

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

### Kennisopdrachten

1. Wat is “strict mode” in Javascript?
2. Waar in je script moet je “strict mode” activeren?
3. Waarom zou je “strict mode” in Javascript activeren?
4. Wat is een functie?
5. Wat is het verschil tussen een variabele en een functie?
6. Wat is de “function body”? Geef een codevoorbeeld bij je antw.
7. Kies de juiste beschrijving bij het nummer.
  - a. Function Call / Running the function / Invoking the function
  - b. Keyword
  - c. Naam van de functie
  - d. Parameter (van de functie)
  - e. Body (van de functie)

```
1      2      3      4
function JSC(inputOne, inputTwo){
  console.log(inputOne ** inputTwo)
  console.log(inputOne, inputTwo);
}      5

JSC(); 6
```

8. Wat verschijnt er in de console?



```
function smile(){  
  console.log("Cheese");  
}
```

9. Wat verschijnt er in de console?



```
function smile(){  
  console.log("Cheese");  
}  
smile();
```

10. Wat verschijnt er in de console?



```
function smile(){  
  console.log("Cheese");  
}  
smile();  
smile();
```

11. Wat verschijnt er in de console?



```
function love(numberOfHuges){  
  console.log(`Je krijgt ${numberOfHuges} knuffels van mij <3`);  
}  
  
love(3);
```

12. Wat verschijnt er in de console?



```
function love(numberOfHuges){  
  console.log(`Je krijgt ${numberOfHuges} knuffels van mij <3`);  
}  
  
love(12);
```

13. Wat verschijnt er in de console?



```
function love(numberOfHuges){  
  console.log(`Je krijgt ${numberOfHuges} ${numberOfHuges > 1 ? 'knuffels' : 'knuffel'} van mij <3`);  
}  
  
//love(12);  
love(1);
```

14. Wat verschijnt er in de console?



```
function love(numberOfHuges){  
  console.log(`Je krijgt ${numberOfHuges} ${numberOfHuges > 1 ? 'knuffels' : 'knuffel'} van mij <3`);  
}  
  
love(12);  
//love(1);
```

15. Wat verschijnt er in de console?



```
function love(numberOfHuges, name){  
  console.log(`Je krijgt ${numberOfHuges} ${numberOfHuges > 1 ? 'knuffels' : 'knuffel'} van mij <3`);  
  if(name === 'Jorn') console.log('extra knuffels!');  
}  
  
love(12, 'Jorn');
```

16. Wat verschijnt er in de console?



```
function love(numberOfHuges, name){  
  console.log(`Je krijgt ${numberOfHuges} ${numberOfHuges > 1 ? 'knuffels' : 'knuffel'} van mij <3`);  
  if(name === 'Jorn') console.log('extra knuffels!');  
}  
  
love(12, 'jorn');
```

17. Wat verschijnt er in de console?



```
function power(base, exponent){  
  return base ** exponent;  
}  
  
power(2, 2);
```

18. Wat verschijnt er in de console?



```
function power(base, exponent){  
  return base ** exponent;  
}  
  
const result = power(2, 2);  
console.log(result);
```

19. Wat verschijnt er in de console?



```
function power(base, exponent){  
  return base ** exponent;  
}  
console.log(power(2, 2));  
const result = power(2, 2);  
console.log(result);
```

20. Wat verschijnt er in de console?



```
function power(base, exponent){  
  return base ** exponent;  
}  
console.log(power(2, 2) + power(2, 2));
```

21. Wat verschijnt er in de console?



```
function power(base, exponent){  
  return base ** exponent;  
}  
console.log(power(2, 2) + power(2, 2) * power(2, 2));
```

22. Wanneer produceert een functie een resultaat?



23. Wat verschijnt er in de console?



```
function power(base, exponent){
  const result = base ** exponent;
}
console.log(power(2, 2));
```

24. Wat is het verschil tussen een “function declaration” en “function expression”? Geef in je uitleg een codevoorbeeld.



25. Wat is het verschil tussen een parameter en een argument? Geef in je uitleg een codevoorbeeld.



26. Wat is een “arrow function”?



27. Waarom het een “arrow function” en “arrow function”?



28. Wat is het verschil tussen een “arrow function” en een “function declaration”?



29. Wat verschijnt er in de console?



```
const minus = (numberOne, numberTwo) => numberOne - numberTwo;
console.log(minus(10, 5));
```

30. Wat verschijnt er in de console?



```
const add = (numberOne, numberTwo) => numberOne + numberTwo;
const minus = (numberOne, numberTwo) => numberOne - numberTwo;
console.log(minus(10, 5) + add(1, 2));
```

31. Wat verschijnt er in de console?



```
✓ const age = (birthYear, currentYear) => {  
  const age = currentYear - birthYear;  
  return age;  
}  
console.log(age(1988, 2022));
```

32. Wat verschijnt er in de console?



```
const sentence = (name, job, state) => {  
  const sentence = `${name} works as a ${job} and  
  ${name} is very ${state}`;  
  console.log(sentence);  
}  
sentence('Henk', 'Teacher', 'happy');
```

33. Wat verschijnt er in de console?



```
const sentence = (name, job, state) => {  
  const sentence = `${name} works as a ${job} and  
  ${name} is very ${state}`;  
  console.log(sentence);  
}  
sentence('Frits', 'Developer', 'happy');
```

34. Wat verschijnt er in de console?



```
✓ const sentence = (name, job, state) => {  
  const sentence = `${name} works as a ${job} and  
  ${name} is very ${state}`;  
  console.log(sentence);  
}  
sentence('Henk', 'Teacher')
```

35. Wat is het “Don’t repeat yourself”-principe? Geef een codevoorbeeld bij je antwoord.



36. Wat doet de “return”-keyword?



37. Waarom is deze functie zinloos?



```
function doMath(x, y){  
  return;  
  console.log(x + y);  
}
```



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

### Kennisopdrachten

1. Wat is een datastructuur?
2. Wat is een array?
3. Wat is het voordeel van een array?
4. Wat is een element in onderstaande array?

```
const friends = ['Piet', 'Frits', 'Kees'];
```

5. Welke code is niet geldig? Leg uit waarom.

```
a. const arrayOne = [];  
b. const arrayTwo;  
c. const arrayOne = 'Herder';  
d. const arrayOne = [1, 2, 3];  
e. const arrayOne[1] = 'Koedier';  
f. const arrayOne[2] = 'Moemoe';
```

6. Wat verschijnt er in de console?

```
const add = (x, y) => x + y;  
const numbers = [add(2, 3), add(2, 5)];  
|
```

7. Wat verschijnt er in de console?

```
const add = (x, y) => x + y;  
const numbers = [add(2, 3), add(2, 5)];  
console.log(numbers.length);  
numbers.push(add(4, 6));  
console.log(numbers.length);
```

8. Wat verschijnt er in de console?

```
const add = (x, y) => x + y;  
const numbers = [add(2, 3), add(2, 5)];  
console.log(numbers.length);  
numbers.push(add(4, 6));  
console.log(numbers.length ** numbers.length);
```

9. Wat verschijnt er in de console?

```
const add = (x, y) => x + y;
const numbers = [add(2, 3), add(2, 5)];
console.log(numbers.length);
console.log(numbers.push(add(4, 6)));
console.log(numbers.length ** numbers.length);
```

10. Wat verschijnt er in de console?

```
const arrayOne = ['Konijn', 'Koe', 'Vos', 'Das', 'Beer'];
console.log(arrayOne[6]);
```

11. Als een functie een “return value” heeft, moet je er dan iets mee doen? Leg je antwoord uit.

12. Wat verschijnt er in de console?

```
const add = [1, 2, 3];
add.unshift(4);
console.log(add);
```

13. Wat verschijnt er in de console?

```
const add = [1, 2, 3];
add.unshift(4);
console.log(add.length + add[1]);
```

14. Wat verschijnt er in de console?

```
const add = [1, 2, 3];  
add.unshift(4);  
add.pop();  
add.pop();  
console.log(add.length + add[1]);
```

15. Wat verschijnt er in de console?

```
const add = [1, 2, 3];  
add.unshift(4);  
const x = add.pop();  
const y = add.pop();  
console.log(add.length + add[1] + x + y);
```

16. Wat verschijnt er in de console?

```
const years = new Array(1991, 1992, 1993, friendOne);  
console.log(years);
```

17. Wat verschijnt er in de console?

```
const friendOne = "Henk";  
const friendTwo = "Frits";  
const friendThree = "Kees";  
  
const friends = [friendOne, friendTwo, friendThree];  
console.log(friends);
```

18. Wat verschijnt er in de console?

```
const add = [1, 2, 3];  
add.unshift(4);  
const x = add.pop();  
const y = add.pop();  
const z = add.shift();  
console.log(z);  
console.log(add.length + add[1] + x + y);
```

19. Wat verschijnt er in de console?

```
const add = [1, 2, 3];  
add.unshift(4);  
const x = add.pop();  
const y = add.pop();  
const z = add.shift();  
console.log(z);  
console.log(add.length + add[0] + x + y);
```

20. Wat verschijnt er in de console?

```
const add = [1, 2, 3];  
console.log(add.indexOf(1));  
add.unshift(4);  
console.log(add.indexOf(1));
```

21. Wat verschijnt er in de console?

```
const add = [1, 2, 3];  
console.log(add.includes(1));  
add.unshift(4);  
console.log(add.includes(1));
```

22. Wat verschijnt er in de console?

```
const add = [1, 2, 3];  
console.log(add.includes('1'));  
add.unshift(4);  
console.log(add.includes('1'));
```

23. Wat verschijnt er in de console?

```
const add = [1, 2, 3];  
console.log(add.indexOf(3) * add.indexOf(4));
```

24. Wat verschijnt er in de console?

```
const animals = ['Fox', 'Cow', 'Sheep', 'Pig', 'Dog', 'Cat'];  
if (animals.includes('Fox')){  
  console.log('Great!')  
  animals.unshift('Tiger');  
  console.log(animals[0]);  
}  
function animalLegalCheck(animal){  
  return animals.includes(animal);  
}
```

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

### Kennisopdrachten

1. Wat maakt een 'object' anders dan een 'array'? Geef in je uitleg een codevoorbeeld.
2. Wat is een 'key'?
3. Wat is een 'value'?
4. Wat is een 'key-value pair'?
5. Wat is een "property"?
6. Benoem de onderdelen a tot en met e.

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

7. Wanneer gebruik je de "bracket notation" en wanneer gebruik je de "dot notation" voor het verkrijgen van een "object property"?



8. Wat verschijnt er in de console?

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

9. Wat verschijnt er in de console?

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

10. Wat verschijnt er in de console?

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

11. Wat verschijnt er in de console?

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

12. Wat verschijnt er in de console?

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

13. Wat verschijnt er in de console?

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

14. Wat verschijnt er in de console?

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

15. Wat verschijnt er in de console?

```
const objectOne = {  
  firstName: 'Piet',  
  lastName: 'Janssens',  
  getFullName: function(){  
    |   return `${this.firstName} ${this.lastName}`  
  }  
}  
  
console.log(objectOne.getFullName);
```

16. Wat verschijnt er in de console?

```
const objectOne = {  
  firstName: 'Piet',  
  lastName: 'Janssens',  
  getFullName: function(){  
    return `${this.firstName} ${this.lastName}`  
  }  
}  
  
console.log(objectOne['getFullName']());
```

17. Wat verschijnt er in de console?

```
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());
```

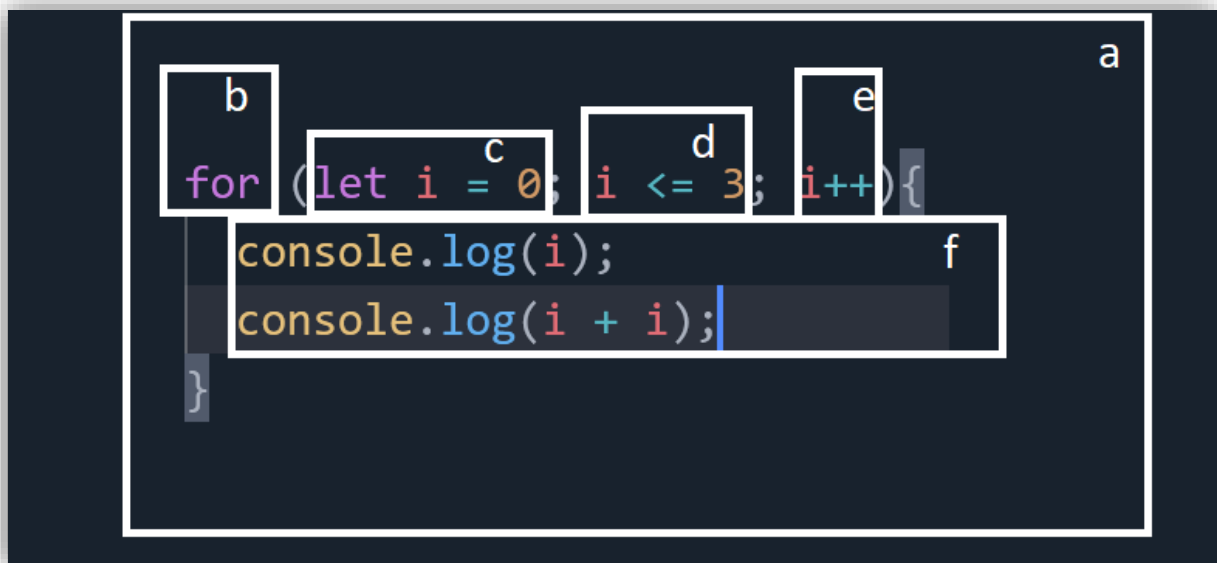
18. Wat verschijnt er in de console?

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

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

### Kennisopdrachten

1. Waar is een “for loop” bij uitstek geschikt voor? Geef een voorbeeld uit jouw leven waarvoor een “for loop” handig zou zijn.
2. Benoem de onderdelen a tot en met f. Beschrijf ook wat ieder onderdeel doet.



3. Wat verschijnt er in de console?

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

4. Wat verschijnt er in de console?

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

5. Wat verschijnt er in de console?

```
for (let i = 0; i < 5; i = i + 2){  
  console.log(i);  
}
```

6. Wat verschijnt er in de console?

```
for (let i = 5; i < 5; i = i + 2){  
  console.log(i);  
}
```



7. Wat verschijnt er in de console?

```
for (let i = 2; i < 4; i = i + 1){  
  console.log(i);  
}
```

8. Wat verschijnt er in de console?

```
for (let i = 2; i >= 0; i--){  
  console.log(i);  
}
```

9. Wat verschijnt er in de console?

```
for (let i = 0, j = 0; i < 5; i++, j = j + 2){  
  console.log(i + j);  
}
```

10. Wat verschijnt er in de console?

```
const arrayOne = ['Koe', 'Herder'];  
for (let index = 0; index < 2; index++){  
  console.log(arrayOne[index]);  
}
```

11. Wat verschijnt er in de console?

```
const arrayOne = ['Koe', 'Herder'];  
for (let index = 0; index < 5; index++){  
  console.log(arrayOne[index]);  
}
```

12. Geef vijf voorbeelden uit het echte leven waar een “for-loop” handig is.

13. Wat doet “continue” in een loop? Geef een codevoorbeeld bij je antwoord.

14. Wat doet “break” in een loop? Geef een codevoorbeeld bij je antwoord.

15. Wat is het verschil tussen een “for-loop” en een “while-loop”?

a. Wanneer gebruik je de “while-loop” en niet de “for-loop”?

16. Wat verschijnt er in de console?

```
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`);  
}
```

17. Wat verschijnt er in de console?

```
let i = 10;  
while(i >= 0){  
  console.log(`Ik tel tot ${i}`);  
  i--;  
}
```

18. Wat verschijnt er in de console?

```
let i = 10;
while(i >= 0){
  console.log(guus(i));
  i--;
}
function guus(i){
  return `Ik tel tot ${i}`;
}
```