Oefening M8 — Focus & Field Monitor (DOM-properties + acties)

In deze oefening bouw je een klein hulpmiddel dat de gebruiker begeleidt tijdens het invullen van een mini-formulier. Het doel is om verschillende **DOM-eigenschappen** en **acties** te oefenen: je gaat lezen uit invoervelden, de focus en blur van elementen gebruiken, tellen hoeveel velden al ingevuld zijn en die informatie tonen in een meldingsvak.

De gebruiker krijgt drie invoervelden te zien: een titel, een korte beschrijving en een e-mailadres. Terwijl de gebruiker het formulier gebruikt, gebeurt er een aantal dingen automatisch:

1. **Focus-highlights rondom de velden**

   Wanneer een invoerveld de focus krijgt (de gebruiker klikt erin of tabt ernaartoe), wordt het volledige invoervak visueel in de kijker gezet. Je doet dit door de omliggende container een extra stijl te geven zodat de gebruiker meteen ziet waar hij bezig is. Verlaat de gebruiker het veld (blur), dan verdwijnt deze highlight weer, behalve als er een foutmelding is voor dat veld.

2. **Live teller van ingevulde velden**

   Onder de invoervelden staat een meldingsvak dat toont hoeveel velden momenteel geldig ingevuld zijn. Je werkt met drie toestanden:

   - **Nog niets of weinig ingevuld** Als er geen enkel veld geldig is, krijgt het meldingsvak een neutrale, grijze stijl en toont het de tekst `Nog geen velden ingevuld.`
   - **Deels ingevuld** Zodra er minstens één veld geldig is, maar nog niet alle drie, toont het vak de tekst `X van de 3 velden zijn ingevuld.` (waarbij X vervangen wordt door het echte aantal) en krijgt het een informatieve, blauwe stijl.
   - **Alle velden ingevuld** Wanneer alle drie de velden geldig zijn, wordt de stijl groen en verschijnt de tekst: `Alle velden zijn ingevuld` 🎉.

   Je bepaalt "geldig" voor elk veld met eenvoudige regels:

   - Titel: niet leeg.
   - Beschrijving: minstens 10 tekens.
   - E-mail: bevat minstens één `@` en één `.`.

3. **Inline feedback per veld**

   Elk invoerveld heeft onderaan een klein tekstje dat de status van dat veld toont.

   - Bij een lege of ongeldige waarde toon je een korte waarschuwing in een subtiele rode stijl, bijvoorbeeld `Titel is verplicht` of `E-mailadres lijkt ongeldig`.
   - Bij een geldige waarde krijgt de tekst een groene, bevestigende toon, bijvoorbeeld `OK` of `E-mailadres ziet er goed uit`.

   Deze feedback wordt telkens bijgewerkt wanneer de gebruiker het veld verlaat (blur) of tijdens het typen (input of keyup), zodat hij geen submit-knop nodig heeft om fouten te zien.

4. **Actieknop die reageert op de status**

   Onderaan staat een knop “Focus op lege velden”. Wanneer niet alle velden geldig zijn, kan de gebruiker hierop klikken. Dan:

   - Zoek je het eerste veld dat nog niet geldig is.
   - Zet je de focus expliciet naar dat veld met de juiste DOM-actie.
   - Het meldingsvak krijgt tijdelijk een gele waarschuwingstekst zoals `Je hebt nog onvolledige velden. We hebben je naar het eerste lege veld gebracht.`

   Als alle velden al geldig zijn en de gebruiker toch op de knop klikt, dan verandert het meldingsvak naar een groene stijl met de boodschap `Alles is in orde, geen lege velden meer.`

5. **Overzicht van ingevulde waarden (samenvatting)**

   Helemaal onderaan tonen we een kleine samenvatting van de waarden zodra álle velden geldig zijn. Dit is bijvoorbeeld een lijst met drie lijnen:

   - `Titel: ...`
   - `Beschrijving: ...`
   - `E-mail: ...`

   Zolang niet alle velden geldig zijn, toon je hier een neutrale tekst zoals `Samenvatting verschijnt zodra alle velden geldig zijn.`

Samengevat: je combineert in deze oefening **DOM-properties** (zoals de waarde van inputs lezen), **DOM-acties** (focus zetten) en een klein beetje **stijlverandering** via klassen om de gebruiker voortdurend duidelijke feedback te geven.

Section (HTML)

```
<section class="container py-4" id="ex-m8">
  <div class="row justify-content-center">
    <div class="col-md-8 col-lg-6">
      <div class="card shadow-sm">
        <div class="card-header bg-primary text-white d-flex justify-content-
between align-items-center">
          <span>Oefening M8 – Focus &amp;amp; Field Monitor</span>
          <span class="badge bg-light text-primary">DOM properties &amp;amp;
acties</span>
        </div>
        <div class="card-body">

          <!-- Invoerblokken -->
          <div id="m8_group_title" class="mb-3 p-3 border rounded-3 bg-
light">
            <label for="m8_title" class="form-label mb-1">Titel</label>
            <input
              id="m8_title"
              class="form-control"
              placeholder="bv. Mijn eerste JavaScript-project"
            />
            <div id="m8_title_help" class="form-text mt-1 text-muted">
              Titel is verplicht.
            </div>
          </div>
```

```
          <div id="m8_group_desc" class="mb-3 p-3 border rounded-3 bg-light">
            <label for="m8_desc" class="form-label mb-1">Korte
beschrijving</label>
            <textarea
              id="m8_desc"
              class="form-control"
              rows="3"
              placeholder="Beschrijf in minstens 10 tekens wat je project
doet."
            ></textarea>
            <div id="m8_desc_help" class="form-text mt-1 text-muted">
              Minstens 10 tekens.
            </div>
          </div>

          <div id="m8_group_email" class="mb-3 p-3 border rounded-3 bg-
light">
            <label for="m8_email" class="form-label mb-1">E-mail</label>
            <input
              id="m8_email"
              type="email"
              class="form-control"
              placeholder="bv. jij@voorbeeld.be"
            />
            <div id="m8_email_help" class="form-text mt-1 text-muted">
              E-mailadres bevat minstens een @ en een punt.
            </div>
          </div>

          <!-- Actieknop -->
          <button id="m8_focus_btn" class="btn btn-primary w-100 mb-3">
            Focus op lege velden
          </button>

          <!-- Status van aantal ingevulde velden -->
          <div id="m8_status" class="alert alert-secondary mb-3"
role="status" aria-live="polite">
            Nog geen velden ingevuld.
          </div>

          <!-- Samenvatting -->
          <div class="card bg-light border-0">
            <div class="card-header bg-white border-0 pb-1">
              <h6 class="card-title mb-0">Samenvatting</h6>
            </div>
            <div class="card-body pt-2">
              <ul id="m8_summary" class="list-unstyled mb-0">
                <li class="text-muted">Samenvatting verschijnt zodra alle
velden geldig zijn.</li>
              </ul>
            </div>
          </div>

        </div>
      </div>
```

```
    </div>
  </div>
</section>
```

JavaScript-oplossing

```
// ------------------------------------
// Oefening M8 – Focus & Field Monitor
// ------------------------------------

function m8_isTitleValid() {
  const value = document.getElementById('m8_title').value.trim();
  return value.length > 0;
}

function m8_isDescValid() {
  const value = document.getElementById('m8_desc').value.trim();
  return value.length >= 10;
}

function m8_isEmailValid() {
  const value = document.getElementById('m8_email').value.trim();
  return value.includes('@') && value.includes('.');
}

function m8_updateFieldFeedback() {
  const title = document.getElementById('m8_title');
  const desc = document.getElementById('m8_desc');
  const email = document.getElementById('m8_email');

  const titleHelp = document.getElementById('m8_title_help');
  const descHelp = document.getElementById('m8_desc_help');
  const emailHelp = document.getElementById('m8_email_help');

  // Titel
  if (m8_isTitleValid()) {
    titleHelp.className = 'form-text mt-1 text-success';
    titleHelp.textContent = 'OK.';
  } else {
    titleHelp.className = 'form-text mt-1 text-danger';
    titleHelp.textContent = 'Titel is verplicht.';
  }

  // Beschrijving
  if (m8_isDescValid()) {
    descHelp.className = 'form-text mt-1 text-success';
    descHelp.textContent = 'Beschrijving is lang genoeg.';
  } else {
    const len = desc.value.trim().length;
    descHelp.className = 'form-text mt-1 text-danger';
    descHelp.textContent = `Te kort (${len}/10).`;
  }

  // E-mail
  if (m8_isEmailValid()) {
    emailHelp.className = 'form-text mt-1 text-success';
```

```
    emailHelp.textContent = 'E-mailadres lijkt geldig.';
  } else {
    emailHelp.className = 'form-text mt-1 text-danger';
    emailHelp.textContent = 'E-mailadres lijkt ongeldig.';
  }
}

function m8_countValidFields() {
  let count = 0;
  if (m8_isTitleValid()) count++;
  if (m8_isDescValid()) count++;
  if (m8_isEmailValid()) count++;
  return count;
}

function m8_updateStatus() {
  const status = document.getElementById('m8_status');
  const count = m8_countValidFields();

  if (count === 0) {
    status.className = 'alert alert-secondary mb-3';
    status.textContent = 'Nog geen velden ingevuld.';
  } else if (count < 3) {
    status.className = 'alert alert-info mb-3';
    status.textContent = `${count} van de 3 velden zijn ingevuld.`;
  } else {
    status.className = 'alert alert-success mb-3';
    status.textContent = 'Alle velden zijn ingevuld 🎉';
  }
}

function m8_updateSummary() {
  const summary = document.getElementById('m8_summary');
  const allValid = m8_countValidFields() === 3;

  if (!allValid) {
    summary.innerHTML =
      '<li class="text-muted">Samenvatting verschijnt zodra alle velden
geldig zijn.</li>';
    return;
  }

  const titleVal = document.getElementById('m8_title').value.trim();
  const descVal = document.getElementById('m8_desc').value.trim();
  const emailVal = document.getElementById('m8_email').value.trim();

  summary.innerHTML = `
    <li><strong>Titel:</strong> ${titleVal}</li>
    <li><strong>Beschrijving:</strong> ${descVal}</li>
    <li><strong>E-mail:</strong> ${emailVal}</li>
  `;
}

function m8_onFieldChange() {
  m8_updateFieldFeedback();
```

```
  m8_updateStatus();
  m8_updateSummary();
}

// Focus-highlights
function m8_addFocusHighlight(groupId) {
  const group = document.getElementById(groupId);
  group.classList.add('border-primary', 'border-2', 'bg-white');
}

function m8_removeFocusHighlight(groupId) {
  const group = document.getElementById(groupId);
  group.classList.remove('border-primary', 'border-2', 'bg-white');
}

function m8_focusFirstInvalidField() {
  const status = document.getElementById('m8_status');

  if (m8_countValidFields() === 3) {
    status.className = 'alert alert-success mb-3';
    status.textContent = 'Alles is in orde, geen lege velden meer.';
    return;
  }

  status.className = 'alert alert-warning mb-3';
  status.textContent =
    'Je hebt nog onvolledige velden. We hebben je naar het eerste lege veld
gebracht.';

  // Zoek eerste ongeldige veld in vaste volgorde
  if (!m8_isTitleValid()) {
    document.getElementById('m8_title').focus();
    return;
  }
  if (!m8_isDescValid()) {
    document.getElementById('m8_desc').focus();
    return;
  }
  if (!m8_isEmailValid()) {
    document.getElementById('m8_email').focus();
  }
}

document.addEventListener('DOMContentLoaded', () => {
  const title = document.getElementById('m8_title');
  const desc = document.getElementById('m8_desc');
  const email = document.getElementById('m8_email');

  // Input / wijziging events
  title?.addEventListener('input', m8_onFieldChange);
  desc?.addEventListener('input', m8_onFieldChange);
  email?.addEventListener('input', m8_onFieldChange);

  // Focus & blur events voor highlight rond de groepen
  title?.addEventListener('focus', () =>
```

```
m8_addFocusHighlight('m8_group_title'));
  title?.addEventListener('blur', () =>
m8_removeFocusHighlight('m8_group_title'));

  desc?.addEventListener('focus', () =>
m8_addFocusHighlight('m8_group_desc'));
  desc?.addEventListener('blur', () =>
m8_removeFocusHighlight('m8_group_desc'));

  email?.addEventListener('focus', () =>
m8_addFocusHighlight('m8_group_email'));
  email?.addEventListener('blur', () =>
m8_removeFocusHighlight('m8_group_email'));

  // Knop voor eerste lege veld
  document
    .getElementById('m8_focus_btn')
    ?.addEventListener('click', m8_focusFirstInvalidField);

  // Initiele status
  m8_onFieldChange();
});
```

Oefening M9 — Live Form Feedback & Validatie (Events + event types)

In deze oefening bouw je een klein contactformulier dat **onmiddellijke feedback** geeft terwijl de gebruiker typt of probeert te verzenden. Het doel is om verschillende **event types** te combineren (zoals `input`, `blur` en `submit`) en om te leren werken met `preventDefault()` om de standaard gedragingen van een formulier te beheersen.

De gebruiker ziet drie velden: **naam**, **e-mailadres** en een **bericht**. Onder elk veld staat een hint- of fouttekst. Onderaan staat een meldingsvak dat de algemene status van het formulier toont. Bij het verzenden voorkom je de standaard submit, en toon je een samenvatting.

Je werkt stap voor stap:

1. **Basisstructuur van het formulier**

   Het formulier bevat:

   - Een tekstveld voor de naam
   - Een e-mailveld voor het e-mailadres
   - Een textarea voor het bericht
   - Een verzendknop “Verzend bericht”

   Onderaan staat een grijze `alert` met de tekst `Formulier nog niet verzonden.`

   Je geeft het formulier een `id`, zodat je het in JavaScript kunt selecteren.

2. **Validatieregels per veld**

   Je definieert eenvoudige, maar duidelijke validatieregels:

   - **Naam**: verplicht, minstens 2 tekens na trimmen.
   - **E-mail**: verplicht, bevat minstens een `@` en een `.`.

- **Bericht**: verplicht, minstens 10 tekens na trimmen.

Voor elk veld toon je de status in een klein tekstje onder het inputveld:

- Bij ongeldige waarde:
  - tekst in het rood (`text-danger`)
  - bijvoorbeeld `Naam is verplicht (minstens 2 tekens).`
- Bij geldige waarde:
  - tekst in het groen (`text-success`)
  - bijvoorbeeld `Naam is ok.`

Je mag hiervoor de Bootstrap utility classes gebruiken.

3. **Live feedback via events**

   Je koppelt de juiste events:

   - `input` of `keyup` Telkens de gebruiker typt, evalueer je het veld en werk je de boodschap onder het veld bij.
   - `blur` Wanneer de gebruiker het veld verlaat, controleer je opnieuw en toon je meteen wat er fout is als iets ontbreekt.

   Deze feedback wordt dus **onmiddellijk** bijwerkt zonder dat je het formulier eerst moet verzenden.

4. **Formulier verzenden met preventDefault()**

   Wanneer de gebruiker op de verzendknop klikt of het formulier verzendt (enter in een veld):

   - Je luistert naar het `submit`-event van het formulier.
   - In de event handler gebruik je `event.preventDefault()` om te vermijden dat de pagina herlaadt.
   - Je valideert alle velden nog eens, zodat de laatste stand gegarandeerd juist is.
   - Drie scenario's:

     1. **Minstens één veld ongeldig**
        - Het statusvak wordt een gele waarschuwing (`alert-warning`).
        - Tekst: `Er zijn nog fouten in het formulier. Controleer de gemarkeerde velden.`
     2. **Alle velden geldig**
        - Het statusvak wordt groen (`alert-success`).
        - Tekst: `Formulier is geldig en werd virtueel verzonden.`
        - Je toont onderaan een korte samenvatting:
          - Naam: …
          - E-mail: …
          - Bericht: …
     3. **Formulier al eens verzonden en opnieuw gewijzigd**

- Je mag de status opnieuw aanpassen op basis van de nieuwe validatieresultaten.

5. **Visuele aanduiding van foutieve/en geldige velden**

   Je gebruikt de Bootstrap classes `is-invalid` en `is-valid` op de `input` en `textarea` elementen:

   - Bij ongeldige waarde: `is-invalid` toevoegen, `is-valid` verwijderen.
   - Bij geldige waarde: `is-valid` toevoegen, `is-invalid` verwijderen.

   Zo krijgt de gebruiker naast de tekst ook een duidelijk visueel signaal.

6. **Samenvatting na succesvolle validatie**

   Onder de status alert komt een kleine kaart (card) waarin je na een geldige submit de ingevulde waarden toont in een lijst. Zolang het formulier nog niet geldig is verzonden, toon je een neutrale tekst zoals:

   `Samenvatting verschijnt wanneer het formulier geldig is verzonden.`

Je combineert in deze oefening:

- Form events (`submit`)
- Input events (`input`, `blur`)
- `event.preventDefault()`
- Validatielogica
- DOM-manipulatie van classes en tekst

Oefening M9 — Section (HTML)

```
<section class="container py-4" id="ex-m9">
  <div class="row justify-content-center">
    <div class="col-md-8 col-lg-6">
      <div class="card shadow-sm">
        <div class="card-header bg-info text-white d-flex justify-content-
between align-items-center">
          <span>Oefening M9 – Live Form Feedback &amp; Validatie</span>
          <span class="badge bg-light text-info">Events &amp; Forms</span>
        </div>
        <div class="card-body">
          <form id="m9_form" novalidate>
            <div class="mb-3">
              <label for="m9_name" class="form-label">Naam</label>
              <input
                id="m9_name"
                class="form-control"
                placeholder="bv. Tom"
              />
              <div id="m9_name_help" class="form-text text-muted">
                Naam is verplicht (minstens 2 tekens).
              </div>
            </div>

            <div class="mb-3">
              <label for="m9_email" class="form-label">E-mailadres</label>
              <input
                id="m9_email"
```

```
                type="email"
                class="form-control"
                placeholder="bv. jij@voorbeeld.be"
              />
              <div id="m9_email_help" class="form-text text-muted">
                E-mailadres moet een @ en een punt bevatten.
              </div>
            </div>

            <div class="mb-3">
              <label for="m9_msg" class="form-label">Bericht</label>
              <textarea
                id="m9_msg"
                class="form-control"
                rows="4"
                placeholder="Typ hier je bericht (minstens 10 tekens)."
              ></textarea>
              <div id="m9_msg_help" class="form-text text-muted">
                Bericht is verplicht en minstens 10 tekens lang.
              </div>
            </div>

            <button id="m9_submit" class="btn btn-info w-100 mb-3"
type="submit">
              Verzend bericht
            </button>
          </form>

          <div id="m9_status" class="alert alert-secondary mb-3"
role="status" aria-live="polite">
            Formulier nog niet verzonden.
          </div>

          <div class="card bg-light border-0">
            <div class="card-header bg-white border-0 pb-1">
              <h6 class="card-title mb-0">Samenvatting</h6>
            </div>
            <div class="card-body pt-2">
              <ul id="m9_summary" class="list-unstyled mb-0">
                <li class="text-muted">Samenvatting verschijnt wanneer het
formulier geldig is verzonden.</li>
              </ul>
            </div>
          </div>

        </div>
      </div>
    </div>
  </div>
</section>
```

Oefening M9 — JavaScript-oplossing

```
// ------------------------------------------
// Oefening M9 – Live Form Feedback & Validatie
// ------------------------------------------
```

```
function m9_isNameValid() {
  const value = document.getElementById('m9_name').value.trim();
  return value.length >= 2;
}

function m9_isEmailValid() {
  const value = document.getElementById('m9_email').value.trim();
  return value.includes('@') && value.includes('.');
}

function m9_isMsgValid() {
  const value = document.getElementById('m9_msg').value.trim();
  return value.length >= 10;
}

function m9_updateSingleField(fieldId, helpId, isValidFn, invalidText,
validText) {
  const field = document.getElementById(fieldId);
  const help = document.getElementById(helpId);
  const isValid = isValidFn();

  if (isValid) {
    help.className = 'form-text text-success';
    help.textContent = validText;
    field.classList.remove('is-invalid');
    field.classList.add('is-valid');
  } else {
    help.className = 'form-text text-danger';
    help.textContent = invalidText;
    field.classList.remove('is-valid');
    field.classList.add('is-invalid');
  }
}

function m9_updateAllFieldFeedback() {
  m9_updateSingleField(
    'm9_name',
    'm9_name_help',
    m9_isNameValid,
    'Naam is verplicht (minstens 2 tekens).',
    'Naam is ok.'
  );

  m9_updateSingleField(
    'm9_email',
    'm9_email_help',
    m9_isEmailValid,
    'E-mailadres lijkt ongeldig.',
    'E-mailadres lijkt geldig.'
  );

  m9_updateSingleField(
    'm9_msg',
    'm9_msg_help',
    m9_isMsgValid,
```

```
    'Bericht is te kort (minstens 10 tekens).',
    'Bericht is lang genoeg.'
  );
}

function m9_allValid() {
  return m9_isNameValid() && m9_isEmailValid() && m9_isMsgValid();
}

function m9_updateSummary() {
  const summary = document.getElementById('m9_summary');

  if (!m9_allValid()) {
    summary.innerHTML =
      '<li class="text-muted">Samenvatting verschijnt wanneer het formulier
geldig is verzonden.</li>';
    return;
  }

  const nameVal = document.getElementById('m9_name').value.trim();
  const emailVal = document.getElementById('m9_email').value.trim();
  const msgVal = document.getElementById('m9_msg').value.trim();

  summary.innerHTML = `
    <li><strong>Naam:</strong> ${nameVal}</li>
    <li><strong>E-mailadres:</strong> ${emailVal}</li>
    <li><strong>Bericht:</strong> ${msgVal}</li>
  `;
}

function m9_handleSubmit(event) {
  event.preventDefault();

  const status = document.getElementById('m9_status');

  // Valideer alles nog eens
  m9_updateAllFieldFeedback();

  if (!m9_allValid()) {
    status.className = 'alert alert-warning mb-3';
    status.textContent =
      'Er zijn nog fouten in het formulier. Controleer de gemarkeerde
velden.';
    m9_updateSummary();
    return;
  }

  status.className = 'alert alert-success mb-3';
  status.textContent = 'Formulier is geldig en werd virtueel verzonden.';
  m9_updateSummary();
}

document.addEventListener('DOMContentLoaded', () => {
  const form = document.getElementById('m9_form');
  const nameField = document.getElementById('m9_name');
```

```
  const emailField = document.getElementById('m9_email');
  const msgField = document.getElementById('m9_msg');

  // Live feedback tijdens het typen
  nameField?.addEventListener('input', () => {
    m9_updateSingleField(
      'm9_name',
      'm9_name_help',
      m9_isNameValid,
      'Naam is verplicht (minstens 2 tekens).',
      'Naam is ok.'
    );
  });

  emailField?.addEventListener('input', () => {
    m9_updateSingleField(
      'm9_email',
      'm9_email_help',
      m9_isEmailValid,
      'E-mailadres lijkt ongeldig.',
      'E-mailadres lijkt geldig.'
    );
  });

  msgField?.addEventListener('input', () => {
    m9_updateSingleField(
      'm9_msg',
      'm9_msg_help',
      m9_isMsgValid,
      'Bericht is te kort (minstens 10 tekens).',
      'Bericht is lang genoeg.'
    );
  });

  // Blur-events (nog eens checken bij verlaten veld)
  nameField?.addEventListener('blur', m9_updateAllFieldFeedback);
  emailField?.addEventListener('blur', m9_updateAllFieldFeedback);
  msgField?.addEventListener('blur', m9_updateAllFieldFeedback);

  // Submit-event
  form?.addEventListener('submit', m9_handleSubmit);

  // Init
  m9_updateAllFieldFeedback();
  m9_updateSummary();
});
```

Oefening M10 — Toetsenbord- & Sneltoetsmonitor (Keyboard Events)

In deze oefening bouw je een **toetsenbordmonitor** die laat zien welke toetsen de gebruiker indrukt, hoeveel keer bepaalde toetsen gebruikt werden en welke **modifier keys** (Ctrl, Alt, Shift) actief zijn tijdens een toetsdruk. Daarnaast herken je enkele eenvoudige "sneltoetsen" (combinaties) en geef je daar een duidelijke melding voor.

Het doel is om te leren werken met:

- `keydown`- en `keyup`-events
- properties zoals `event.key`, `event.code`, `event.altKey`, `event.ctrlKey`, `event.shiftKey`
- het tellen en weergeven van statistieken in de DOM
- het conditioneel uitvoeren van acties (bijvoorbeeld speciale melding bij bepaalde toetsen)

De pagina bevat drie grote delen:

1. **Live display van de laatste toetsdruk**
2. **Statistieken over belangrijke toetsen**
3. **Herkenning van sneltoetscombinaties**

1. Live display van de laatste toetsdruk

Bovenaan staat een donkere kaart met een groot displayvak. Telkens de gebruiker een toets indrukt (keydown, waar dan ook op de pagina):

- Toon je in een opvallende weergave:
  - De waarde van `event.key` (gezonde fallback naar tekst als het een speciale toets is zoals "Enter" of "Escape").
  - De waarde van `event.code` (bv. `KeyA`, `ArrowLeft`, `Digit1`).
  - Een lijstje van actieve modifier keys op dat moment:
    - "Ctrl ingedrukt" wanneer `event.ctrlKey` true is.
    - "Alt ingedrukt" wanneer `event.altKey` true is.
    - "Shift ingedrukt" wanneer `event.shiftKey` true is.
- Als er geen modifier keys actief zijn, toon je expliciet `Geen modifier keys actief`.

Het displayvak krijgt de tekst in een groot lettertype (zoals bij een rekenmachine in je M7-oefening), zodat studenten meteen zien welke toets ze eigenlijk hebben ingedrukt.

2. Statistieken over belangrijke toetsen

Onder de live display staat een rij kaarten met counters. Je houdt per toets een teller bij:

- **Enter**
- **Escape**
- **Spatiebalk (Space)**
- **Pijltjestoetsen** (ArrowLeft, ArrowRight, ArrowUp, ArrowDown)

Je kan deze pijltjes samen tellen als "Totaal pijltjestoetsen" of desgewenst apart, maar voor deze oefening beschouw je ze als één groep.

Elke keer dat een van deze toetsen wordt ingedrukt (keydown):

- Verhoog je de juiste teller met 1.
- Werk je de tekst op de bijbehorende kaart bij, bijvoorbeeld:
  - `Enter gedrukt: 5×`
  - `Escape gedrukt: 2×`
  - `Spatiebalk gebruikt: 7×`
  - `Pijltjestoetsen gebruikt: 3×`

Daarnaast voorzie je onderaan een kleine "reset"-knop:

- Knoptekst: `Statistieken resetten`
- Wanneer de gebruiker hierop klikt, worden alle tellers weer op 0 gezet en wordt de weergave aangepast.

3. Herkenning van sneltoetscombinaties

Onderaan staat een meldingsvak dat speciale combinaties herkent wanneer een bepaalde optie ingeschakeld is.

Je voorziet:

- Een checkbox "Sneltoetsmodus inschakelen".
- Een beschrijving welke combinaties herkend worden, bijvoorbeeld:
  - **Ctrl + S** → "Sneltoets: Opslaan"
  - **Ctrl + P** → "Sneltoets: Printen"
  - **Ctrl + Shift + N** → "Sneltoets: Nieuwe incognito-achtige actie"
- Zolang de checkbox **niet** is aangevinkt:
  - Worden toetscombinaties niet speciaal behandeld.
  - Het meldingsvak blijft neutraal (`Sneltoetsmodus staat uit.`).
- Wanneer de checkbox **wel** is aangevinkt en de gebruiker zo'n combinatie indrukt:
  - Verander het meldingsvak naar een groene stijl (`alert-success`).
  - Toon een duidelijke boodschap, zoals:
    - `Sneltoets gedetecteerd: Ctrl + S (Opslaan)`
    - `Sneltoets gedetecteerd: Ctrl + P (Printen)`
    - `Sneltoets gedetecteerd: Ctrl + Shift + N (Nieuw venster)`
  - Gebruik `event.preventDefault()` voor deze combinaties zodat de browseractie (zoals printdialoog) idealiter niet wordt uitgevoerd. (Opmerking: sommige browsers kunnen toch hun eigen sneltoetsen voorrang geven, maar voor de oefening is dit voldoende.)

Als een toets wordt ingedrukt die niet overeenkomt met een van je gedefinieerde sneltoetsen:

- En de sneltoetsmodus staat aan:

- Zet het meldingsvak terug naar een informatieve stijl (`alert-info`) met een tekst zoals:
  - `Geen bekende sneltoets.`

- En de sneltoetsmodus staat uit:
  - Houd het vak grijs met de tekst:
    - `Sneltoetsmodus staat uit.`

Technische aandachtspunten

- Je gebruikt één `keydown`-listener op `document` om alle toetsdrukken te volgen.
- Bij het updaten van de DOM zorg je dat:
  - je bestaande elementen selecteert met `getElementById` of `querySelector`
  - je de tekst aanpast met `textContent`
  - je classes bijwerkt met `classList`
- Voor de counters gebruik je eenvoudige variabelen in JavaScript, bijvoorbeeld:
  - `let m10_enterCount = 0;`
  - `let m10_escapeCount = 0;`
  - `let m10_spaceCount = 0;`
  - `let m10_arrowCount = 0;`
- Gebruik duidelijke functienamen zoals `m10_updateDisplay(event)`, `m10_updateCounters(event)` en `m10_checkShortcut(event)` om de logica overzichtelijk te houden.

Oefening M10 — Section (HTML)

```
<section class="container py-4" id="ex-m10">
  <div class="row justify-content-center">
    <div class="col-md-10 col-lg-8">
      <div class="card shadow-sm mb-4">
        <div class="card-header bg-dark text-white d-flex justify-content-
between align-items-center">
          <span>Oefening M10 – Toetsenbord- &amp;amp; Sneltoetsmonitor</span>
          <span class="badge bg-light text-dark">Keyboard events</span>
        </div>

        <div class="card-body bg-black text-white">
          <!-- Live display -->
          <div class="mb-3">
            <div class="small text-uppercase text-muted">Laatste toets</div>
            <div id="m10_key_display" class="fs-1 fw-bold text-end border-0
bg-black" aria-live="polite">
              –
            </div>
          </div>

          <div class="row g-3">
            <div class="col-md-6">
              <div class="p-3 bg-dark rounded-3 h-100">
                <div class="small text-muted text-uppercase mb-1">Key</div>
                <div id="m10_key_value" class="fs-5">-</div>
```

```
            </div>
          </div>
          <div class="col-md-6">
            <div class="p-3 bg-dark rounded-3 h-100">
              <div class="small text-muted text-uppercase mb-1">Code</div>
              <div id="m10_key_code" class="fs-5">-</div>
            </div>
          </div>
        </div>

        <div class="mt-3 p-3 bg-dark rounded-3">
          <div class="small text-muted text-uppercase mb-1">Modifier
keys</div>
          <ul id="m10_modifiers" class="mb-0 small">
            <li>Geen modifier keys actief.</li>
          </ul>
        </div>
      </div>
    </div>

    <!-- Statistieken -->
    <div class="card shadow-sm mb-4">
      <div class="card-header bg-secondary text-white d-flex justify-
content-between align-items-center">
        <span>Toetsenstatistieken</span>
        <button id="m10_reset_stats" class="btn btn-sm btn-outline-light">
          Statistieken resetten
        </button>
      </div>
      <div class="card-body">
        <div class="row g-3">
          <div class="col-md-6 col-lg-3">
            <div class="border rounded-3 p-3 h-100 text-center">
              <div class="small text-muted">Enter</div>
              <div id="m10_enter_count" class="fs-4 fw-bold">0</div>
              <div class="small text-muted">keer gedrukt</div>
            </div>
          </div>
          <div class="col-md-6 col-lg-3">
            <div class="border rounded-3 p-3 h-100 text-center">
              <div class="small text-muted">Escape</div>
              <div id="m10_escape_count" class="fs-4 fw-bold">0</div>
              <div class="small text-muted">keer gedrukt</div>
            </div>
          </div>
          <div class="col-md-6 col-lg-3">
            <div class="border rounded-3 p-3 h-100 text-center">
              <div class="small text-muted">Spatiebalk</div>
              <div id="m10_space_count" class="fs-4 fw-bold">0</div>
              <div class="small text-muted">keer gebruikt</div>
            </div>
          </div>
          <div class="col-md-6 col-lg-3">
            <div class="border rounded-3 p-3 h-100 text-center">
              <div class="small text-muted">Pijltjestoetsen</div>
              <div id="m10_arrow_count" class="fs-4 fw-bold">0</div>
```

```
                <div class="small text-muted">totaal gebruikt</div>
              </div>
            </div>
          </div>
        </div>
      </div>

      <!-- Sneltoetsmodus -->
      <div class="card shadow-sm">
        <div class="card-header bg-info text-white">
          Sneltoetsmodus
        </div>
        <div class="card-body">
          <div class="form-check form-switch mb-3">
            <input class="form-check-input" type="checkbox"
id="m10_shortcut_mode">
            <label class="form-check-label" for="m10_shortcut_mode">
              Sneltoetsmodus inschakelen
            </label>
          </div>

          <ul class="small text-muted mb-3">
            <li><strong>Ctrl + S</strong> → Opslaan</li>
            <li><strong>Ctrl + P</strong> → Printen</li>
            <li><strong>Ctrl + Shift + N</strong> → Nieuw venster</li>
          </ul>

          <div id="m10_shortcut_status" class="alert alert-secondary mb-0"
role="status" aria-live="polite">
            Sneltoetsmodus staat uit.
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

Oefening M10 — JavaScript-oplossing

```
// ------------------------------------------
// Oefening M10 – Toetsenbord- & Sneltoetsmonitor
// ------------------------------------------

let m10_enterCount = 0;
let m10_escapeCount = 0;
let m10_spaceCount = 0;
let m10_arrowCount = 0;

function m10_updateDisplay(event) {
  const display = document.getElementById('m10_key_display');
  const keyVal = document.getElementById('m10_key_value');
  const keyCode = document.getElementById('m10_key_code');
  const modifiers = document.getElementById('m10_modifiers');

  const keyText = event.key === ' ' ? 'Space' : event.key;

  display.textContent = keyText;
```

```
  keyVal.textContent = keyText;
  keyCode.textContent = event.code;

  const items = [];

  if (event.ctrlKey) items.push('Ctrl ingedrukt');
  if (event.altKey) items.push('Alt ingedrukt');
  if (event.shiftKey) items.push('Shift ingedrukt');

  if (items.length === 0) {
    modifiers.innerHTML = '<li>Geen modifier keys actief.</li>';
  } else {
    modifiers.innerHTML = items.map(text => `<li>${text}</li>`).join('');
  }
}

function m10_updateCounters(event) {
  const enterEl = document.getElementById('m10_enter_count');
  const escapeEl = document.getElementById('m10_escape_count');
  const spaceEl = document.getElementById('m10_space_count');
  const arrowEl = document.getElementById('m10_arrow_count');

  switch (event.key) {
    case 'Enter':
      m10_enterCount++;
      enterEl.textContent = m10_enterCount;
      break;
    case 'Escape':
      m10_escapeCount++;
      escapeEl.textContent = m10_escapeCount;
      break;
    case ' ':
      m10_spaceCount++;
      spaceEl.textContent = m10_spaceCount;
      break;
    default:
      break;
  }

  if (
    event.code === 'ArrowLeft' ||
    event.code === 'ArrowRight' ||
    event.code === 'ArrowUp' ||
    event.code === 'ArrowDown'
  ) {
    m10_arrowCount++;
    arrowEl.textContent = m10_arrowCount;
  }
}

function m10_resetStats() {
  m10_enterCount = 0;
  m10_escapeCount = 0;
  m10_spaceCount = 0;
  m10_arrowCount = 0;
```

```
  document.getElementById('m10_enter_count').textContent = '0';
  document.getElementById('m10_escape_count').textContent = '0';
  document.getElementById('m10_space_count').textContent = '0';
  document.getElementById('m10_arrow_count').textContent = '0';
}

function m10_updateShortcutStatus(text, type) {
  const status = document.getElementById('m10_shortcut_status');
  status.className = `alert alert-${type} mb-0`;
  status.textContent = text;
}

function m10_checkShortcut(event) {
  const shortcutMode = document.getElementById('m10_shortcut_mode');
  const enabled = shortcutMode.checked;

  if (!enabled) {
    m10_updateShortcutStatus('Sneltoetsmodus staat uit.', 'secondary');
    return;
  }

  // Standaard: geen bekende sneltoets
  let handled = false;

  // Ctrl + S
  if (event.ctrlKey && !event.shiftKey && !event.altKey && (event.key === 's'
|| event.key === 'S')) {
    event.preventDefault();
    m10_updateShortcutStatus('Sneltoets gedetecteerd: Ctrl + S (Opslaan)',
'success');
    handled = true;
  }

  // Ctrl + P
  if (event.ctrlKey && !event.shiftKey && !event.altKey && (event.key === 'p'
|| event.key === 'P')) {
    event.preventDefault();
    m10_updateShortcutStatus('Sneltoets gedetecteerd: Ctrl + P (Printen)',
'success');
    handled = true;
  }

  // Ctrl + Shift + N
  if (
    event.ctrlKey &&
    event.shiftKey &&
    !event.altKey &&
    (event.key === 'n' || event.key === 'N')
  ) {
    event.preventDefault();
    m10_updateShortcutStatus('Sneltoets gedetecteerd: Ctrl + Shift + N (Nieuw
venster)', 'success');
    handled = true;
  }
```

```
  if (!handled) {
    m10_updateShortcutStatus('Geen bekende sneltoets.', 'info');
  }
}

function m10_handleKeyDown(event) {
  m10_updateDisplay(event);
  m10_updateCounters(event);
  m10_checkShortcut(event);
}

document.addEventListener('DOMContentLoaded', () => {
  document.addEventListener('keydown', m10_handleKeyDown);

  document.getElementById('m10_reset_stats')?.addEventListener('click', () =>
{
    m10_resetStats();
  });

  // Init shortcut status
  m10_updateShortcutStatus('Sneltoetsmodus staat uit.', 'secondary');

  document.getElementById('m10_shortcut_mode')?.addEventListener('change', ()
=> {
    const shortcutMode = document.getElementById('m10_shortcut_mode');
    if (shortcutMode.checked) {
      m10_updateShortcutStatus('Sneltoetsmodus is ingeschakeld. Probeer
Ctrl+S, Ctrl+P of Ctrl+Shift+N.', 'info');
    } else {
      m10_updateShortcutStatus('Sneltoetsmodus staat uit.', 'secondary');
    }
  });
});
```

Oefening M11 — Geboortedag & Verjaardagsinfo (Date-object)

In deze oefening gebruik je het **Date-object** om op basis van een geboortedatum allerlei nuttige informatie te tonen. De gebruiker voert zijn geboortedatum in via een datumveld. Jij berekent met JavaScript:

1. Hoe oud de gebruiker ongeveer is in jaren.
2. Op welke dag van de week hij/zij geboren is.
3. Wanneer de volgende verjaardag is.
4. Hoeveel dagen het nog is tot die volgende verjaardag.

Het doel is om vertrouwd te raken met:

- `new Date()` en het parsen van een datum uit een inputveld
- methodes zoals `getFullYear()`, `getMonth()`, `getDate()`, `getDay()`
- rekenwerk met milliseconden om het aantal dagen tussen twee datums te benaderen
- het opbouwen van een duidelijke tekstuele samenvatting op basis van deze berekeningen

Je bouwt het gedrag in enkele stappen:

1. Datum invoeren en basiscontrole

Bovenaan in de kaart staat:

- Een `input type="date"` voor de geboortedatum.
- Een korte uitleg in een `form-text` onder het veld met wat er precies berekend zal worden.
- Een knop “Bereken gegevens”.

Als de gebruiker op de knop klikt:

- Lees je de waarde van het datumveld.
- Als de waarde leeg is of niet kan worden omgezet naar een geldige `Date`, toon je meteen een gele waarschuwing in het meldingsvak, met de tekst: `Vul een geldige geboortedatum in.`
- Je voert dan geen verdere berekeningen uit.

2. Leeftijd berekenen

Wanneer de datum geldig is:

- Maak een `Date`-object met de geboortedatum, bijvoorbeeld `const birth = new Date(value)`.
- Maak ook een `Date`-object voor “nu”: `const now = new Date()`.
- Bereken een eerste schatting van de leeftijd in jaren: `now.getFullYear() - birth.getFullYear()`.
- Controleer daarna of de verjaardag dit jaar al voorbij is:
  - De verjaardag is al voorbij als:
    - de huidige maand groter is dan de geboortemaand, of

    - de maanden gelijk zijn en de huidige dag ≥ geboortedag.
  - Als de verjaardag nog **niet** geweest is, verminder je de leeftijd met 1 jaar.

De leeftijd is een eenvoudig geheel getal, zoals `37`. Je noteert dit als `Je bent ongeveer 37 jaar oud.` in de uiteindelijke samenvatting.

3. Dag van de week van geboorte

Met `birth.getDay()` krijg je een getal van 0 (zondag) tot 6 (zaterdag). Je maakt zelf een array met Nederlandse namen, bijvoorbeeld:

- `['Zondag', 'Maandag', 'Dinsdag', 'Woensdag', 'Donderdag', 'Vrijdag', 'Zaterdag']`.

Zo kan je bijvoorbeeld tonen:

- `Je bent geboren op een Woensdag.`

4. Volgende verjaardag bepalen

Je berekent een `Date`-object dat de eerstvolgende verjaardag voorstelt:

- Start met een datum in het huidige jaar met dezelfde maand en dag als de geboortedatum: `let nextBirthday = new Date(now.getFullYear(), birth.getMonth(), birth.getDate());`
- Als deze datum **voor** vandaag ligt, betekent het dat de verjaardag dit jaar al voorbij is. In dat geval verhoog je het jaar met 1: `nextBirthday = new Date(now.getFullYear() + 1, birth.getMonth(), birth.getDate());`

Deze `nextBirthday` gebruik je zowel voor een nette datumweergave (met `toLocaleDateString()`) als voor de resterende dagen.

5. Aantal dagen tot volgende verjaardag

Je berekent het verschil in milliseconden tussen de volgende verjaardag en nu:

- `const diffMs = nextBirthday - now;`

Daarna reken je dit om naar dagen:

- `const diffDays = Math.ceil(diffMs / (1000 * 60 * 60 * 24));`

Je rondt naar boven af met `Math.ceil`, zodat je geen `0.3` dagen krijgt maar gewoon “1 dag”, enzovoort.

Je toont dit in de tekst als: `Je volgende verjaardag is over 128 dag(en).`

6. Overzicht in het meldingsvak en aparte samenvattingskaart

Je geeft de resultaten in twee lagen:

1. **Meldingsvak (alert)**
   - Krijgt een groene stijl bij een geldige datum.
   - Bevat één korte samenvattende zin, bijvoorbeeld: `Je bent ongeveer 37 jaar oud. Je bent geboren op een Woensdag.`
2. **Samenvattingslijst**

- In een card onder de alert toon je in een lijst:
  - Leeftijd: 37 jaar
  - Geboortedag: Woensdag
  - Volgende verjaardag: 13/05/2026
  - Dagen tot volgende verjaardag: 128

Zolang er nog geen geldige datum is ingevoerd, toon je in deze lijst een neutrale placeholdertekst zoals: `Samenvatting verschijnt zodra een geldige geboortedatum is berekend.`

Oefening M11 — Section (HTML)

```
<section class="container py-4" id="ex-m11">
  <div class="row justify-content-center">
    <div class="col-md-8 col-lg-6">
      <div class="card shadow-sm">
        <div class="card-header bg-secondary text-white d-flex justify-
content-between align-items-center">
          <span>Oefening M11 – Geboortedag &amp; Verjaardagsinfo</span>
          <span class="badge bg-light text-secondary">Date-object</span>
        </div>
        <div class="card-body">

          <div class="mb-3">
            <label for="m11_date" class="form-label">Geboortedatum</label>
            <input id="m11_date" type="date" class="form-control" />
            <div class="form-text">
              We berekenen je leeftijd, de dag van de week waarop je geboren
bent,
              en hoe lang het nog duurt tot je volgende verjaardag.
            </div>
          </div>

          <button id="m11_btn" class="btn btn-secondary w-100 mb-3">
            Bereken gegevens
          </button>

          <div id="m11_status" class="alert alert-secondary mb-3"
role="status" aria-live="polite">
            Vul je geboortedatum in en klik op "Bereken gegevens".
          </div>

          <div class="card bg-light border-0">
            <div class="card-header bg-white border-0 pb-1">
              <h6 class="card-title mb-0">Samenvatting</h6>
            </div>
            <div class="card-body pt-2">
              <ul id="m11_summary" class="list-unstyled mb-0">
                <li class="text-muted">
                  Samenvatting verschijnt zodra een geldige geboortedatum is
berekend.
                </li>
              </ul>
            </div>
          </div>
```

```
        </div>
      </div>
    </div>
  </div>
</section>
```

Oefening M11 — JavaScript-oplossing

```
// ------------------------------------------------
// Oefening M11 – Geboortedag & Verjaardagsinfo
// ------------------------------------------------

const M11_DAYS = [
  'Zondag',
  'Maandag',
  'Dinsdag',
  'Woensdag',
  'Donderdag',
  'Vrijdag',
  'Zaterdag'
];

function m11_calculateInfo(dateStr) {
  const birth = new Date(dateStr);
  const now = new Date();

  if (Number.isNaN(birth.getTime())) {
    return null;
  }

  // Leeftijd
  let age = now.getFullYear() - birth.getFullYear();

  const hasHadBirthdayThisYear =
    now.getMonth() > birth.getMonth() ||
    (now.getMonth() === birth.getMonth() && now.getDate() >=
birth.getDate());

  if (!hasHadBirthdayThisYear) {
    age--;
  }

  // Dag van de week
  const dayName = M11_DAYS[birth.getDay()];

  // Volgende verjaardag (huidig jaar of volgend jaar)
  let nextBirthday = new Date(
    now.getFullYear(),
    birth.getMonth(),
    birth.getDate()
  );

  if (nextBirthday < now) {
    nextBirthday = new Date(
      now.getFullYear() + 1,
```

```
      birth.getMonth(),
      birth.getDate()
    );
  }

  // Dagen tot volgende verjaardag
  const diffMs = nextBirthday - now;
  const diffDays = Math.ceil(diffMs / (1000 * 60 * 60 * 24));

  return {
    age,
    dayName,
    nextBirthday,
    diffDays
  };
}

function m11_updateUI(result) {
  const status = document.getElementById('m11_status');
  const summary = document.getElementById('m11_summary');

  if (!result) {
    status.className = 'alert alert-warning mb-3';
    status.textContent = 'Vul een geldige geboortedatum in.';
    summary.innerHTML =
      '<li class="text-muted">Samenvatting verschijnt zodra een geldige
geboortedatum is berekend.</li>';
    return;
  }

  const { age, dayName, nextBirthday, diffDays } = result;

  status.className = 'alert alert-success mb-3';
  status.textContent = `Je bent ongeveer ${age} jaar oud. Je bent geboren op
een ${dayName}.`;

  const nextBirthdayStr = nextBirthday.toLocaleDateString();

  summary.innerHTML = `
    <li><strong>Leeftijd:</strong> ${age} jaar</li>
    <li><strong>Geboortedag:</strong> ${dayName}</li>
    <li><strong>Volgende verjaardag:</strong> ${nextBirthdayStr}</li>
    <li><strong>Dagen tot volgende verjaardag:</strong> ${diffDays}</li>
  `;
}

function m11_handleClick() {
  const input = document.getElementById('m11_date');
  const value = input.value;

  if (!value) {
    m11_updateUI(null);
    return;
  }
```

```
  const result = m11_calculateInfo(value);
  m11_updateUI(result);
}

document.addEventListener('DOMContentLoaded', () => {
  const btn = document.getElementById('m11_btn');
  const input = document.getElementById('m11_date');

  btn?.addEventListener('click', m11_handleClick);

  // Optioneel: bij Enter in het datumveld ook berekenen
  input?.addEventListener('keyup', (event) => {
    if (event.key === 'Enter') {
      m11_handleClick();
    }
  });
});
```

Oefening M12 — Gebruikers en admins met ES6 classes

In deze oefening ontwerp je een klein systeem waarmee je via een formulier **gebruikers** en **admins** kunt aanmaken, op basis van **ES6 classes**. Je werkt met een basisklasse `User` en een subklasse `Admin` die erft van `User`. Via een eenvoudige interface voeg je personen toe, toon je ze in mooie kaarten en laat je duidelijk zien wie gewone gebruiker is en wie beheerder.

Het doel is om te oefenen met:

- ES6 `class` en `extends`
- `constructor` en `super`
- methodes op de class (bijvoorbeeld `getLabel()` of `getRoleBadgeText()`)
- een lijst van objecten bijhouden in een array
- dynamisch HTML opbouwen op basis van die lijst

De pagina bevat:

1. Een formulier om een nieuwe gebruiker of admin in te voeren
2. Een overzicht van alle aangemaakte personen in een grid van kaarten
3. Een teller met het aantal gebruikers en admins

1. Formulier voor aanmaken van personen

In de kaart staat bovenaan een formulier met drie velden:

- `Naam`
- `Leeftijd`
- `Rol`: een keuzelijst met opties `User` en `Admin`

Daarnaast is er een knop `Maak account`.

Wanneer de gebruiker op de knop klikt:

- Lees je de waarden van de drie velden.
- Je voert een eenvoudige validatie uit:

  - Naam mag niet leeg zijn na trim.
  - Leeftijd moet een getal groter dan nul zijn.
- Als een van deze controles faalt:
  - Toon je een `alert` pop up via `alert(...)` met een duidelijke foutboodschap, bijvoorbeeld `Vul een geldige naam en leeftijd in.`
  - Je maakt geen object aan en past de lijst niet aan.

2. ES6 classes User en Admin

Je definieert in JavaScript:

- Een class `User` met minstens:
  - eigenschappen `name`, `age`, `role`
  - een constructor die `name` en `age` als parameters krijgt, en `role` instelt op `"User"`
  - een methode `getLabel()` die bijvoorbeeld een string teruggeeft zoals:
    - `Sara (25) is een gewone gebruiker.`
- Een class `Admin` die `extends User`:
  - roept in de constructor `super(name, age)` aan
  - overschrijft de `role` naar `"Admin"`
  - mag `getLabel()` overschrijven om duidelijk te maken dat het om een admin gaat, bijvoorbeeld:
    - `Tom (34) is administrator.`

In de rest van de code behandel je beide typen objecten op dezelfde manier, maar de weergave verschilt licht door de rol.

3. Lijst van accounts bijhouden

Je houdt alle aangemaakte accounts bij in een array, bijvoorbeeld `const m12_accounts = []`.

Wanneer de knop `Maak account` wordt gebruikt:

- Maak je, afhankelijk van de gekozen rol, een nieuwe instantie:
  - Bij rol `User`: `new User(name, age)`
  - Bij rol `Admin`: `new Admin(name, age)`
- Duw je dit object in de array `m12_accounts`.
- Roep je een functie aan die de volledige lijst opnieuw tekent in de DOM.

4. Overzicht van accounts als kaarten

Onder het formulier staat een responsieve grid (Bootstrap row) waarin elke account als kaart wordt getoond. Een kaart toont:

- Bovenaan: naam en rol

  - Naam als titel
  - Rol via een badge, bijvoorbeeld:
    - Grijze badge voor `User`
    - Rode badge voor `Admin`
- In de kaarttekst:
  - Leeftijd
  - Eventueel extra tekst op basis van `getLabel()`, bijvoorbeeld in kleinere, lichtgrijze tekst onderaan.

Je bouwt de kaarten dynamisch op met `innerHTML` op basis van de inhoud van de `m12_accounts` array.

5. Tellers voor aantal users en admins

Boven de grid staat een klein statusbalkje waarin je twee tellers toont:

- `Totaal accounts`
- `Users`
- `Admins`

Bij elke verandering van de lijst (toevoegen):

- Tel je met `filter` of een eenvoudige lus hoeveel objecten `role === 'User'` zijn en hoeveel `role === 'Admin'`.
- Werk je de tekst van de teller elementen bij in de DOM.

6. Formulier resetten na het toevoegen

Wanneer een nieuw account succesvol werd aangemaakt:

- Maak je de naam en leeftijd velden leeg.
- Zet je de rol terug naar de standaardwaarde `User`.
- Zet je de focus terug op het naamveld zodat de gebruiker vlot een volgende invoer kan doen.

Oefening M12 — Section (HTML)

```
<section class="container py-4" id="ex-m12">
  <div class="row justify-content-center">
    <div class="col-md-10 col-lg-8">
      <div class="card shadow-sm">
        <div class="card-header bg-primary text-white d-flex justify-content-
between align-items-center">
          <span>Oefening M12 — Gebruikers en admins met ES6 classes</span>
          <span class="badge bg-light text-primary">Classes</span>
        </div>
        <div class="card-body">

          <!-- Invoerformulier -->
          <div class="row g-3 mb-3">
            <div class="col-md-4">
              <label for="m12_name" class="form-label">Naam</label>
              <input
                id="m12_name"
```

```
                class="form-control"
                placeholder="bv. Sara"
              />
            </div>
            <div class="col-md-4">
              <label for="m12_age" class="form-label">Leeftijd</label>
              <input
                id="m12_age"
                type="number"
                class="form-control"
                placeholder="bv. 25"
                min="1"
              />
            </div>
            <div class="col-md-4">
              <label for="m12_role" class="form-label">Rol</label>
              <select id="m12_role" class="form-select">
                <option value="user" selected>User</option>
                <option value="admin">Admin</option>
              </select>
            </div>
          </div>

          <button id="m12_add" class="btn btn-primary w-100 mb-3">
            Maak account
          </button>

          <!-- Tellers -->
          <div class="alert alert-secondary d-flex flex-wrap justify-content-
between align-items-center mb-3">
            <span>
              Totaal accounts:
              <strong id="m12_total_count">0</strong>
            </span>
            <span class="small">
              Users:
              <strong id="m12_user_count">0</strong>
               | 
              Admins:
              <strong id="m12_admin_count">0</strong>
            </span>
          </div>

          <!-- Lijst van accounts -->
          <div id="m12_list" class="row g-3" aria-live="polite">
            <div class="col-12">
              <p class="text-muted mb-0">
                Nog geen accounts aangemaakt, vul het formulier in en klik op
"Maak account".
              </p>
            </div>
          </div>

        </div>
      </div>
    </div>
```

```
  </div>
</section>
```

Oefening M12 — JavaScript-oplossing

```
// ------------------------------------------------------
// Oefening M12 – Gebruikers en admins met ES6 classes
// ------------------------------------------------------

class M12User {
  constructor(name, age) {
    this.name = name;
    this.age = age;
    this.role = 'User';
  }

  getLabel() {
    return `${this.name} (${this.age}) is een gewone gebruiker.`;
  }
}

class M12Admin extends M12User {
  constructor(name, age) {
    super(name, age);
    this.role = 'Admin';
  }

  getLabel() {
    return `${this.name} (${this.age}) is administrator.`;
  }
}

const m12_accounts = [];

function m12_updateCounters() {
  const totalEl = document.getElementById('m12_total_count');
  const userEl = document.getElementById('m12_user_count');
  const adminEl = document.getElementById('m12_admin_count');

  const total = m12_accounts.length;
  const users = m12_accounts.filter(acc => acc.role === 'User').length;
  const admins = m12_accounts.filter(acc => acc.role === 'Admin').length;

  totalEl.textContent = total;
  userEl.textContent = users;
  adminEl.textContent = admins;
}

function m12_renderList() {
  const list = document.getElementById('m12_list');

  if (m12_accounts.length === 0) {
    list.innerHTML = `
      <div class="col-12">
        <p class="text-muted mb-0">
          Nog geen accounts aangemaakt, vul het formulier in en klik op "Maak
```

```
account".
        </p>
      </div>
    `;
    return;
  }

  list.innerHTML = m12_accounts
    .map(acc => {
      const isAdmin = acc.role === 'Admin';
      const badgeClass = isAdmin ? 'bg-danger' : 'bg-secondary';
      const roleText = isAdmin ? 'Admin' : 'User';

      return `
        <div class="col-md-6 col-lg-4">
          <div class="card h-100 shadow-sm">
            <div class="card-body d-flex flex-column">
              <div class="d-flex justify-content-between align-items-start
mb-2">
                <h5 class="card-title mb-0">${acc.name}</h5>
                <span class="badge ${badgeClass}">${roleText}</span>
              </div>
              <p class="card-text mb-1">
                Leeftijd: <strong>${acc.age}</strong>
              </p>
              <p class="card-text small text-muted mt-auto">
                ${acc.getLabel()}
              </p>
            </div>
          </div>
        </div>
      `;
    })
    .join('');
}

function m12_addAccount() {
  const nameInp = document.getElementById('m12_name');
  const ageInp = document.getElementById('m12_age');
  const roleSel = document.getElementById('m12_role');

  const name = nameInp.value.trim();
  const age = Number(ageInp.value);
  const role = roleSel.value;

  if (!name || !age || age <= 0) {
    alert('Vul een geldige naam en leeftijd in.');
    return;
  }

  let account;
  if (role === 'admin') {
    account = new M12Admin(name, age);
  } else {
    account = new M12User(name, age);
  }
```

```
  m12_accounts.push(account);
  m12_updateCounters();
  m12_renderList();

  // Formulier resetten
  nameInp.value = '';
  ageInp.value = '';
  roleSel.value = 'user';
  nameInp.focus();
}

document.addEventListener('DOMContentLoaded', () => {
  document.getElementById('m12_add')?.addEventListener('click',
m12_addAccount);
  m12_updateCounters();
  m12_renderList();
});
```

Oefening M13 - Productcatalogus met filters en ES6 classes

In deze oefening bouw je een eenvoudige productcatalogus die producten toont in een grid. Je gebruikt **ES6 classes** om producten te modelleren, en je laat de gebruiker via een paar filters kiezen welke producten zichtbaar zijn. Het doel is om te oefenen met:

- Een `Product` class met eigenschappen zoals `naam`, `categorie`, `prijs` en `beschrijving`.
- Een array van product objecten.
- Filteren met JavaScript op basis van selectievakjes en invoervelden.
- Dynamisch HTML opbouwen met kaarten (Bootstrap cards).

De interface bestaat uit:

1. Een filterbalk met categorie selectie en maximale prijs.
2. Een informatieve meldingsbox met het aantal gevonden producten.
3. Een grid van productkaarten.

1. Product klasse en datastructuur

Je modelleert een product met een ES6 class:

- `name` (string, bijvoorbeeld “JavaScript Basis”).
- `category` (string, bijvoorbeeld “books”, “courses”, “software”).
- `price` (number, in euro).
- `description` (korte beschrijving).
- Optioneel een `level` of `tag` die je in de kaart kan tonen.

Bijvoorbeeld:

- Boek “JavaScript Basis”, categorie `books`, prijs 29.99, beschrijving “Boek voor absolute beginners in JS”.
- Cursus “HTML en CSS”, categorie `courses`.
- Software “Task Manager App”, categorie `software`.

Je maakt in JavaScript een vaste lijst van enkele producten (minstens 5 tot 6), zodat de student meteen iets heeft om mee te werken.

2. Filterbalk

Boven de catalogus komt een kleine filterbalk met:

- Een dropdown “Categorie” met opties:
  - “Alle categorieën”
  - “Boeken”
  - “Cursussen”
  - “Software”
- Een numeriek veld “Maximale prijs” (bijvoorbeeld placeholder “bijv. 50”).
- Een knop “Filter toepassen”.

Werking:

- Wanneer de gebruiker op “Filter toepassen” klikt:
  - Lees je de geselecteerde categorie.
    - Als “Alle categorieën” is geselecteerd, filter je niet op categorie.
    - Zo niet, toon je enkel producten met die categorie.
  - Lees je de maximale prijs.
    - Als dit veld leeg is of niet positief, negeer je de prijsfilter.
    - Anders toon je enkel producten waarvan de prijs kleiner of gelijk is aan deze waarde.

Je mag de mapping tussen zichtbare naam en interne categorie zelf bepalen, bijvoorbeeld:

- Dropdown “Boeken” -> category “books”
- Dropdown “Cursussen” -> category “courses”
- Dropdown “Software” -> category “software”

3. Meldingsvak met aantal resultaten

Onder de filterbalk staat een alert die aangeeft wat het resultaat is van de filtering:

- Als er nog niets gefilterd is (bij eerste load), toon je een informatieve tekst, bijvoorbeeld: `Er worden momenteel alle producten getoond.`
- Als de filter is toegepast:
  - Bij minstens één gevonden product:
    - groene of blauwe alert
    - tekst zoals: `Gevonden producten: 4`
  - Bij nul resultaten:
    - waarschuwingsalert
    - tekst: `Geen producten gevonden met deze filter.`

De alert krijgt dus een andere klasse afhankelijk van de situatie: `alert-secondary`, `alert-success`, `alert-info` of `alert-warning`.

4. Grid van productkaarten

De producten zelf worden getoond in een responsive grid met Bootstrap kolommen:

- Je gebruikt een `row` en binnenin `col-md-6 col-lg-4` voor de kaarten.
- Elke kaart toont:
  - Productnaam als titel.
  - Categorie als badge, bijvoorbeeld:
    - `Boek` met een neutrale badge.
    - `Cursus` in een andere kleur.
    - `Software` in een andere kleur.
  - Beschrijving in de body.
  - Prijs onderaan in een vetgedrukte stijl, bijvoorbeeld `€29,99`.

De kaarten worden volledig door JavaScript gegenereerd op basis van de gefilterde productlijst. Als er geen producten zijn, moet de container leeg zijn en benut je het meldingsvak om dit te melden.

5. Initieel gedrag

Bij het laden van de pagina:

- Toon je alle producten, zonder filters.
- Zet je de alert in een neutrale, grijze stijl met de tekst: `Er worden momenteel alle producten getoond.`

Pas wanneer de gebruiker op “Filter toepassen” klikt, pas je de filters toe en werk je de alert tekst en stijl bij.

Zo combineert deze oefening classes, arrays, filter logica en DOM manipulatie met Bootstrap styling.

Oefening M13 - Section (HTML)

```
<section class="container py-4" id="ex-m13">
  <div class="row justify-content-center">
    <div class="col-md-11 col-lg-9">
      <div class="card shadow-sm">
        <div class="card-header bg-success text-white d-flex justify-content-
between align-items-center">
          <span>Oefening M13 - Productcatalogus met filters</span>
          <span class="badge bg-light text-success">Classes + filter</span>
        </div>
        <div class="card-body">

          <!-- Filterbalk -->
          <div class="row g-3 align-items-end mb-3">
            <div class="col-md-4">
              <label for="m13_cat" class="form-label">Categorie</label>
              <select id="m13_cat" class="form-select">
```

```
                <option value="all" selected>Alle categorieën</option>
                <option value="books">Boeken</option>
                <option value="courses">Cursussen</option>
                <option value="software">Software</option>
              </select>
            </div>
            <div class="col-md-4">
              <label for="m13_max" class="form-label">Maximale prijs
(€)</label>
              <input
                id="m13_max"
                type="number"
                step="0.01"
                class="form-control"
                placeholder="bijv. 50"
                min="0"
              />
            </div>
            <div class="col-md-4 d-grid">
              <button id="m13_filter" class="btn btn-success">
                Filter toepassen
              </button>
            </div>
          </div>

          <!-- Info over resultaten -->
          <div id="m13_status" class="alert alert-secondary mb-3"
role="status" aria-live="polite">
            Er worden momenteel alle producten getoond.
          </div>

          <!-- Grid met producten -->
          <div id="m13_products" class="row g-3">
            <!-- Kaarten worden hier via JS ingevuld -->
          </div>

        </div>
      </div>
    </div>
  </div>
</section>
```

Oefening M13 - JavaScript oplossing

```
// ------------------------------------------------
// Oefening M13 - Productcatalogus met filters
// ------------------------------------------------

// ES6 class voor een product
class M13Product {
  constructor(name, category, price, description) {
    this.name = name;
    this.category = category; // 'books', 'courses', 'software'
    this.price = price;       // number
    this.description = description;
  }
}
```

```
// Vooraf gedefinieerde lijst met producten
const m13_allProducts = [
  new M13Product(
    'JavaScript Basis',
    'books',
    29.99,
    'Boek voor absolute beginners in JavaScript.'
  ),
  new M13Product(
    'Geavanceerde JavaScript Patterns',
    'books',
    49.5,
    'Voor ontwikkelaars die hun JS kennis willen verdiepen.'
  ),
  new M13Product(
    'Online cursus HTML en CSS',
    'courses',
    39,
    'Stapsgewijze videoreeks om webpagina s te leren bouwen.'
  ),
  new M13Product(
    'Full Stack traject',
    'courses',
    99,
    'Een traject dat front end en back end combineert.'
  ),
  new M13Product(
    'Task Manager App',
    'software',
    15,
    'Eenvoudige tool om dagelijkse taken bij te houden.'
  ),
  new M13Product(
    'Budget Planner',
    'software',
    9.99,
    'Software om inkomsten en uitgaven overzichtelijk te beheren.'
  )
];

// Kleine helper om een nette categorie naam te tonen
function m13_categoryLabel(category) {
  switch (category) {
    case 'books':
      return 'Boek';
    case 'courses':
      return 'Cursus';
    case 'software':
      return 'Software';
    default:
      return category;
  }
}

// Helper voor badge klasse op basis van categorie
```

```
function m13_categoryBadgeClass(category) {
  switch (category) {
    case 'books':
      return 'bg-outline-secondary border text-muted';
    case 'courses':
      return 'bg-primary text-white';
    case 'software':
      return 'bg-info text-dark';
    default:
      return 'bg-light text-muted';
  }
}

// Producten tekenen
function m13_renderProducts(products, isFiltered) {
  const container = document.getElementById('m13_products');
  const status = document.getElementById('m13_status');

  if (!products || products.length === 0) {
    container.innerHTML = '';
    status.className = 'alert alert-warning mb-3';
    status.textContent = 'Geen producten gevonden met deze filter.';
    return;
  }

  // Als er gefilterd is, tonen we de hoeveelheid
  if (isFiltered) {
    status.className = 'alert alert-success mb-3';
    status.textContent = `Gevonden producten: ${products.length}`;
  } else {
    status.className = 'alert alert-secondary mb-3';
    status.textContent = 'Er worden momenteel alle producten getoond.';
  }

  container.innerHTML = products
    .map(product => {
      const badgeClass = m13_categoryBadgeClass(product.category);
      const catLabel = m13_categoryLabel(product.category);

      return `
        <div class="col-md-6 col-lg-4">
          <div class="card h-100 shadow-sm">
            <div class="card-body d-flex flex-column">
              <div class="d-flex justify-content-between align-items-start
mb-2">
                <h5 class="card-title mb-0">${product.name}</h5>
                <span class="badge ${badgeClass}">${catLabel}</span>
              </div>
              <p class="card-text mb-2">${product.description}</p>
              <p class="mt-auto fw-bold">
                €${product.price.toFixed(2)}
              </p>
            </div>
          </div>
        </div>
      `;
```

```
    })
    .join('');
}

// Filter toepassen op basis van de input
function m13_applyFilter() {
  const catSelect = document.getElementById('m13_cat');
  const maxPriceInput = document.getElementById('m13_max');

  const selectedCategory = catSelect.value; // 'all', 'books', ...
  const maxPriceStr = maxPriceInput.value.trim();
  const maxPrice = maxPriceStr ? Number(maxPriceStr) : null;

  let filtered = m13_allProducts.slice();

  // Categorie filter
  if (selectedCategory !== 'all') {
    filtered = filtered.filter(p => p.category === selectedCategory);
  }

  // Prijs filter
  if (maxPrice !== null && !Number.isNaN(maxPrice) && maxPrice >= 0) {
    filtered = filtered.filter(p => p.price <= maxPrice);
  }

  const isFiltered =
    selectedCategory !== 'all' || (maxPrice !== null
&& !Number.isNaN(maxPriceStr));

  m13_renderProducts(filtered, isFiltered);
}

document.addEventListener('DOMContentLoaded', () => {
  // Initieel alle producten tonen
  m13_renderProducts(m13_allProducts, false);

  // Filter knop koppelen
  document.getElementById('m13_filter')?.addEventListener('click',
m13_applyFilter);

  // Optional: filter ook bij Enter in het prijsveld
  document.getElementById('m13_max')?.addEventListener('keyup', event => {
    if (event.key === 'Enter') {
      m13_applyFilter();
    }
  });
});
```

Oefening M14 — Rick & Morty Mini-Viewer (eerste 20 personages)

Opgave

In deze oefening bouw je een eenvoudige viewer die de eerste 20 personages uit de Rick & Morty API ophaalt en toont.

De student leert:

- werken met `fetch()` en `async/await`
- JSON verwerken
- loading state tonen
- foutmeldingen tonen
- DOM-kaarten genereren
- client-side filtering (op naam) zonder extra API-calls

Je gebruikt deze API:

`https://rickandmortyapi.com/api/character/?page=1`

Het resultaat wordt getoond in een kaartgrid van 20 personen.

Er is:

1. Een knop “Personages laden”
2. Een zoekveld “Filter op naam”
3. Een statusvak
4. Een grid met kaarten

Filtering gebeurt lokaal in JavaScript (dus geen API-parameters).

HTML (zoals in de cursusstijl)

```
<section class="container py-4" id="ex-m14">
  <div class="row justify-content-center">
    <div class="col-md-11 col-lg-9">
      <div class="card shadow-sm">

        <div class="card-header bg-info text-white">
          Oefening M14 — Rick & Morty Mini-Viewer
        </div>

        <div class="card-body">

          <button id="m14_load" class="btn btn-info w-100 mb-3">
            Personages laden
          </button>

          <div class="mb-3">
            <label for="m14_search" class="form-label">Filter op naam</label>
            <input
              id="m14_search"
              class="form-control"
              placeholder="Typ om te filteren…"
              disabled
            />
```

```
          </div>

          <div id="m14_status" class="alert alert-secondary mb-3"
role="status" aria-live="polite">
            Klik op "Personages laden" om te beginnen.
          </div>

          <div id="m14_cards" class="row g-3"></div>

        </div>
      </div>
    </div>
  </div>
</section>
```

JavaScript-oplossing

```
// ------------------------------------------------------
// Oefening M14 – Rick & Morty Mini-Viewer (clean version)
// ------------------------------------------------------

let m14_characters = [];
let m14_filtered = [];

function m14_setStatus(msg, type = "secondary") {
  const box = document.getElementById("m14_status");
  box.className = `alert alert-${type} mb-3`;
  box.textContent = msg;
}

function m14_renderCards(list) {
  const container = document.getElementById("m14_cards");

  if (!list || list.length === 0) {
    container.innerHTML = "";
    return;
  }

  container.innerHTML = list
    .map(ch => {
      return `
        <div class="col-md-6 col-lg-4">
          <div class="card h-100 shadow-sm">
            <img src="${ch.image}" class="card-img-top" alt="${ch.name}">
            <div class="card-body">
              <h5 class="card-title">${ch.name}</h5>

              <p class="card-text mb-1">
                <strong>Status:</strong> ${ch.status}
              </p>

              <p class="card-text mb-1">
                <strong>Soort:</strong> ${ch.species}
              </p>

              <p class="card-text mb-0">
```

```
                <strong>Locatie:</strong> ${ch.location.name}
              </p>
            </div>
          </div>
        </div>
      `;
    })
    .join("");
}

async function m14_loadCharacters() {
  const url = "https://rickandmortyapi.com/api/character/?page=1";

  m14_setStatus("Bezig met laden…", "warning");
  document.getElementById("m14_cards").innerHTML = "";
  document.getElementById("m14_search").disabled = true;

  try {
    const res = await fetch(url);
    if (!res.ok) throw new Error("HTTP fout");

    const data = await res.json();
    m14_characters = data.results;
    m14_filtered = [...m14_characters];

    m14_renderCards(m14_filtered);

    m14_setStatus("20 personages geladen.", "success");
    document.getElementById("m14_search").disabled = false;
  } catch (err) {
    m14_setStatus("Er ging iets mis bij het laden.", "danger");
  }
}

function m14_applyFilter() {
  const search =
document.getElementById("m14_search").value.trim().toLowerCase();

  if (!search) {
    m14_filtered = [...m14_characters];
    m14_setStatus("20 personages geladen.", "success");
    m14_renderCards(m14_filtered);
    return;
  }

  m14_filtered = m14_characters.filter(ch =>
    ch.name.toLowerCase().includes(search)
  );

  if (m14_filtered.length === 0) {
    m14_setStatus("Geen resultaten voor deze zoekopdracht.", "info");
  } else {
    m14_setStatus(`${m14_filtered.length} resultaten gevonden.`, "success");
  }
```

```
  m14_renderCards(m14_filtered);
}

document.addEventListener("DOMContentLoaded", () => {
  document.getElementById("m14_load")?.addEventListener("click",
m14_loadCharacters);
  document.getElementById("m14_search")?.addEventListener("input",
m14_applyFilter);
});
```