

Samenvatting (nieuw)

JavaScript Samenvatting - Examenvoorbereiding

Inhoudsopgave

1. [Basisconcepten](#)
 2. [DOM Manipulatie](#)
 3. [Event Listeners](#)
 4. [Variabelen en Data Types](#)
 5. [Arrays](#)
 6. [Conditional Logic](#)
 7. [Loops](#)
 8. [String Operaties](#)
 9. [Math Operaties](#)
 10. [Form Handling](#)
 11. [Element Manipulatie](#)
-

Basisconcepten

Variabelen Declareren

- `let` : Gebruik voor variabelen die kunnen veranderen
- `const` : Gebruik voor constanten (kan niet opnieuw toegewezen worden)
- **Arrow Functions:** `const functieNaam = () => { ... }`

```
let naam = "Jan";           // kan veranderen
const stad = "Gent";        // kan NIET veranderen
let leeftijd = 25;
let isStudent = true;
```

Data Types

- **String:** Tekst tussen aanhalingstekens (`"tekst"`)
- **Number:** Getallen (`25` , `3.14`)
- **Boolean:** `true` of `false`
- **Array:** Lijst van waarden (`[]`)

DOM Manipulatie

Elementen Selecteren

querySelector() - Selecteert ÉÉN element

```
// Selecteer element met ID  
let btn = document.querySelector("#btnTest");  
  
// Selecteer element met class (eerste match)  
let para = document.querySelector(".para");
```

querySelectorAll() - Selecteert ALLE elementen

```
// Selecteer alle elementen met class  
let paras = document.querySelectorAll(".para");  
  
// Selecteer alle divs  
let divs = document.querySelectorAll("div");  
  
// Selecteer alle elementen met name attribuut  
let radios = document.querySelectorAll("[name='keuze']");
```

Belangrijk:

- `querySelector()` geeft 1 element terug (of `null`)
- `querySelectorAll()` geeft een NodeList terug (lijst van elementen)
- Gebruik `#` voor ID's, `.` voor classes

Element Eigenschappen Wijzigen

textContent - Tekst wijzigen

```
let output = document.querySelector("#output");  
output.textContent = "Nieuwe tekst";
```

innerHTML - HTML wijzigen

```
let ul = document.querySelector("#lstItems");  
ul.innerHTML = ul.innerHTML + '<li>Nieuw item</li>';
```

value - Input waarde lezen/schrijven

```
let input = document.querySelector("#txtInput");
let waarde = input.value;           // lezen
input.value = "Nieuwe waarde";    // schrijven
```

style - CSS styles wijzigen

```
let div = document.querySelector("#divKleur");
div.style.backgroundColor = "red";
div.style.display = "none";        // verberg
div.style.display = "block";       // toon
div.style.color = "blue";
```

classList - CSS classes beheren

```
let para = document.querySelector(".para");
para.classList.toggle("belangrijk"); // toggle class
para.classList.add("nieuw");        // voeg toe
para.classList.remove("oud");      // verwijder
```

src - Image source wijzigen

```
let img = document.querySelector("#imgFoto");
img.src = "nieuwe-afbeelding.jpg";
```

checked - Checkbox/Radio status

```
let checkbox = document.querySelector("#chkAkkoord");
if (checkbox.checked) {
    // checkbox is aangevinkt
}
```

remove() - Element verwijderen

```
let item = event.target;
item.remove();
```

Event Listeners

Basis Syntax

```
element.addEventListener('eventType', functieNaam);
```

Veelgebruikte Events

click - Klik event

```
let btn = document.querySelector("#btnKlik");
btn.addEventListener('click', handleClick);

const handleClick = () => {
    console.log("Geklikt!");
};
```

change - Wijziging event (voor select, checkbox, radio)

```
let select = document.querySelector("#selOptie");
select.addEventListener('change', handleSelect);
```

keyup - Toetsenbord event

```
let input = document.querySelector("#txtToets");
input.addEventListener('keyup', handleKeyPress);
```

focus en blur - Focus events

```
let input = document.querySelector("#txtFocus");
input.addEventListener('focus', handleFocus);
input.addEventListener('blur', handleBlur);
```

submit - Form submit event

```
let form = document.querySelector("#formSubmit");
form.addEventListener('submit', handleSubmit);

const handleSubmit = (e) => {
```

```
    e.preventDefault(); // voorkom standaard gedrag  
};
```

Event Object

```
const handleClick = () => {  
  let element = event.target; // het element dat geklikt is  
  console.log(element);  
};
```

Arrow Functions in Event Listeners

```
// Met parameters  
btn.addEventListener('click', () => veranderKleur('red'));  
  
// Zonder parameters  
btn.addEventListener('click', handleClick);
```

Variabelen en Data Types

Globale Variabelen

```
let teller = 0; // globale variabele (buiten functie)  
  
const verhoog = () => {  
  teller++; // kan gebruikt worden in functies  
};
```

Null Checken

Altijd checken of element bestaat!

```
let btn = document.querySelector("#btnTest");  
if (btn) {  
  // element bestaat, veilig om te gebruiken  
  btn.addEventListener('click', handleClick);  
}
```

Meerdere Null Checks

```
if (txtInput && txtOutput) {  
    // beide elementen bestaan  
}
```

Arrays

Array Declareren

```
let boodschappen = []; // lege array
```

Array Operaties

push() - Item toevoegen

```
boodschappen.push("Appels");  
boodschappen.push(txtInput.value);
```

Array Leegmaken

```
boodschappen = [];
```

Door Array Loopen

```
for (let i = 0; i < boodschappen.length; ++i) {  
    console.log(boodschappen[i]);  
}
```

Array Lengte

```
let aantal = boodschappen.length;
```

Conditional Logic

If-Else Statements

```
if (voorwaarde) {
    // code als waar
} else if (andereVoorwaarde) {
    // code als andere voorwaarde waar is
} else {
    // code als alles onwaar is
}
```

Voorbeelden

```
// Leeftijd checken
if (leeftijd >= 18) {
    output.textContent = "Volwassen";
} else if (leeftijd >= 13) {
    output.textContent = "Tiener";
} else {
    output.textContent = "Kind";
}

// String checken
if (email.trim() === "") {
    output.textContent = "Email is verplicht!";
    return; // stop functie
}

// Lengte checken
if (wachtwoord.length < 6) {
    output.textContent = "Te kort!";
}
```

isNaN() - Check of getal geldig is

```
let leeftijd = parseInt(txtInput.value);
if (isNaN(leeftijd)) {
    output.textContent = "Geef een geldig getal in!";
}
```

trim() - Spaties verwijderen

```
if (txtInput.value.trim() !== "") {
    // input is niet leeg
}
```

Loops

For Loop

```
for (let i = 0; i < items.length; ++i) {
    let item = items[i];
    // doe iets met item
}
```

Door DOM Elementen Loopen

```
let items = document.querySelectorAll(".item");
for (let i = 0; i < items.length; ++i) {
    let item = items[i];
    item.addEventListener('click', handleClick);
    item.style.color = "blue";
}
```

`++i` vs `i++`

- `++i` : Verhoog eerst, gebruik dan (pre-increment)
- `i++` : Gebruik eerst, verhoog dan (post-increment)
- In for loops maakt het meestal niet uit, maar `++i` is iets sneller

String Operaties

String Eigenschappen en Methodes

`length` - Lengte van string

```
let tekst = "Hallo";
let lengte = tekst.length; // 5
```

`toUpperCase()` - Hoofdletters

```
let tekst = "hallo";
let hoofdletters = tekst.toUpperCase(); // "HALLO"
```

toLowerCase() - Kleine letters

```
let tekst = "HALLO";
let kleine = tekst.toLowerCase(); // "hallo"
```

charAt() - Karakter op positie

```
let tekst = "Hallo";
let eerste = tekst.charAt(0); // "H"
```

trim() - Spaties verwijderen

```
let tekst = " hallo ";
let schoon = tekst.trim(); // "hallo"
```

Template Literals (Backticks)

```
let naam = "Jan";
let bericht = `Welkom, ${naam}!`; // "Welkom, Jan!"

// Meerdere regels
let html = `Lengte: ${tekst.length}<br>
Hoofdletters: ${tekst.toUpperCase()}`;
```

Math Operaties

Basis Rekenkunde

```
let som = getal1 + getal2;      // optellen
let verschil = getal1 - getal2; // aftrekken
let product = getal1 * getal2; // vermenigvuldigen
let quotient = getal1 / getal2; // delen
```

Getallen Converteren

parseInt() - Naar geheel getal

```
let leeftijd = parseInt(txtInput.value);
```

parseFloat() - Naar decimaal getal

```
let getal = parseFloat(txtInput.value);
```

toFixed() - Decimalen afronden

```
let quotient = 10 / 3;  
let afgerond = quotient.toFixed(2); // "3.33"
```

Increment/Decrement

```
teller++; // verhoog met 1  
teller--; // verlaag met 1
```

Form Handling

Input Types

Text Input

```
let input = document.querySelector("#txtNaam");  
let waarde = input.value;
```

Checkbox

```
let checkbox = document.querySelector("#chkAkkoord");  
if (checkbox.checked) {  
    // is aangevinkt  
}
```

Radio Buttons

```
let radios = document.querySelectorAll("[name='keuze']");  
for (let i = 0; i < radios.length; ++i) {  
    if (radios[i].checked) {  
        let waarde = radios[i].value;  
        break;
```

```
    }  
}
```

Select/Dropdown

```
let select = document.querySelector("#selOptie");  
let waarde = select.value;
```

Form Validatie

```
const valideer = () => {  
    let email = txtEmail.value;  
    let wachtwoord = txtWachtwoord.value;  
  
    if (email.trim() === "") {  
        output.textContent = "Email is verplicht!";  
        return; // stop functie  
    }  
  
    if (wachtwoord.length < 6) {  
        output.textContent = "Wachtwoord te kort!";  
        return;  
    }  
  
    output.textContent = "Formulier is geldig!";  
};
```

Prevent Default

```
const handleSubmit = (e) => {  
    e.preventDefault(); // voorkom standaard form submit  
    // eigen code hier  
};
```

Element Manipulatie

Nieuwe Elementen Maken

```
const maakNieuwElement = () => {  
    let container = document.querySelector("#divContainer");
```

```
let nieuwDiv = document.createElement("div");
nieuwDiv.textContent = "Nieuw element!";
nieuwDiv.className = "nieuwElement";
container.appendChild(nieuwDiv);
};
```

Stappen:

1. `createElement("tag")` - Maak element
2. Stel eigenschappen in (`textContent`, `className`, etc.)
3. `appendChild()` - Voeg toe aan parent

Element Verwijderen

```
let item = event.target;
item.remove();
```

Display Toggle

```
if (div.style.display === "none") {
    div.style.display = "block";
} else {
    div.style.display = "none";
}
```

Belangrijke Patronen

Setup Pattern

```
const setupFunctie = () => {
    let btn = document.querySelector("#btnTest");
    if (btn) {
        btn.addEventListener('click', handleClick);
    }
};
```

Null Check Pattern

```
let element = document.querySelector("#id");
if (element) {
```

```
// veilig om te gebruiken  
}
```

Meerdere Null Checks

```
if (input && output) {  
    // beide bestaan  
}
```

Window Load Event

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

Waarom? Zorgt dat DOM volledig geladen is voordat JavaScript draait.

Veelgemaakte Fouten

1. **Vergeten null check:** Altijd checken of element bestaat!
 2. **Verkeerde selector:** `#` voor ID, `.` voor class
 3. **querySelector vs querySelectorAll:** Eerste geeft 1 element, tweede geeft lijst
 4. **textContent vs innerHTML:** `textContent` voor tekst, `innerHTML` voor HTML
 5. **value vs textContent:** `value` voor inputs, `textContent` voor andere elementen
 6. **Vergeten parseInt() / parseFloat():** Input values zijn altijd strings!
-

Handige Tips voor Examen

1. **Lees de vraag goed:** Wat wordt er precies gevraagd?
2. **Gebruik de juiste methode:**
 - 1 element? → `querySelector()`
 - Meerdere elementen? → `querySelectorAll()`
3. **Check altijd of element bestaat:** `if (element) { ... }`
4. **Gebruik console.log():** Voor debugging tijdens examen
5. **Let op syntax:**
 - Arrow functions: `const naam = () => { }`
 - Event listeners: `addEventListener('event', functie)`
 - Template literals: Backticks `'tekst ${variabele}'`

-
6. **Return statement:** Gebruik `return` om functie vroegtijdig te stoppen
 7. **Trim checken:** Gebruik `.trim()` om lege strings te checken
 8. **Type conversie:** Vergeet niet `parseInt()` of `parseFloat()` te gebruiken
-

Snelle Referentie

Wat	Hoe
Element selecteren (ID)	<code>document.querySelector("#id")</code>
Element selecteren (class)	<code>document.querySelectorAll(".class")</code>
Tekst wijzigen	<code>element.textContent = "tekst"</code>
HTML wijzigen	<code>element.innerHTML = "<p>tekst</p>"</code>
Input waarde lezen	<code>input.value</code>
Input waarde schrijven	<code>input.value = "waarde"</code>
CSS class toevoegen	<code>element.classList.add("class")</code>
CSS class verwijderen	<code>element.classList.remove("class")</code>
CSS class toggle	<code>element.classList.toggle("class")</code>
Style wijzigen	<code>element.style.eigenschap = "waarde"</code>
Event listener toevoegen	<code>element.addEventListener('event', functie)</code>
Array item toevoegen	<code>array.push(item)</code>
String lengte	<code>string.length</code>
Naar getal converteren	<code>parseInt(string) of parseFloat(string)</code>
Element verwijderen	<code>element.remove()</code>
Nieuw element maken	<code>document.createElement("tag")</code>
Element toevoegen	<code>parent.appendChild(child)</code>

Succes met je examen! 🎓