

Oefening M1 — Som van twee getallen .....	2
Section .....	2
Oefening M2 — Celsius naar Kelvin .....	5
Section .....	6
Oefening M3 — BMI-berekening .....	7
Section .....	7
Oefening M4 — BMI met categorie en kleur .....	10
Section .....	10
Oefening M5 — Prijs berekenen na korting .....	14
Section .....	14
Dit zijn uitbreidingsoefeningen. Kan je deze? Top, indien niet, niet erg momenteel. ....	16
Oefening M6 — Prijs na korting (uitbreiding met typekeuze, coupon en minimumprijs) .....	16
Section .....	17
Oefening M7 - Notenkraaker — Rekenmachine in de stijl van Windows .....	20
Tips voor implementatie .....	21
1. Werken met tekst en getallen .....	21
2. Afrondingen en correct weergeven .....	22
3. Basisbewerkingen uitvoeren .....	22
4. Werken met functies .....	22
5. Geheugenfunctie .....	22
6. Speciale berekeningen .....	22
7. Manipulatie van tekstinvoer .....	23
8. Event-afhandeling .....	23
9. Beheren van meerdere stappen .....	23
10. Extra oefeningstips .....	23
Section .....	23

## Oefening M1 — Som van twee getallen

Je leest de twee ingevoerde waarden uit de velden en verwijdert eventuele overtollige spaties aan het begin en einde. Daarna beoordeel je of beide invoervelden effectief een getal voorstellen. Wanneer één van de velden leeg is of geen geldig getal bevat, geef je een waarschuwing weer: je verandert de stijl van het meldingsvak naar de gele variant en toont exact de tekst “Vul twee geldige getallen in”. Wanneer beide waarden wel geldig zijn, bereken je de som met een aparte rekenfunctie die uitsluitend het resultaat teruggeeft. Vervolgens zet je het meldingsvak om naar de groene variant en toon je het resultaat in de vorm “a + b = uitkomst”, waarbij je de actuele waarden en de berekende uitkomst duidelijk in dezelfde regel weergeeft. Tot slot zorg je ervoor dat een klik op de knop de volledige verwerking uitvoert, zodat de gebruiker telkens direct feedback ziet.

### Section

```
<section class="container py-4" id="ex-m1">
  <div class="row justify-content-center">
    <div class="col-md-6 col-lg-5">
      <div class="card shadow-sm">
        <div class="card-header bg-primary text-white">Oefening M1 – Som van
twee getallen</div>
        <div class="card-body">
          <label for="m1_a" class="form-label">Getal A</label>
          <input id="m1_a" type="number" step="any" class="form-control mb-2"
/>
          <label for="m1_b" class="form-label">Getal B</label>
          <input id="m1_b" type="number" step="any" class="form-control mb-2"
/>
          <button id="m1_btn" class="btn btn-primary w-100 mb-
3">Bereken</button>
          <div id="m1_out" class="alert alert-secondary mb-0" role="status"
aria-live="polite">
            Resultaat verschijnt hier...
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

### Oefening M1 — Som van twee getallen

Getal A

Getal B

Bereken

$2 + 3 = 5$

### Oefening M1 — Som van twee getallen

Getal A

Getal B

Bereken

Vul twee geldige getallen in

## Oefening M1 — Som van twee getallen

Getal A

2

Getal B

3

Bereken

$$2 + 3 = 5$$

## Oefening M2 — Celsius naar Kelvin

In deze oefening lees je een temperatuur in graden Celsius in en zet je die om naar Kelvin. De omzetting gebeurt met de formule:

$$K = C + 273,15$$

Eerst wordt de ingevoerde waarde opgehaald en wordt gecontroleerd of er effectief iets is ingevuld. Wanneer het veld leeg is of de invoer geen geldig getal voorstelt, verschijnt een waarschuwing: het meldingsvak krijgt de gele opmaak en toont exact de tekst “Vul een geldig getal in”.

Wanneer de waarde wel correct is, bereken je de temperatuur in Kelvin met behulp van de formule hierboven. De berekening gebeurt in een afzonderlijke functie die enkel de uitkomst bepaalt, zodat de rekenlogica losstaat van de weergave. Daarna geef je het resultaat weer in een groene melding in de vorm “C° = K K”, waarbij beide getallen overeenstemmen met de ingevoerde Celsiuswaarde en de berekende Kelvinwaarde.

Tot slot zorg je ervoor dat deze handeling wordt uitgevoerd wanneer de gebruiker op de knop “Zet om” klikt, zodat de invoer direct verwerkt en het resultaat onmiddellijk zichtbaar wordt.

Oefening M2 — Celsius naar Kelvin

Temperatuur (°C)

bijv. 25

Zet om

Resultaat verschijnt hier...

Oefening M2 — Celsius naar Kelvin

Temperatuur (°C)

25

Zet om

25°C = 298.15 K

## Section

```
<section class="container py-4" id="ex-m2">
  <div class="row justify-content-center">
    <div class="col-md-6 col-lg-5">
      <div class="card shadow-sm">
        <div class="card-header bg-primary text-white">Oefening M2 – Celsius
naar Kelvin</div>
        <div class="card-body">
          <label for="m2_c" class="form-label">Temperatuur (°C)</label>
          <input id="m2_c" type="number" step="any" class="form-control mb-2"
placeholder="bijv. 25" />
          <button id="m2_btn" class="btn btn-primary w-100 mb-3">Zet
om</button>
          <div id="m2_out" class="alert alert-secondary mb-0" role="status"
aria-live="polite">
            Resultaat verschijnt hier...
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

## Oefening M3 — BMI-berekening

In deze oefening bereken je de **Body Mass Index (BMI)** op basis van een ingevoerd gewicht in kilogram en een ingevoerde lengte in centimeter. Eerst lees je beide waarden in en controleer je of er werkelijk iets is ingevuld en of de waarden betekenisvol zijn; een lege invoer of een lengte die niet boven nul ligt moet worden afgewezen. Als de invoer niet deugt, laat je een duidelijke waarschuwing zien in het meldingsvak met de tekst “**Vul geldige waarden in**” en geef je geen resultaat.

Wanneer de gegevens wel kloppen, zet je de lengte om van centimeter naar meter volgens:

$$m = cm / 100$$

Gebruik vervolgens de BMI-formule:

$$BMI = \text{gewicht} / (m \times m)$$

Je voert de berekening uit in een afzonderlijke rekenfunctie die alleen de uitkomst bepaalt. Daarna presenteer je het resultaat in het meldingsvak met een groene aanduiding en schrijf je het getal met twee decimalen op het scherm in de vorm “**BMI = X.XX**”. Tot slot zorg je ervoor dat deze volledige verwerking gebeurt wanneer de gebruiker op de knop “**Bereken BMI**” klikt, zodat de controle, de berekening en de weergave netjes in één handeling verlopen.

### Section

```
<section class="container py-4" id="ex-m3">
  <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">Oefening M3 – BMI-
berekening</div>
        <div class="card-body">
          <label for="bmi_w" class="form-label">Gewicht (kg)</label>
          <input id="bmi_w" type="number" step="0.1" class="form-control mb-
2" placeholder="bijv. 72.5" />
          <label for="bmi_h" class="form-label">Lengte (cm)</label>
          <input id="bmi_h" type="number" step="1" class="form-control mb-2"
placeholder="bijv. 175" />
          <button id="bmi_btn" class="btn btn-secondary w-100 mb-3">Bereken
BMI</button>
          <div id="bmi_out" class="alert alert-secondary mb-0" role="status"
aria-live="polite">
            Resultaat verschijnt hier...
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

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

### Oefening M3 — BMI-berekening

Gewicht (kg)

bijv. 72.5

Lengte (cm)

bijv. 175

Bereken BMI

Resultaat verschijnt hier...

### Oefening M3 — BMI-berekening

Gewicht (kg)

bijv. 72.5

Lengte (cm)

bijv. 175

Bereken BMI

Vul geldige waarden in



### Oefening M3 — BMI-berekening

Gewicht (kg)

82

Lengte (cm)

178

Bereken BMI

BMI = 25.88

## Oefening M4 — BMI met categorie en kleur

### Section

In deze uitbreiding bereken je opnieuw de lichaamsmassawaarde op basis van een ingevoerd gewicht in kilogram en een ingevoerde lengte in centimeter. Je controleert eerst of beide velden effectief ingevuld zijn, of de waarden betekenisvol zijn, en of de lengte groter is dan nul. Als de invoer niet klopt, toon je een duidelijke waarschuwing in de meldingsbox met de tekst “Vul geldige waarden in”.

Wanneer de invoer wel correct is, zet je de lengte om van centimeter naar meter met  $m = cm / 100$ , en bereken je de waarde met  $BMI = gewicht / (m \times m)$ . Na de berekening beoordeel je in welke categorie de uitkomst valt aan de hand van gangbare drempels: onder **18,5** is er sprake van **ondergewicht**, vanaf **18,5** tot en met **24,9** geldt **gezond gewicht**, van **25,0** tot en met **29,9** geldt **overgewicht**, en vanaf **30,0** spreek je van **obesitas**. Je past de kleur van de meldingsbox aan de categorie aan, zodat de gebruiker dit visueel herkent: een informatieve blauwe tint voor ondergewicht, een groene tint voor gezond gewicht, een gele tint voor overgewicht en een rode tint voor obesitas. Vervolgens toon je in dezelfde meldingsbox zowel de berekende waarde afgerond op twee decimalen als het bijhorende label, in de vorm “BMI = X.XX — Categorie: ...”, zodat de uitkomst in één oogopslag duidelijk is. Tot slot zorg je ervoor dat het volledige proces wordt uitgevoerd wanneer de gebruiker op de knop “Bereken BMI en categorie” klikt.

### Oefening M4 — BMI met categorie

Gewicht (kg)

Lengte (cm)

Bereken BMI en categorie

BMI = 17.36 — Categorie: Ondergewicht

Categorieën: Ondergewicht, Gezond gewicht, Overgewicht, Obesitas.

#### Oefening M4 — BMI met categorie

Gewicht (kg)

75

Lengte (cm)

178

Bereken BMI en categorie

BMI = 23.67 — Categorie: Gezond gewicht

Categorieën: Ondergewicht, Gezond gewicht, Overgewicht, Obesitas.

#### Oefening M4 — BMI met categorie

Gewicht (kg)

82

Lengte (cm)

178

Bereken BMI en categorie

BMI = 25.88 — Categorie: Overgewicht

Categorieën: Ondergewicht, Gezond gewicht, Overgewicht, Obesitas.

## Oefening M4 — BMI met categorie

Gewicht (kg)

Lengte (cm)

Bereken BMI en categorie

BMI = 37.87 — Categorie: Obesitas

Categorieën: Ondergewicht, Gezond gewicht, Overgewicht, Obesitas.

```
<section class="container py-4" id="ex-m4">
  <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">Oefening M4 — BMI
met categorie</div>
        <div class="card-body">
          <label for="bmi2_w" class="form-label">Gewicht (kg)</label>
          <input id="bmi2_w" type="number" step="0.1" class="form-control mb-
2" placeholder="bijv. 72.5" />
          <label for="bmi2_h" class="form-label">Lengte (cm)</label>
          <input id="bmi2_h" type="number" step="1" class="form-control mb-2"
placeholder="bijv. 175" />

          <button id="bmi2_btn" class="btn btn-secondary w-100 mb-3">Bereken
BMI en categorie</button>

          <!-- start als neutrale box; JS vervangt de klasse op basis van de
categorie -->
          <div id="bmi2_out" class="alert alert-secondary mb-0" role="status"
aria-live="polite">
            Resultaat verschijnt hier...
          </div>

          <div class="form-text mt-2">
            Categorieën: Ondergewicht, Gezond gewicht, Overgewicht, Obesitas.
```

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

## Oefening M5 — Prijs berekenen na korting

### Section

In deze oefening bereken je de nieuwe prijs van een product na toepassing van een korting in procent. De gebruiker voert twee waarden in: de oorspronkelijke prijs in euro en het kortingspercentage. Eerst lees je beide waarden in en controleer je of ze geldig zijn. Een lege invoer, een niet-numerieke waarde of een percentage dat kleiner is dan nul of groter dan honderd mag niet aanvaard worden. In dat geval geef je een waarschuwing weer in een gele meldingsbox met de tekst **“Vul een geldige prijs en korting in (0–100%)”**.

Wanneer de invoer wel correct is, bereken je de nieuwe prijs met de formule:

$$\text{Nieuwe prijs} = \text{prijs} \times (1 - \text{korting} / 100)$$

De berekening gebeurt in een aparte functie die enkel het resultaat teruggeeft. Daarna rond je de uitkomst af op twee decimalen en toon je ze in een groene meldingsbox. In de tekst vermeld je zowel het kortingspercentage als het eindbedrag, bijvoorbeeld **“Nieuwe prijs: €85,00 (Korting: 15%)”**. Zo ziet de gebruiker meteen het effect van de korting. Tot slot zorg je ervoor dat deze volledige verwerking gebeurt wanneer er op de knop **“Bereken nieuwe prijs”** wordt geklikt, zodat het resultaat direct zichtbaar wordt.

Oefening M5 — Prijs berekenen na korting

Oorspronkelijke prijs (€)

100

Korting (%)

15

Bereken nieuwe prijs

Nieuwe prijs: €85.00 (Korting: 15%)

Oefening M5 — Prijs berekenen na korting

Oorspronkelijke prijs (€)

bijv. 100

Korting (%)

bijv. 15

Bereken nieuwe prijs

Vul een geldige prijs en korting in (0–100%)

```

<section class="container py-4" id="ex-m5">
  <div class="row justify-content-center">
    <div class="col-md-6 col-lg-5">
      <div class="card shadow-sm">
        <div class="card-header bg-primary text-white">Oefening M5 – Prijs
berekennen na korting</div>
        <div class="card-body">
          <label for="price" class="form-label">Oorspronkelijke prijs
(€)</label>
          <input id="price" type="number" step="0.01" class="form-control mb-
2" placeholder="bijv. 100" />
          <label for="discount" class="form-label">Korting (%)</label>
          <input id="discount" type="number" step="0.1" class="form-control
mb-2" placeholder="bijv. 15" />
          <button id="discount_btn" class="btn btn-primary w-100 mb-
3">Bereken nieuwe prijs</button>
          <div id="discount_out" class="alert alert-secondary mb-0"
role="status" aria-live="polite">
            Resultaat verschijnt hier...
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

```

Dit zijn uitbreidingsoefeningen.

Kan je deze? Top, indien niet, niet erg momenteel.

## Oefening M6 — Prijs na korting (uitbreiding met typekeuze, coupon en minimumprijs)

In deze uitbreiding wordt de nieuwe prijs op drie opeenvolgende stappen berekend. Eerst kiest de gebruiker op welke manier de korting wordt toegepast: als percentage of als vast bedrag. Wanneer de korting in procent wordt gekozen, wordt de nieuwe prijs bepaald met de relatie **nieuwe prijs = oorspronkelijke prijs × (1 – korting/100)**. Wanneer de korting als vast bedrag wordt gekozen, wordt de nieuwe prijs berekend als **nieuwe prijs = oorspronkelijke prijs – bedrag**. Voor beide gevallen controleer je vooraf of de ingevoerde waarden logisch zijn: de prijs mag niet negatief zijn, het kortingspercentage moet zich tussen nul en honderd bevinden en een vast bedrag mag niet groter zijn dan de oorspronkelijke prijs.

Vervolgens kan de gebruiker optioneel een coupon invullen. Als de coupon overeenkomt met de afgesproken code, wordt er bovenop de eerdere korting nog eens tien procent extra korting toegepast op de tussenprijs. Die extra stap gebeurt met dezelfde gedachtegang als bij een procentuele korting: **nieuwe prijs = tussenprijs × (1 – 10/100)**. Als er geen geldige coupon is ingevuld, blijft de tussenprijs ongewijzigd.

Daarna wordt een bodem ingesteld met een minimum eindbedrag. Als de berekende prijs lager uitvalt dan dit minimum, stel je de uitkomst gelijk aan het minimum en geef je duidelijk aan dat de bodem is toegepast. In de weergave toon je een beknopt overzicht met de oorspronkelijke prijs, de eerste korting, de eventuele extra korting via de coupon, de tussenprijs, of de bodem is toegepast, en tot slot de eindprijs. De melding krijgt een passende kleur: een waarschuwingskleur wanneer de minimumgrens in werking trad, een informatieve kleur wanneer er effectief geen vermindering optrad, en een succeskleur wanneer de korting zonder bodem werd toegepast. Zo ziet de gebruiker in één oogopslag zowel de berekening als de context van het resultaat.



## Oefening M6 — Prijs na korting (uitbreiding)

Oorspronkelijke prijs (€)

Kortingswijze

Vast bedrag (€)



Korting (waarde)

Coupon (optioneel)

Coupon **SAVE10** geeft extra 10% korting bovenop de gekozen korting.

Minimum eindprijs (€)

De eindprijs wordt nooit lager dan dit bedrag.

Bereken nieuwe prijs

Oorspronkelijke prijs:	€100.00
Korting (bedrag):	-€30.00
Extra coupon (10%):	-€7.00
Tussensprijs:	€63.00
Minimumprijs toegepast:	nee
Eindprijs:	€63.00

## Section

```
<section class="container py-4" id="ex-m6">
  <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">Oefening M6 – Prijs na
korting (uitbreiding)</div>
      <div class="card-body">
        <div class="row g-2">
          <div class="col-12">
            <label for="ex_price" class="form-label">Oorspronkelijke prijs
(€)</label>
            <input id="ex_price" type="number" step="0.01" class="form-
control" placeholder="bijv. 100" />
          </div>

          <div class="col-sm-6">
            <label for="ex_mode" class="form-label">Kortingswijze</label>
            <select id="ex_mode" class="form-select">
              <option value="pct" selected>Percentage (%)</option>
              <option value="eur">Vast bedrag (€)</option>
            </select>
          </div>
          <div class="col-sm-6">
            <label for="ex_disc" class="form-label">Korting
(waarde)</label>
            <input id="ex_disc" type="number" step="0.01" class="form-
control" placeholder="bijv. 15" />
          </div>

          <div class="col-12">
            <label for="ex_coupon" class="form-label">Coupon
(optioneel)</label>
            <input id="ex_coupon" class="form-control" placeholder="bijv.
SAVE10" />
            <div class="form-text">Coupon <strong>SAVE10</strong> geeft
extra 10% korting bovenop de gekozen korting.</div>
          </div>

          <div class="col-12">
            <label for="ex_floor" class="form-label">Minimum eindprijs
(€)</label>
            <input id="ex_floor" type="number" step="0.01" class="form-
control" value="1.00" />
            <div class="form-text">De eindprijs wordt nooit lager dan dit
bedrag.</div>
          </div>

          <button id="ex_btn" class="btn btn-primary w-100 my-3">Bereken
nieuwe prijs</button>

          <div id="ex_out" class="alert alert-secondary mb-0" role="status"

```

```
aria-live="polite">  
  Resultaat verschijnt hier...  
  </div>  
  </div>  
  </div>  
  </div>  
  </div>  
</section>
```

## Oefening M7 - Notenkraak — Rekenmachine in de stijl van Windows

In deze eindoefening bouw je een rekenmachine die eruitziet en aanvoelt zoals de standaard rekenmachine in Windows. De bovenste regel fungeert als weergave en toont altijd de huidige invoer of het laatst berekende resultaat. Daaronder staat een rij met geheugenknoppen en vervolgens het hoofdtoetsenblok met bewerkingen en cijfers.

De rekenmachine werkt met twee soorten waarden: de huidige invoer die je stap voor stap opbouwt door cijfers te kiezen, en een bewaarde waarde die nodig is om bewerkingen over meerdere stappen uit te voeren. Zodra je een bewerking kiest, wordt de op dat moment ingevoerde waarde vastgehouden als tussenresultaat en start je met een nieuwe invoer voor de volgende operand. Wanneer je nadien op het gelijktteken drukt, wordt de gekozen bewerking toegepast op het bewaarde getal en de huidige invoer en verschijnt het resultaat op de weergave. Daarna kan je verder rekenen met dat resultaat, of opnieuw beginnen door de invoer te vervangen.

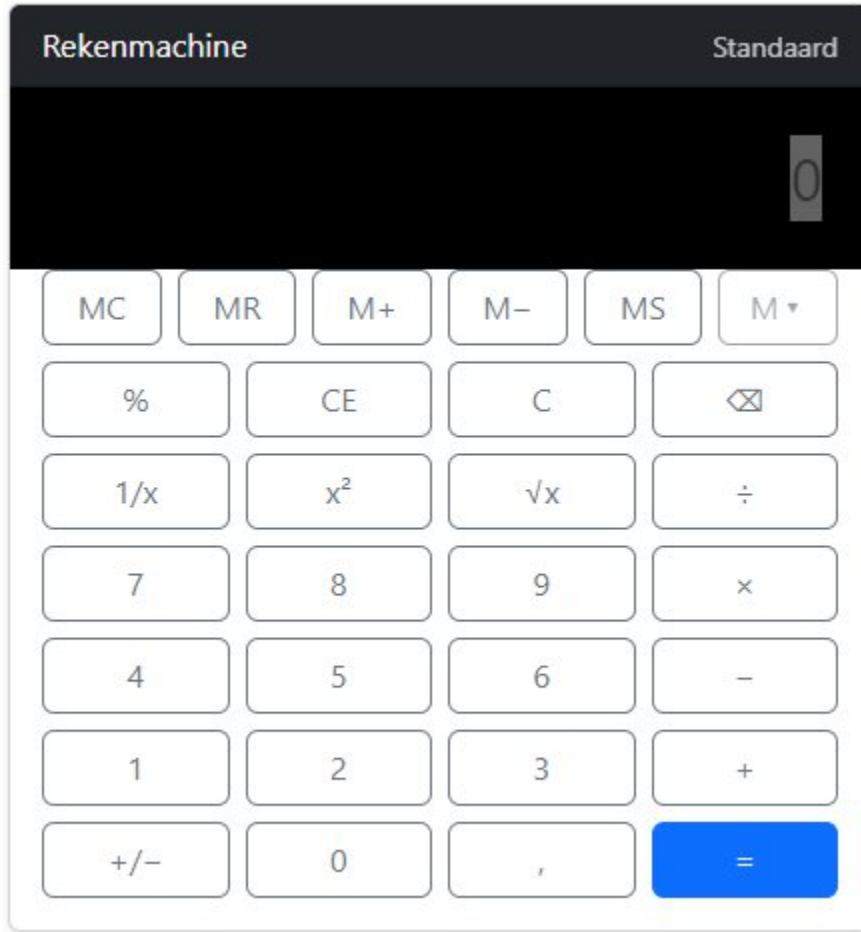
De knoppen voor wissen hebben ieder een specifiek doel. De knop die enkel de huidige invoer wist, zet alleen het getal waaraan je bezig bent terug naar nul. De knop die alles wist, zet zowel de invoer als het tussenresultaat en de gekozen bewerking terug zodat je volledig herstart. De terugtoets verwijdert het meest recent ingevoerde teken; wanneer er niets zinnigs meer overblijft, valt de invoer terug op nul. De knop voor het wisselen van teken verandert de huidige invoer van positief naar negatief of omgekeerd, met behoud van de rest van het getal. De procentknop geeft het percentage van de huidige invoer; deze oefening behandelt de eenvoudige betekenis waarbij een getal door honderd wordt gedeeld en onmiddellijk wordt getoond. De drie functietoetsen voor omkeren, kwadraat en vierkantswortel passen hun respectievelijke rekenregel toe op de huidige invoer; bij een onmogelijke wortel of een deling door nul hoort er geen getal uit en moet duidelijk zijn dat de uitkomst niet gedefinieerd is.

De rekenmachine heeft ook geheugentoetsen. Met wissen maak je het geheugen leeg. Met opslaan vervang je de geheugenwaarde door de huidige invoer. Met ophalen laad je de geheugenwaarde weer in het scherm zodat je er verder mee kunt rekenen. Met optellen aan geheugen verhoog je de geheugenwaarde met de huidige invoer, en met aftrekken van geheugen verlaag je die op vergelijkbare wijze. De indicator voor een geheugenlijst is aanwezig als verwijzing naar de Windows-interface, maar in deze oefening wordt er geen lijst met meerdere posities beheerd.

Bij het intoetsen van een komma wordt een decimaal getal opgebouwd; een tweede komma in hetzelfde getal is niet toegestaan. Invoer na het gelijktteken start automatisch een nieuw getal, zodat je na een berekening meteen aan de volgende kunt beginnen. Het resultaat wordt leesbaar gemaakt door overbodige nullen aan het einde te verwijderen wanneer dat gepast is, terwijl de betekenis van het getal behouden blijft.

Het geheel zorgt ervoor dat je probleemloos de gebruikelijke rekenacties kunt samenstellen: je voert getallen in, kiest een bewerking, bevestigt met het gelijktteken, en gebruikt tussendoor desgewenst geheugen, tekenwisseling, percentage of een van de

drie functies voor omkeren, kwadraat en vierkantswortel. Zo ontstaat een compacte, betrouwbare rekenmachine met herkenbare bediening en duidelijke feedback via de weergave en de kleuraanduiding van de knoppen.



## Tips voor implementatie

### 1. Werken met tekst en getallen

De invoer die uit tekstvelden of knoppen komt, is altijd een **string**. Om er berekeningen mee uit te voeren, moet je die eerst omzetten naar een **number**. Denk aan functies zoals:

- `parseFloat()` → zet een tekst om naar een getal met decimalen;
- `Number.isNaN()` → controleert of de omzetting wel een geldig getal opleverde.

Wanneer je een getal terug naar tekst omzet om het te tonen op het scherm, kan je gebruikmaken van:

- `toFixed(2)` om een vast aantal decimalen te tonen;
- `String()` of `toString()` om het terug als tekst te gebruiken;

- eventueel `replace(".", ",")` om de decimale punt aan te passen aan de Europese notatie.

## 2. Afrondingen en correct weergeven

Gebruik `Math.round()` of `toFixed()` om afrondfouten op te lossen. Bijvoorbeeld wanneer `0.1 + 0.2` resulteert in `0.30000000000000004`. Je mag het resultaat afronden tot een bepaald aantal cijfers om het netjes te tonen.

## 3. Basisbewerkingen uitvoeren

De rekenmachine gebruikt vier basisoperatoren:

- `+`, `-`, `*`, `/`

Je kunt de juiste operator herkennen via een `switch` of `if`-structuur. Zorg dat je ook controleert op deling door nul (bijvoorbeeld `if (b === 0) → foutmelding of “Niet gedefinieerd”`).

## 4. Werken met functies

Gebruik voor elke aparte taak een eigen functie:

- één functie die de **bewerking uitvoert**;
- één functie die **de weergave bijwerkt**;
- eventueel één functie die **de toestand beheert** (huidige waarde, vorige waarde, operator).

Zo blijft de code overzichtelijk en goed testbaar.

## 5. Geheugenfunctie

Je kunt een **variabele** gebruiken om het geheugen bij te houden (zoals `let mem = 0;`). Gebruik operatoren als `+=` of `-=` om waarden toe te voegen of af te trekken van dat geheugen. Bij het ophalen zet je die geheugenwaarde gewoon opnieuw in het display.

## 6. Speciale berekeningen

De Windows-rekenmachine bevat ook:

- $1/x \rightarrow$  omgekeerde: gebruik `1 / waarde`;
- $x^2 \rightarrow$  kwadraat: gebruik `waarde * waarde`;
- $\sqrt{x} \rightarrow$  vierkantswortel: gebruik `Math.sqrt(waarde)`.

Let bij de wortel op dat negatieve getallen geen reële uitkomst hebben — dat kun je testen met een voorwaarde en dan een melding tonen.

## 7. Manipulatie van tekstinput

Bij het verwijderen van het laatste teken of het omkeren van het teken:

- gebruik `slice()` om het laatste karakter te verwijderen;
- gebruik `startsWith("-")` om te controleren of een getal negatief is;
- gebruik string-samenvoeging om het teken om te keren `("-" + rest of string.slice(1))`.

## 8. Event-afhandeling

Elke knop moet een **click-event** hebben. Je kan alle knoppen selecteren met `querySelectorAll()` en in een lus een eventlistener toevoegen. Bij het opstarten gebruik je `DOMContentLoaded` om te zorgen dat alles pas actief wordt nadat de pagina volledig is geladen.

## 9. Beheren van meerdere stappen

De rekenmachine moet onthouden:

- wat de **vorige waarde** was;
- welke **operator** actief is;
- of de **laatste actie** een berekening (=) was.

Gebruik logische variabelen om deze toestand te volgen. Dat maakt het mogelijk om bewerkingen zoals  $7 + 3 \times 2 =$  correct af te handelen.

## 10. Extra oefeningstips

- Voeg een **toetsenbordondersteuning** toe met `keydown`-events (optioneel).
- Voeg een **error-status** toe wanneer er een ongeldige berekening gebeurt.
- Maak het display dynamisch breder of kleiner met `textContent.length`.
- Voeg een **reset**-knop toe die alles terugzet naar "0".

## Section

```
<section class="container py-4" id="ex-calc">
  <div class="row justify-content-center">
    <div class="col-md-6 col-lg-4">
      <div class="card shadow-sm">
        <div class="card-header bg-dark text-white d-flex justify-content-between align-items-center">
          <span>Rekenmachine</span>
          <span class="small opacity-75">Standaard</span>
        </div>

        <div class="card-body bg-black">
          <!-- Display -->
```

```

        <div class="border-0 bg-black text-white text-end p-2 fs-1"
id="calc_display" aria-live="polite">0</div>
    </div>

    <div class="card-body pt-0">
        <!-- Memory row -->
        <div class="d-grid gap-2 mb-2" style="grid-template-columns:
repeat(6, 1fr);">
            <button class="btn btn-outline-secondary"
id="calc_mc">MC</button>
            <button class="btn btn-outline-secondary"
id="calc_mr">MR</button>
            <button class="btn btn-outline-secondary"
id="calc_mplus">M+</button>
            <button class="btn btn-outline-secondary"
id="calc_mminus">M-</button>
            <button class="btn btn-outline-secondary"
id="calc_ms">MS</button>
            <button class="btn btn-outline-secondary" id="calc_mv"
disabled>M▼</button>
        </div>

        <!-- Main keypad -->
        <div class="d-grid gap-2" style="grid-template-columns: repeat(4,
1fr);">
            <button class="btn btn-outline-secondary" id="btn_pct">%</button>
            <button class="btn btn-outline-secondary" id="btn_ce">CE</button>
            <button class="btn btn-outline-secondary" id="btn_c">C</button>
            <button class="btn btn-outline-secondary"
id="btn_back">⊞</button>

            <button class="btn btn-outline-secondary"
id="btn_inv">1/x</button>
            <button class="btn btn-outline-secondary"
id="btn_pow2">x2</button>
            <button class="btn btn-outline-secondary"
id="btn_sqrt">√x</button>
            <button class="btn btn-outline-secondary" data-op="÷">÷</button>

            <button class="btn btn-outline-secondary" data-
digit="7">7</button>
            <button class="btn btn-outline-secondary" data-
digit="8">8</button>
            <button class="btn btn-outline-secondary" data-
digit="9">9</button>
            <button class="btn btn-outline-secondary" data-op="×">×</button>

            <button class="btn btn-outline-secondary" data-
digit="4">4</button>

```



```

        <button class="btn btn-outline-secondary" data-
digit="5">5</button>
        <button class="btn btn-outline-secondary" data-
digit="6">6</button>
        <button class="btn btn-outline-secondary" data-op="-">-</button>

        <button class="btn btn-outline-secondary" data-
digit="1">1</button>
        <button class="btn btn-outline-secondary" data-
digit="2">2</button>
        <button class="btn btn-outline-secondary" data-
digit="3">3</button>
        <button class="btn btn-outline-secondary" data-op="+">+</button>

        <button class="btn btn-outline-secondary"
id="btn_sign">+/-</button>
        <button class="btn btn-outline-secondary" data-
digit="0">0</button>
        <button class="btn btn-outline-secondary"
id="btn_comma">,</button>
        <button class="btn btn-primary" id="btn_eq">=</button>
    </div>
</div>
</div>
</div>
</div>
</div>
</section>

```