

Labo 17 – Verslag JavaScript | Deel 5

Opleidingsonderdeel: **Web Development 1**

Bachelor in de: **Toegepaste informatica**

Afstudeerrichting: **Applicatieontwikkeling**

Specialisatie: **Cybersecurity & infrastructure**

Leerling: **Jari Opsomer**

Docent: **Pim Debaere**

Academiejaar: **2025-2026**

2.2 Opdracht: trigrams

Een trigram is een opeenvolging van drie letters die na elkaar voorkomen in een tekst. Bijvoorbeeld, in het woord "**plezier**" zitten volgende trigrams vervat: **ple - lez - ezi - zie - ier**.

JS-code:

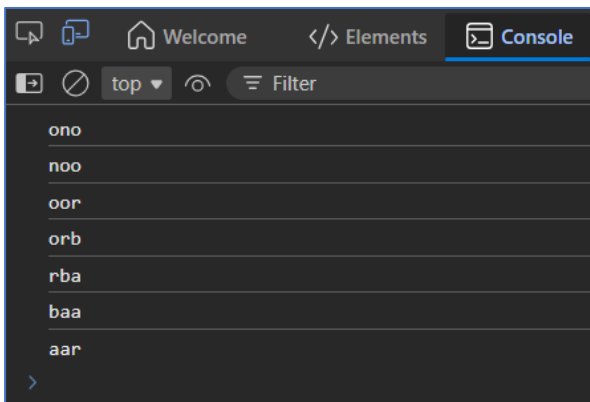
```
const setup = () => {
  const createTrigrams = (word) => {
    const trigrams = [];

    for (let i = 0; i < word.length - 2; i++) {
      trigrams.push(word.substring(i, i + 3));
    }

    return trigrams;
  };

  const trigrams = createTrigrams("onoorbaar");
  trigrams.forEach((trigram) => console.log(trigram));
}
window.addEventListener("load", setup);
```

Resultaat:



2.3 Opdracht: de en het

Gegeven is de tekst "Gisteren zat de jongen op de stoep en at de helft van de appel". Schrijf een programma dat telkens 'de' vervangt door 'het' en het resultaat op de console zet.

Merk op: maak géén gebruik van `replace` of `replaceAll` en ook niet van de `split` methode!

JS-code:

```
const setup = () => {
  let zin = "Gisteren zat de jongen op de stoep en at de helft van de appel";

  let huidigWoord = "";
  let nieuweZin = "";

  for (let i = 0; i < zin.length; i++) {
    let letter = zin[i]; // Haal de huidige letter op

    // Als we een spatie tegenkomen, weten we dat een woord compleet is
    if (letter === " ") {

      // Is het woord exact "de"?
      if (huidigWoord === "de") {
        nieuweZin += "het"; // Voeg "het" toe aan onze nieuwe zin
      } else {
        nieuweZin += huidigWoord; // Voeg het originele woord toe
      }

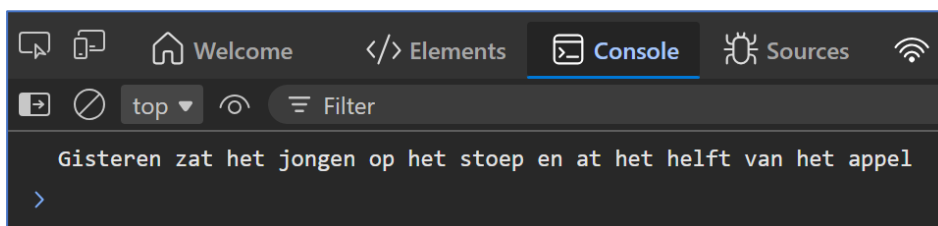
      // Vergeet niet de spatie zelf ook weer toe te voegen
      nieuweZin += " ";

      // Maak het 'huidigWoord' weer leeg zodat we aan het volgende woord kunnen beginnen
      huidigWoord = "";
    } else {
      // Als het geen spatie is, plakken we de letter vast aan het woord dat we aan het bouwen zijn
      huidigWoord += letter;
    }
  }

  // BELANGRIJK: Aan het einde van de zin staat geen spatie meer!
  if (huidigWoord === "de") {
    nieuweZin += "het";
  } else {
    nieuweZin += huidigWoord;
  }

  console.log(nieuweZin);
}
window.addEventListener("load", setup);
```

Resultaat:



- \n newline
- \' single quote
- \" double quote
- \\ backslash
- \u89ab voor unicode karakter 89ab

2.7 Opdracht: gemeenten

HTML-code:

```
<body>
  <button id="startKnop">Start</button>
  <label for="gemeente">Kies je gemeente</label>
  <select name="gemeente" id="gemeente"></select>

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

JS-code:

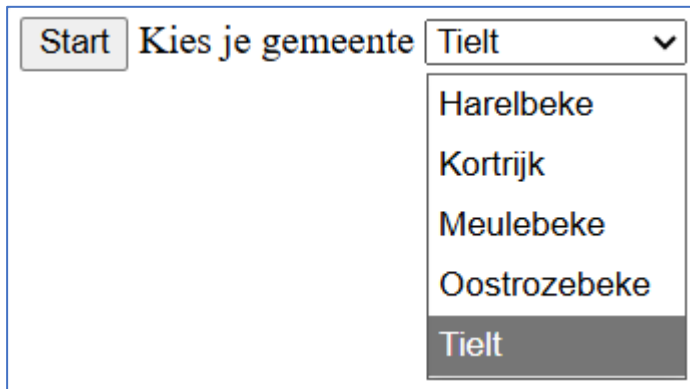
```
// Voeg gemeentes toe aan de dropdown op basis van gebruikersinput
const voegGemeentesToe = () => {
  const keuzelijst = document.getElementById("gemeente");
  const gemeentes = [];

  // Blijf vragen tot gebruiker "stop" invoert of annuleert
  let invoer;
  while ((invoer = prompt("Voeg een gemeente toe")) && invoer !== "stop")
  {
    gemeentes.push(invoer);
  }

  // Sorteert alfabetisch en voeg toe aan keuzelijst
  gemeentes.sort().reverse().forEach((gemeente) => {
    const optie = new Option(gemeente, gemeente);
    keuzelijst.add(optie, 0);
  });
};

// Koppel knop aan functie na laden pagina
window.addEventListener("load", () => {
  document.getElementById("startKnop").addEventListener("click", voegGemeentesToe);
});
```

Resultaat:



The screenshot shows a web form with a 'Start' button and a label 'Kies je gemeente'. A dropdown menu is open, showing 'Tielt' as the selected option. The dropdown list contains the following options: Harelbeke, Kortrijk, Meulebeke, Oostrozebeke, and Tielt (highlighted).

2.9 Opdracht: formwaarden

HTML-code:

```
<body>
<form id="mijnFormulier">
  <div class="form-group">
    <label for="roker">Is roker:</label>
    <div><input type="checkbox" id="roker" name="roker"></div>
  </div>

  <div class="form-group">
    <p>Moedertaal:</p>
    <div class="radio-group">
      <input type="radio" id="nl" name="moedertaal" value="nl">
      <label for="nl">Nederlands</label>

      <input type="radio" id="fr" name="moedertaal" value="fr">
      <label for="fr">Frans</label>

      <input type="radio" id="en" name="moedertaal" value="en">
      <label for="en">Engels</label>
    </div>
  </div>

  <div class="form-group">
    <label for="buurland">Favoriete buurland:</label>
    <select id="buurland" name="buurland">
      <option value="nederland">Nederland</option>
      <option value="frankrijk">Frankrijk</option>
      <option value="duitsland">Duitsland</option>
    </select>
  </div>

  <div class="form-group">
    <label for="bestelling">Bestelling:</label>
    <select id="bestelling" name="bestelling" multiple size="6">
      <option value="aardappelen">aardappelen</option>
      <option value="brood">brood</option>
      <option value="melk">melk</option>
      <option value="biefstuk">biefstuk</option>
      <option value="chips">chips</option>
      <option value="krant">krant</option>
    </select>
  </div>

  <button type="button" id="toonResultaat">Toon resultaat</button>
</form>

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

CSS-code:

```
.form-group {
  display: grid;
  grid-template-columns: auto 85%;
  margin-bottom: 10px;
}

#buurland,
#bestelling {
  width: 200px;
}
```

JS-code:

```
const setup = () => {
  document.getElementById('toonResultaat').addEventListener('click', () => {

    const isRoker = document.getElementById('roker').checked;
    let moedertaal = "";
    const moedertaalOpties = document.getElementsByName('moedertaal');
    const selectedOptie = Array.from(moedertaalOpties).find(optie =>
optie.checked);
    moedertaal = selectedOptie ? selectedOptie.value : undefined;
    const buurland = document.getElementById('buurland').value;

    const bestellingSelect = document.getElementById('bestelling');
    const bestelling = Array.from(bestellingSelect.selectedOptions).map(option
=> option.value);

    console.log(isRoker ? 'is roker' : "is geen roker");
    console.log('Moedertaal is', moedertaal);
    console.log('Favoriete buurland is', buurland);

    if (bestelling.length > 0) {
      console.log('Bestellingen bestaan uit ', bestelling.join(", "));
    } else {
      console.log('Geen bestellingen gevonden.');
```

Resultaat:

The screenshot shows a web application interface on the left and a browser developer console on the right. The form contains the following elements:

- Is roker:** A checkbox that is currently unchecked.
- Moedertaal:** Radio buttons for 'Nederlands' (selected), 'Frans', and 'Engels'.
- Favoriete buurland:** A dropdown menu with 'Nederland' selected.
- Bestelling:** A list box with 'aardappelen', 'brood', 'melk', 'biefstuk', 'chips', and 'krant'. 'aardappelen' and 'biefstuk' are selected.
- Toon resultaat:** A button to trigger the JavaScript code.

The browser console on the right shows the following log messages:

```
is geen roker
Moedertaal is nl
Favoriete buurland is nederland
Bestellingen bestaan uit aardappelen, brood, biefstuk
```

2.10 Opdracht: formvalidatie

HTML-code:

```
<body>

<form id="personalForm">
  <div class="form-group">
    <label for="firstName">Voornaam:</label>
    <input type="text" id="firstName" name="firstName"
placeholder="John">
    <span class="error" id="firstName-error"></span>
  </div>

  <div class="form-group">
    <label for="lastName">Familienaam:</label>
    <input type="text" id="lastName" name="lastName" placeholder="Doe">
    <span class="error" id="lastName-error"></span>
  </div>

  <div class="form-group">
    <label for="birthDate">Geboortedatum:</label>
    <input type="text" id="birthDate" name="birthDate"
placeholder="jjjj-mm-dd">
    <span class="error" id="birthDate-error"></span>
  </div>

  <div class="form-group">
    <label for="email">Email:</label>
    <input type="text" id="email" name="email"
placeholder="John.Doe@example.com">
    <span class="error" id="email-error"></span>
  </div>

  <div class="form-group">
    <label for="children">Aantal kinderen:</label>
    <input type="text" id="children" name="children" placeholder="2">
    <span class="error" id="children-error"></span>
  </div>

  <button type="button" id="validateBtn">Valideer</button>
</form>

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

</body>
```

CSS-code:

```
.form-group {
  margin-bottom: 15px;
}

.error {
  color: red;
  margin-left: 10px;
}

.invalid {
  border: 2px solid red;
}
```

JS-code:

```
// Koppel de knop aan validateForm na het laden van de pagina
document.addEventListener('DOMContentLoaded', () => {
  document.getElementById('validateBtn').addEventListener('click',
validateForm);
});

// Markeer een veld als ongeldig met een foutboodschap
const showError = (element, message) => {
  element.classList.add('invalid');
  document.getElementById(`${element.id}-error`).textContent = message;
};

// Verwijder alle foutmeldingen en ongeldige stijlen
const resetValidation = () => {
  document.querySelectorAll('input').forEach(input => {
    input.classList.remove('invalid');
    document.getElementById(`${input.id}-error`).textContent = '';
  });
};

const validateFirstName = () => {
  const firstName = document.getElementById('firstName');
  if (firstName.value.length > 30) {
    showError(firstName, 'max. 30 karakters');
    return false;
  }
  return true;
};

const validateLastName = () => {
  const lastName = document.getElementById('lastName');
  if (lastName.value.length === 0) {
    showError(lastName, 'verplicht veld');
    return false;
  }
  if (lastName.value.length > 50) {
    showError(lastName, 'max 50 karakters');
    return false;
  }
  return true;
};

const validateBirthDate = () => {
  const birthDate = document.getElementById('birthDate');
```

```

    if (birthDate.value.length === 0) {
      showError(birthDate, 'verplicht veld');
      return false;
    }
    if (!/^\d{4}-\d{2}-\d{2}$/.test(birthDate.value)) {
      showError(birthDate, 'formaat is niet jjjj-mm-dd');
      return false;
    }
    return true;
  };

const validateEmail = () => {
  const email = document.getElementById('email');
  if (email.value.length === 0) {
    showError(email, 'verplicht veld');
    return false;
  }
  // Controleer: precies één @, met tekst voor én na het @-teken
  const parts = email.value.split('@');
  if (parts.length !== 2 || parts[0].length === 0 || parts[1].length ===
0) {
    showError(email, 'geen geldig email adres');
    return false;
  }
  return true;
};

const validateChildren = () => {
  const children = document.getElementById('children');
  const childrenValue = parseInt(children.value);
  if (isNaN(childrenValue) || childrenValue < 0) {
    showError(children, 'is geen positief getal');
    return false;
  }
  if (childrenValue >= 99) {
    showError(children, 'is te vruchtbaar');
    return false;
  }
  return true;
};

// Valideer alle velden en toon resultaat
const validateForm = () => {
  resetValidation();

  const firstNameValid = validateFirstName();
  const lastNameValid = validateLastName();
  const birthDateValid = validateBirthDate();
  const emailValid = validateEmail();
  const childrenValid = validateChildren();

  if (firstNameValid && lastNameValid && birthDateValid && emailValid &&
childrenValid) {
    alert('Proficiat!');
  }
};

```

Resultaat:

localhost:63342/webdev/labo_17/2-10_opdracht-formvalidatie/index.html?_ijt=hb1dea7ekhad00ku7m4ijvefrs&_ij_reload=RELOAD_ON_

Toledo Portal

localhost:63342 vertelt
Proficiat!
OK

Voornaam:

Familienaam:

Geboortedatum:

Email:

Aantal kinderen:

Voornaam: max. 30 karakters

Familienaam: max 50 karakters

Geboortedatum: formaat is niet jjjj-mm-dd

Email: geen geldig email adres

Aantal kinderen: is geen positief getal