reportError(txtGeboorteDatum, "formaat is niet jjjj-mm-dd");
    }
}
};

const valideerEmail = () => {
    let txtEmail = document.getElementById("txtEmail");
    let email = txtEmail.value.trim();
    if (email.length==0) {
        reportError(txtEmail, "verplicht veld");
    } else {
        let formatCorrect=true;
        let idx=email.indexOf("@");

        if (idx<1 || idx==email.length-1) {
            // @ teken komt niet voor, of helemaal vooraan of helemaal achteraan de
            tekst
            formatCorrect=false;
        }
        idx=email.indexOf("@", idx+1);
        if (formatCorrect && idx!=-1) {
            // @-teken komt meermaals voor
            formatCorrect=false;
        }
        if (formatCorrect) {
            clearError(txtEmail);
        } else {
            reportError(txtEmail, "geen geldig email adres");
        }
    }
};

const valideerAantalKinderen = () => {
    let txtAantalKinderen = document.getElementById("txtAantalKinderen");
    let aantalKinderenText = txtAantalKinderen.value.trim();
    if (aantalKinderenText.length==0) {
        reportError(txtAantalKinderen, "verplicht veld");
    } else if (!isPositiveNumber(aantalKinderenText)) {
        reportError(txtAantalKinderen, "is geen positief getal");
    } else {
        let aantal=parseInt(aantalKinderenText);
        if (aantal>=99) {
            reportError(txtAantalKinderen, "te vruchtbaar");
        } else {
            clearError(txtAantalKinderen);
        }
    }
};

const isPositiveNumber = (text) => {
    let number=parseInt(text, 10);
    return !isNaN(number) && number>=0;
    // merk op dat een tekst als "34m" ook aanvaard wordt, wat eigenlijk niet de
    bedoeling is
    // een betere oplossing zou bv. een reguliere expressie gebruiken
};

const isPositiveNonZeroNumber = (text) => {

```

```

    let number=parseInt(text, 10);
    return !isNaN(number) && number>0;
    // merk op dat een tekst als "34m" ook aanvaard wordt, wat eigenlijk niet de
    bedoeling is
    // een betere oplossing zou bv. een reguliere expressie gebruiken
};

const reportError = (element, message) => {
    let elementId=element.getAttribute("id"); // bv. txtVoornaam
    let errElementId="err"+elementId.substring(3, elementId.length); // bv.
errVoornaam
    let errElement=document.getElementById(errElementId);
    element.className="invalid";
    errElement.innerHTML = message;
};

const clearError = (element) => {
    let elementId=element.getAttribute("id"); // bv. txtVoornaam
    let errElementId="err"+elementId.substring(3, elementId.length); // bv.
errVoornaam
    let errElement=document.getElementById(errElementId);
    element.className="";
    errElement.innerHTML = "";
};

const clearAllErrors = () => {
    let fieldIds=["txtVoornaam", "txtFamilienaam", "txtGeboorteDatum", "txtEmail",
"txtAantalKinderen"];
    for (let i=0;i<fieldIds.length;i++) {
        clearError( document.getElementById(fieldIds[i]) );
    }
}

```

JS-code | code.js:

```

// Voorgedefinieerde personen
let personen = [
    {
        voornaam: 'Jan',
        familienaam: 'Janssens',
        geboorteDatum: new Date('2010-10-10'),
        email: 'jan@example.com',
        aantalKinderen: 0
    },
    {
        voornaam: 'Mieke',
        familienaam: 'Mickelsen',
        geboorteDatum: new Date('1980-01-01'),
        email: 'mieke@example.com',
        aantalKinderen: 1
    },
    {
        voornaam: 'Piet',
        familienaam: 'Pieters',
        geboorteDatum: new Date('1970-12-31'),
        email: 'piet@example.com',
        aantalKinderen: 2
    }
];

// Houdt bij welke persoon momenteel getoond wordt in het formulier.
// -1 betekent: nieuwe (nog niet bestaande) persoon
let huidigIndex = -1;

// Hulpfunctie: formatteer een Date-object als "jjjj-mm-dd"
const formateerDatum = (datum) => {
    let jaar = datum.getFullYear();

```

```

    let maand = String(datum.getMonth() + 1).padStart(2, '0');
    let dag = String(datum.getDate()).padStart(2, '0');
    return `${jaar}-${maand}-${dag}`;
};

// Hulpfunctie: maak een <option> element aan en voeg het toe aan de lijst
const voegOptieToe = (persoon, index) => {
    let lstPersonen = document.getElementById("lstPersonen");
    let optie = document.createElement("option");
    optie.id = index; // id = index in de personen array
    optie.value = index;
    optie.textContent = `${persoon.familienaam} ${persoon.voornaam}`;
    lstPersonen.appendChild(optie);
};

// Toon de gegevens van een persoon in het formulier
const toonPersoon = (persoon) => {
    document.getElementById("txtVoornaam").value = persoon.voornaam;
    document.getElementById("txtFamilienaam").value = persoon.familienaam;
    document.getElementById("txtGeboorteDatum").value =
formateerDatum(persoon.geboorteDatum);
    document.getElementById("txtEmail").value = persoon.email;
    document.getElementById("txtAantalKinderen").value = persoon.aantalKinderen;
    clearAllErrors();
};

// Maak het formulier leeg
const leegFormulier = () => {
    document.getElementById("txtVoornaam").value = "";
    document.getElementById("txtFamilienaam").value = "";
    document.getElementById("txtGeboorteDatum").value = "";
    document.getElementById("txtEmail").value = "";
    document.getElementById("txtAantalKinderen").value = "";
    clearAllErrors();
};

// Controleer of het formulier geen validatiefouten bevat
const isFormulierGeldig = () => {
    let foutSpans = document.querySelectorAll(".errorMessage");
    for (let span of foutSpans) {
        if (span.innerHTML.trim() !== "") {
            return false;
        }
    }
    return true;
};

// Lees de ingevoerde gegevens uit het formulier
const leesFormulierData = () => {
    return {
        voornaam: document.getElementById("txtVoornaam").value.trim(),
        familienaam: document.getElementById("txtFamilienaam").value.trim(),
        geboorteDatum: new
Date(document.getElementById("txtGeboorteDatum").value.trim()),
        email: document.getElementById("txtEmail").value.trim(),
        aantalKinderen:
parseInt(document.getElementById("txtAantalKinderen").value.trim())
    };
};

// Update de tekst van een bestaande optie in de lijst
const updateOptie = (index, persoon) => {
    let optie = document.getElementById(String(index));
    if (optie) {
        optie.textContent = `${persoon.familienaam} ${persoon.voornaam}`;
    }
};

```

```

// Event listener: klik op "Bewaar"
const bewaarBewerktePersoon = () => {
  // Valideer alle velden
  valideer();

  // Ga enkel verder als het formulier geldig is
  if (!isFormulierGeldig()) {
    return;
  }

  let data = leesFormulierData();

  if (huidigIndex === -1) {
    // Nieuwe persoon: object aanmaken en toevoegen aan array en lijst
    let nieuwePersoon = {
      voornaam: data.voornaam,
      familienaam: data.familienaam,
      geboorteDatum: data.geboorteDatum,
      email: data.email,
      aantalKinderen: data.aantalKinderen
    };
    personen.push(nieuwePersoon);
    let nieuweIndex = personen.length - 1;
    voegOptieToe(nieuwePersoon, nieuweIndex);

    // Stel de nieuwe index in en selecteer de persoon in de lijst
    huidigIndex = nieuweIndex;
    document.getElementById("lstPersonen").value = huidigIndex;
  } else {
    // Bestaande persoon: eigenschappen aanpassen
    let persoon = personen[huidigIndex];
    persoon.voornaam = data.voornaam;
    persoon.familienaam = data.familienaam;
    persoon.geboorteDatum = data.geboorteDatum;
    persoon.email = data.email;
    persoon.aantalKinderen = data.aantalKinderen;

    // Naam in de lijst bijwerken
    updateOptie(huidigIndex, persoon);
  }
};

// Event listener: klik op "Nieuw"
const bewerkNieuwePersoon = () => {
  huidigIndex = -1;
  leegFormulier();

  // Deselecteer eventuele selectie in de lijst
  document.getElementById("lstPersonen").value = "";
};

// Event listener: selectie verandert in de lijst
const persoonGeselecteerd = (event) => {
  huidigIndex = parseInt(event.target.value);
  toonPersoon(personen[huidigIndex]);
};

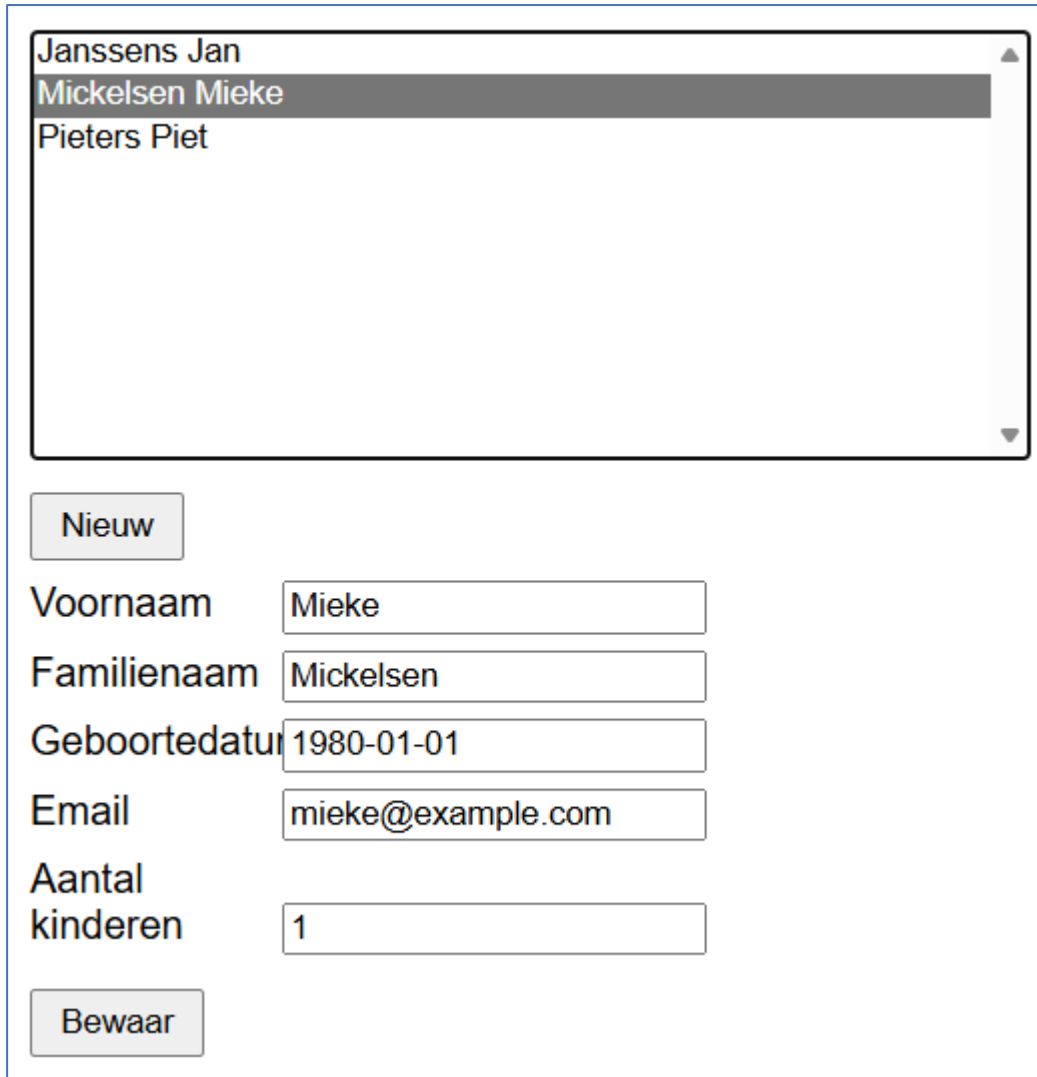
// Setup: event listeners registreren en lijst vullen met voorgedefinieerde data
const setup = () => {
  // Vul de lijst met de voorgedefinieerde personen
  for (let i = 0; i < personen.length; i++) {
    voegOptieToe(personen[i], i);
  }

  // Registreer event listeners
  document.getElementById("btnBewaar").addEventListener("click",
bewaarBewerktePersoon);
  document.getElementById("btnNieuw").addEventListener("click",

```

```
bewerkNieuwePersoon);  
    document.getElementById("lstPersonen").addEventListener("change",  
    persoonGeselecteerd);  
};  
  
window.addEventListener("load", setup);
```

Resultaat:



The screenshot shows a web form with a list of names and input fields. The list contains three names: Janssens Jan, Mickelsen Mieke (highlighted), and Pieters Piet. Below the list is a 'Nieuw' button. The form fields are: Voornaam (Mieke), Familiennaam (Mickelsen), Geboortedatum (1980-01-01), Email (mieke@example.com), and Aantal kinderen (1). A 'Bewaar' button is at the bottom.

<input type="button" value="Nieuw"/>	
Voornaam	<input type="text" value="Mieke"/>
Familiennaam	<input type="text" value="Mickelsen"/>
Geboortedatum	<input type="text" value="1980-01-01"/>
Email	<input type="text" value="mieke@example.com"/>
Aantal kinderen	<input type="text" value="1"/>
<input type="button" value="Bewaar"/>	