

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

Kennisopdrachten

1. Wat is “strict mode” in Javascript?

Strict mode zorgt ervoor dat je in je code bugs makkelijker kan achterhalen en kan debuggen.

2. Waar in je script moet je “strict mode” activeren?

Altijd op eerste regel van je javascript document

3. Waarom zou je “strict mode” in Javascript activeren?

Zodat je de code makkelijker kan debuggen bij een bug

4. Wat is een functie?

Een functie is een code die je schrijft om steeds her te gebruiken waardoor je niet handmatig de zelfde code honderd keer hoeft te schrijven.

5. Wat is het verschil tussen een variabele en een functie?

Het verschil dat je in een variable kleine data kan opslaan en in een functie echt een functie meegeeft wat de code moet doen bijvoorbeeld een template literal maken met de aangegeven nummers

6. Wat is de “function body”? Geef een codevoorbeeld bij je uitleg.

Dat is de code die je meegeeft in de functie die uitgevoerd moet worden bij telkens het aanroepen van Functions

7. Kies de juiste beschrijving bij het nummer.
- a. Function Call / Running the function / Invoking the function **6**
 - b. Keyword **1**
 - c. Naam van de functie **2**
 - d. Parameter (van de functie) **3,4**
 - e. Body (van de functie) **5**

```
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?

Niks omdat de functie niet is aangeroepen

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

9. Wat verschijnt er in de console?

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

Cheese

10. Wat verschijnt er in de console?

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

Cheese

Cheese

11.

Je krijgt 3 knuffels van mij<3

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

12. Wat verschijnt er in de console?

Je krijgt 12 knuffels van mij<3

```
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);
```

Je krijgt 11 knuffels van mij<3

14. Wat verschijnt er in de console?

Je krijgt 12 knuffels van mij<3

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

15.

```
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');
```

Je krijgt 12 knuffels van mij<3

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');
```

Je krijgt 12 knuffels van mij<3

17. Wat verschijnt er in de console?

Niks

```
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);
```

4

19.

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

4

4

20. Wat verschijnt er in de console?

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

8

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));
```

32

22. Wanneer produceert een functie een resultaat?

Wanneer je de return keyword gebruikt in de functie of de functie aanroept.

23. Wat verschijnt er in de console?

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

4

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

Zie code (code file) De declarations kan je al aanroepen voordat je hem defined en bij de expression kan dat helaas niet. Bij expression sla je de functie op in een variable waardoor je het de functie geen naam hoeft te geven en dat wordt ook genoemd als anonymous function.

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

Een parameter is de waarde in de function en de argument is de waarde die meegeven om de waarde van function terug te krijgen

26. Wat is een “arrow function”?

Is een functie die meestal wordt gebruikt op één regel die dan heel kort is

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

Omdat je het schrijft met een arrow => dit lijkt op een arrow.

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

De verschil is dat de arrow function in een variable wordt opgeslagen en de declaration niet en in de arrow functie return automatisch de output

29. Wat verschijnt er in de console?

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

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));
```

14

31. Wat verschijnt er in de console?

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

34

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');
```

Henks works as a Teacher and Henk is very 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');
```

Frits works as a Developer and Frits is very 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')
```

Henk works as a Teacher and Henk is very Undefined.

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

Dat is dat je niet de zelfde code mag herhalen waardoor je een functie gemakkelijk kan gebruiken waardoor je de functie kan aanroepen dan in plaats van de code 20 keer te schrijven.

36. Wat doet de “return”-keyword?

Deze keyword geeft een waarde terug wat de gebruiker heeft ingevoerd.

37. Waarom is deze functie zinloos?

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

Omdat dit een variant is van herhalen van de code omdat de return keyword al iets terug gaat geven.

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

Kennisopdrachten

1. Wat is een datastructuur?

Is de structuur hoe een stuk informatie is opgeslagen

2. Wat is een array?

Een array is een soort variabele waar je meerdere informatie kan opslaan

3. Wat is het voordeel van een array?

Dan hoef je niet bijvoorbeeld 5 variabele aan te maken maar één array met 5 data 's er in

4. Wat is een element in onderstaande array?

“Piet”, “Frits”, “Kees”

```
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';
```

B. omdat het een waarde moet hebben. En bij a het juist is gedaan.

6. Wat verschijnt er in de console?

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

5,7

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?

3


```
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);
```

27

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);
```

27

10. Wat verschijnt er in de console?

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

Undefined

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

Dan moet je hem een waarde meegeven of waarde eraf halen.

12. Wat verschijnt er in de console?

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

4,1,2,3

13. Wat verschijnt er in de console?

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

6

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]);
```

3

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);
```

8

16. Wat verschijnt er in de console?

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

[1991,1992,1993,[]];

17.

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

Henk,Frits,Kees

17. 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);
```

1

5

19.

```
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);
```

4

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));
```

[0] [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));
```

True
True

22.

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

False

False

23. Wat verschijnt er in de console?

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

0

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(animals){
  return animals.includes(animals);
}
```

Great

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.

Je kan in een array bijvoorbeeld niet aangeven wat voor value het is en bij de objects wel.

firstName : Niloyan, dan weten dat de voornaam van mij wordt bedoeld.

2. Wat is een 'key'?

Dat is de naam wat je value geeft bijvoorbeeld firstName

3. Wat is een 'value'?

Dat is de waarde wat je wilt opslaan bijvoorbeeld : firstName : "Niloyan " <= value van firstName

4. Wat is een 'key-value pair'?

Dat is de key en de value samen

5. Wat is een "property"?

objectName.property // person.age

6. Benoem de onderdelen a tot en met e.

A objectNaam

B: key

C.value

D. Key-values


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

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

Je kan de dot notation gebruiken bij het verkrijgen van een data om iets te zien of om een zin te maken.

Bracket notation gebruiken we om een applicatie te voorzien van data van de object dan kan je hem makkelijker de data ophalen.

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']);
```

Henk, Henk

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']);
```

de Vries

10.

```
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);
```

Undefined

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']);
```

Undefined

12.

```
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);
```

Undefined

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);
```

True

14.

```
✓ 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());
```

Henk de Vries

15. Wat verschijnt er in de console?

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

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

Geen resultaat

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

21,21

15. 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());
```

21



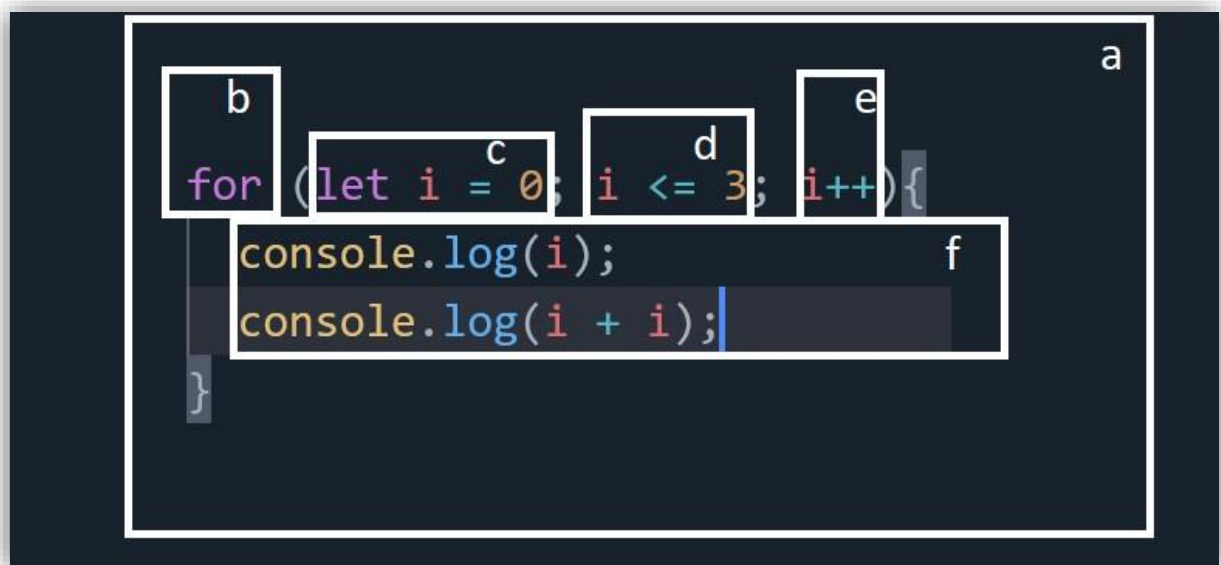
Opdrachten sectie 3 – Javascript fundamentals (part 2, video 46 tot en met 50)

Kennisopdrachten

1. Waar is een “for loop” bij uitstek geschikt voor? Geef een voorbeeld uit jouw leven waarvoor een “for loop” handig zou zijn.

Stel ik wil bijhouden hoe vaak ik naar de wc loop dan wil ik dit op een efficiënte manier bijhouden, dan gebruik ik for loop om het steeds bij te houden

2. Benoem de onderdelen a tot en met f. Beschrijf ook wat ieder onderdeel doet.



- a. Loop statement
- b. For woord
- c. De loop start
- d. Hoe vaak iets moet uitgevoerd moet worden
- e. Hoeveel stappen moet de loop nemen

3. Wat verschijnt er in de console?

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

1,2,3

4. Wat verschijnt er in de console?

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

0,0,1,2,2,4,3,6

5. Wat verschijnt er in de console?

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

0,2,4

6. Wat verschijnt er in de console?

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

0,2,4

7. Wat verschijnt er in de console?

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

2,3,4

8. Wat verschijnt er in de console?

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

2,1,0

9. Wat verschijnt er in de console?

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

0,3,6,9,12

10. Wat verschijnt er in de console?

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

Koe , Herder

11. Wat verschijnt er in de console?

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

Koe , Herder,undefined,undeifinde,.....

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

1. In de gym , 2. Bij houden hoe vaak je iets zegt 3. Om iets te tellen 4. Om een slider te maken, 5. Aantal coronabesmettingen

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

Dan gaat hij door met de code tot de statement komt

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

Dan stopt de loop helemaal

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

For loop kan je gebruiken om iets op te tellen of om bepaalde dingen te herhalen

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

While loop gebruik je tijdens een grote statement waar je iets niet hoeft te herhalen (dezelfde zin bijvoorbeeld) en de for loop kan je hem wel gebruiken om de zelfde zin te herhalen.

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--;  
}
```

Ik tel tot 10

Ik tel tot 9

.....

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}`;
}
```