

Opdrachten sectie 3 – Javascript fundamentals (part 2, video 31 tot en met 38)

Vaardigheidsopdrachten

In onderstaande opdrachten ga je zelf functies schrijven en aanpassen. Hieronder een voorbeeld. De opdracht is: schrijf een functie die twee getallen deelt en het resultaat teruggeeft. De opdracht is op vier manieren opgelost.

```
function minus(numberOne, numberTwo){  
  return numberOne - numberTwo;  
}
```

```
function minus(numberOne, numberTwo){  
  const result = numberOne - numberTwo;  
  return result;  
}
```

```
const minus = function(numberOne, numberTwo){  
  return numberOne - numberTwo;  
};
```

```
const minus = (numberOne, numberTwo) => numberOne - numberTwo;
```

1. Schrijf een functie die in de console “Mijn eerste functie” logt.
 - a. Roep de functie aan.
2. Schrijf een functie die “Mijn eerste functie” teruggeeft.
 - a. Roep de functie aan.
 - b. Sla het resultaat op in een variabele.

- c. Log de waarde van de variabele naar de console.
3. Schrijf een functie die twee getallen optelt en het resultaat teruggeeft.
4. Schrijf een functie die twee getallen aftrekt en het resultaat teruggeeft.
5. Schrijf een functie die vier getallen vermenigvuldigt en het resultaat teruggeeft.
 - a. Roep de functie aan met de parameters 4, 5, 6, 7. Log het resultaat naar de console.
 - b. Roep de functie aan met de parameters 2, 3, 4, 5 en sla het resultaat op in een variabele.
 - c. Log de waarde van de variabele naar de console.
6. Schrijf een functie die uitzoekt welke van twee getallen het grootst is.
 - a. Geef het grootste getal terug.
 - b. Log het resultaat naar de console.
 - c. Test je functie uitvoerig met diverse testdata.
7. Schrijf een functie die bepaalt of de naam gelijk is aan 'Herder'. Het resultaat dat je teruggeeft is een *boolean*.
8. Schrijf een functie die uitzoekt of het meegegeven getal groter is dan 1000. Het resultaat dat je teruggeeft is een *boolean*.
9. Schrijf een functie die – wanneer de functie wordt aangeroepen – de zin “Ik ben een koe” naar de console schrijft.
 - a. Pas de functie aan zodat de er bij aanroep een dier kan worden meegegeven. Het meegegeven dier wordt vervolgens naar de console geschreven. Bijvoorbeeld: 'Ik ben een koe' of 'Ik ben een schaap'.
10. Wat verschijnt er in de console?

```
const nameCheck = (name) => {
  switch(name){
    case 'Jorn':
      return 'great';
    default:
      return 'okish';
  }
}

const money = money => money * 2;

const sentence = (name, salary) => {
  const wordOne = nameCheck(name);
  const wordTwo = money(salary);
  console.log(`The name is ${name} and I 'am ${wordOne}
    and my salary is ${wordTwo} and it just got doubled!`);
}

sentence('Frits', 3000);
```

11. Pas de
volgende code

aan, zodat er in totaal drie functies zijn.

```
function doMath(x, y){  
  const numberOne = x * 3;  
  const numberTwo = y * 4;  
  return numberOne + numberTwo;  
}
```

12. Schrijf een functie die op basis van je leeftijd en vooropleiding beslist of je mag studeren. Je mag studeren als je ouder bent dan 10 jaar en als je vooropleiding vwo, havo of vmbo is. De functie geeft een *boolean* terug.

Opdrachten sectie 3 – Javascript fundamentals (part 2, video 39 tot en met 41)

Vaardigheidsopdrachten

1. Schrijf de code die leidt tot onderstaande resultaat.

```
1  const arrayOne = ['Konijn', 'Koe', 'Vos', 'Das', 'Beer'];
```

CONSOLE ✕

Koe

2. Schrijf de code die leidt tot onderstaande resultaat.

```
1  const arrayOne = ['Konijn', 'Koe', 'Vos', 'Das', 'Beer'];
```

CONSOLE ✕

Beer

3. Schrijf de code die leidt tot onderstaande resultaat.

```
1  const arrayOne = ['Konijn', 'Koe', 'Vos', 'Das', 'Beer'];
```

CONSOLE ✕

Dier 1:Koe Dier 2: Vos

4. Schrijf de code die leidt tot onderstaande resultaat.

```
1  const arrayOne = ['Konijn', 'Koe', 'Vos', 'Das', 'Beer'];
```

CONSOLE ✕

Dier 1: Koe

Dier 2: Vos

Lengte array: 5

5. Maak een array met daarin vier elementen.
- a. Voeg een element aan de array toe (nadat je de array gemaakt hebt).
6. Verander het tweede element in de array naar 'Herder'.

Opdrachten sectie 3 – Javascript fundamentals (part 2, video 42 tot en met 45)

Vaardigheidsopdrachten

In de volgende acht opdrachten maak je gebruik van onderstaande object.

```
const objectOne = {  
  firstName: 'Henk',  
  lastName: 'de Vries',  
  age: 2021 - 1988,  
  job: 'teacher',  
  friends: ['Koe', 'Vos'],  
  getFullName: function(){  
    return `${this.firstName} ${this.lastName}`;  
  },  
};
```

1. Schrijf de code die leidt tot onderstaande resultaat.

CONSOLE ✕

My job is being a teacher

2. Schrijf de code die leidt tot onderstaande resultaat.

```
CONSOLE ✕  
  
My job is being a teacher and I am  
33 years old
```

3. Schrijf de code die leidt tot onderstaande resultaat.

```
CONSOLE ✕  
  
My friends are Koe and Vos
```

4. Schrijf de code die leidt tot onderstaande resultaat.

```
CONSOLE ✕  
  
8
```

5. Wat kan er beter in deze code? Verbeter de code.

```
const objectOne = {
  firstName: 'Piet',
  lastName: 'Janssens',
  job: 'Fighter',
  getJobAndName: function(name, job){
    return `My name is ${name} and I am a ${job}.`;
  }
}

console.log(objectOne['getJobAndName']('Piet Janssens', 'Fighter'));
```

6. Maak een object met drie “properties”.
- Log alle drie de “properties” naar de console.
 - Voeg een functie aan je object toe die de drie properties teruggeeft wanneer deze functie wordt aangeroepen.
 - Voeg aan je object een vierde “property” toe.
7. Wat is er semantisch niet correct aan onderstaande code?

```
const objectOne = {
  firstName: 'Piet',
  lastName: 'Janssens',
  job: 'Fighter',
  birthYear: 2000,
  getAge: function(name, job){
    if (!this.age) this.age = new Date().getFullYear() - this.birthYear;
    return this.age;
  }
}

console.log(objectOne.getAge());
```

8. Maak twee objecten, één voor jezelf en één voor één van de docenten. Geef beide objecten de “properties” naam, leeftijd, geboortjaar, baan en woonplaats.
- Schrijf een functie die de leeftijd van jezelf en van de docent naar de console logt.
 - Schrijf een functie die de naam van de oudste persoon naar de console logt.

- c. Schrijf een functie die de naam van de personen die in Almere wonen naar de console logt.

Opdrachten sectie 3 – Javascript fundamentals (part 2, video 46 tot en met 50)

Vaardigheidsopdrachten

1. Schrijf iedere stap die de genomen wordt in de code uit.

```
for (let i = 0; i <= 3; i++){  
    console.log(i);  
}
```

2. Schrijf de code die de getallen één tot en met tien naar de console logt. Gebruik een “for-loop”.
 - a. Doe dezelfde opdracht, maar gebruik nu een “while-loop”.
3. Schrijf de code die de getallen één tot en met honderd naar de console logt. Gebruik een “for-loop”.
 - a. Doe dezelfde opdracht, maar gebruik nu een “while-loop”.
4. Schrijf de code die de getallen één tot en met duizend naar de console logt. Gebruik een “for-loop”.
 - a. Doe dezelfde opdracht, maar gebruik nu een “while-loop”.
5. Schrijf de code die de getallen honderd tot en met duizend naar de console logt. Gebruik een “for-loop”.
 - a. Doe dezelfde opdracht, maar gebruik nu een “while-loop”.
6. Schrijf de code die de getallen honderd tot en met nul naar de console logt (dus 100, 99, 98, enzovoorts). Gebruik een “for-loop”.
 - a. Doe dezelfde opdracht, maar gebruik nu een “while-loop”.
7. Schrijf de code die even getallen van één tot en met duizend naar de console logt. Gebruik een “for-loop”.
 - a. Doe dezelfde opdracht, maar gebruik nu een “while-loop”.
8. Schrijf de code die alle honderdtallen van één tot en met tienduizend naar de console logt. Gebruik een “for-loop”.
 - a. Doe dezelfde opdracht, maar gebruik nu een “while-loop”.

9. Wat is het probleem in deze loop?

```
for (let i = 0; i >= 0; i++){  
  console.log(i);  
}
```

10. Wat is het probleem in deze loop?

```
for (let i = 50; i < 10; i++){  
  console.log(i);  
}
```

11. Wat is het probleem in deze loop?

```
for (let i = 0; i < 10; i = i + 0){  
  console.log(i);  
}
```

12. Wat is het probleem in deze loop?

```
for (let i = 0; i < 10; i++){  
  
}
```

13. Log elk element uit onderstaande array naar de console.

```
const arrayOne = ['Koe', 'Herder'];
```

14. Log elk element uit onderstaande array naar de console.

```
const arrayOne = ['Koe', 'Herder'];  
arrayOne.push('Vos');  
arrayOne.unshift('Das');
```

15. Maak een array met daarin honderd elementen.

- a. Log elk element uit je array naar de console.

16. Maak een array met de getallen één tot en met vijftien.

- a. Log elk element uit je array naar de console.
- b. Tel alle elementen uit je array bij elkaar op en log het resultaat naar de console.
- c. Als de som van alle elementen groter is dan honderd dan log je het resultaat niet naar de console, anders log je het dubbele van de som naar de console.

17. Log elk element uit onderstaande array naar de console.

```
const arrayOne = [  
  [1, 2, 3],  
  [4, 5, 6],  
  [7, 8, 9]  
]
```

18. Maak een array met de getallen vijf tot en met vijftien.

- Maak een nieuwe array die het dubbele van de getallen uit de originele array bevat (dus de waarden 10, 12, 14, 16, 18 en 20, 22, 24, 26, 28, 30). Los dit probleem op met een “for-loop”.
- Maak een nieuwe array die de even getallen uit de originele array bevat. Los dit probleem op met een “for-loop”.
- Maak een nieuwe array die de oneven getallen uit de originele array bevat. Los dit probleem op met een “for-loop”.
- Voeg aan de bestaande array het dubbele van elke bestaande waarde toevoegt (e.g. voor 5 wordt 10 toegevoegd, voor 6 wordt 12 toegevoegd).
- Log de lengte van de array uit opdracht 1 naar de console.

19. Bereken de leeftijd op basis van de onderstaande geboortejaren voor ieder element uit onderstaande array. De leeftijd log je naar de console. Je mag het huidige jaar hard coderen (maar dynamisch is beter!).

- Houd er nu rekening mee dat geboortejaren in de array, die niet van het datatype ‘number’ zijn in plaats van de leeftijd, de zin ‘Kan niet’ naar de console loggen.

```
const birthYears = [1988, 1982, 1988, 1988, 1922, 1990, 2000, 2020, 2033, 2001];
```

20. Log naar de consoles de tafels één tot en met tien (rekenen). Voordat iedere tafel start log je naar de console om welke tafel het gaat. Hieronder staat de tafel van één als voorbeeld.

```
CONSOLE X
Tafel van 1!
1
2
3
4
5
6
7
8
9
10
```

21. Maak een nieuwe array die alle combinaties van de elementen van twee arrays bevat. Bijvoorbeeld [1, 2, 3] en [4, 5] geeft als nieuwe array [14, 15, 24, 25, 34, 35]. Gebruik om je code te testen onderstaande arrays.

```
const arrayOne = [5, 2, 3, 45, 5, 7, 'Koe'];
const arrayTwo = [7, 'Dier', 'tjes', 'rt', 12, 5, 6];
```

22. Wat verschijnt er in de console? Verklaar het resultaat en pas de code aan zodat het wel werkt.

```
for (let part = 1; part <= 3; part++){
  console.log(`Let's start part ${part}!`);
  for (let subpart = 1; subpart <= 5; subpart++){
    console.log(`Working on subpart ${subpart}!`);
  }
  console.log('\n');
}
```

23. Wat verschijnt er in de console? Verklaar het resultaat en pas de code aan zodat het wel werkt.

```
for (let part = 1; part <=5; part++){  
  console.log(`Let's start part ${part}!`);  
  let subpart = 0;  
  while (subpart < 5){  
    console.log(subpart);  
    subpart++;  
    part++;  
  }  
}
```