

Leren programmeren - Extra opdrachten

Bart Duisters

Syntax schrijven

Opdracht 1

- Declareer twee variabelen, genaamd `nummerEen` en `nummerTwee`.
- Ken de waarde 3 (van het type "number") toe aan variabele `nummerEen`.
- Ken de waarde 5 (van het type "number") toe aan variabele `nummerTwee`.
- Initialiseer een variabele genaamd `uitkomst`, ken als waarde de uitkomst van de som van `nummerEen` en `nummerTwee` toe.
- Print de waarde van `uitkomst` uit.

Opdracht 2

- Definieer een functie genaamd `drieGetallenOptellen`.
- De functie accepteert drie parameters, genaamd `x`, `y` en `z`.
- Geef de optelsom van de drie parameters terug als resultaat.

Conditionele statements

Opdracht 1

Extra informatie:

```
const voornaam = "Bart".toLowerCase();
console.log(voornaam); // "bart"

const achternaam = "DuIsTeRs".toLowerCase();
console.log(achternaam); // "duisters"
```

- Maak een functie genaamd `kleineLetters`.
- De functie accepteert één parameter, genaamd `tekst`.
- Als de parameter bestaat uit alleen kleine letters, geef `true` terug.
- Als de parameter niet bestaat uit alleen kleine letters, geef `false` terug.

Verwacht resultaat:

```
console.log(kleinLetters("bart")); // true
console.log(kleinLetters("Bart")); // false
console.log(kleinLetters("BaRt")); // false
```

Opdracht 2

- Definieer een functie genaamd `uitlegBijTypeVanParameter`.
- De functie accepteert één parameter, genaamd `parameter`.
- Controleer het type van de parameter, indien het type:
 - "boolean" is, dan print uit "De mogelijke waarden zijn: false en true"
 - "number" is, dan print uit "Mogelijke waarden zijn: 1, 69, 3.141592"
 - "string" is, dan print uit "Mogelijke waarden zijn: 'a', 'Bart', '3'"
 - "object" is, dan print uit "Mogelijke waarden zijn: [1, 2], {naam: 'Bart'}, null"
 - "undefined" is, dan print uit "De waarde is: undefined"
 - "function" is, dan print uit "De waarde is een functie, bv: function hihi() {}"
- Indien het type niet gekend is, dan wordt er uitgeprint "Geen informatie over dit type!".

Opdracht 3

Indien Opdracht 2 opgelost is met een `switch`, herschrijf het met `if else if else`. Indien

Opdracht 2 opgelost is met een `if else if else`, herschrijf het met `switch`.

Iteratieve statements

Opdracht 1

- Definieer een functie genaamd `doortellen`.
- De functie accepteert één parameter `getal`.
- Indien het `getal` groter is dan 100 of gelijk is aan 100, print uit "Het getal is 100 of meer!".
- Indien het `getal` kleiner is dan 100, laat de functie dan `doortellen` tot 100.

Voorbeelden:

```
doortellen(100); // "Het getal is 100 of meer!" wordt uitgeprint
doortellen(98); // De volgende getallen worden uitgeprint: 99 100
doortellen(89);
// De volgende getallen worden uitgeprint:
// 90 91 92 93 94 95 96 97 98 99 100
```

Opdracht 2

Vertrek vanuit de oplossing van Opdracht 1.

- De functie genaamd `doortellen` accepteert twee parameters:
- De eerste parameter blijft gelijk aan Opdracht 1.
- De tweede parameter noemt `printOneven`.
- Indien de parameter `printOneven` de waarde `true` bevat, dan worden enkel de `printOneven` getallen uitgeprint tijdens het `doortellen`.
- Indien de parameter `printOneven` de waarde `false` bevat, dan worden enkel de even getallen uitgeprint tijdens het `doortellen`.

Voorbeelden:

```
doortellen(89, true);
// De volgende getallen worden uitgeprint:
// 91 93 95 97 99

doortellen(89, false);
// De volgende getallen worden uitgeprint:
// 90 92 94 96 98 100
```

Alle kennis

Opdracht 1

Deel 1

```
let totaal = 0;

for (let i = 0; i < 10; i++) {
  totaal += i; // Zelfde als: totaal = totaal + i;
  console.log("totaal is nu: ", totaal);
}
```

Herschrijf bovenstaande code met een `while` loop i.p.v. een `for` loop.

Deel 2

Gebruik de oplossing van Opdracht 1 als basis:

- Definieer een functie genaamd `berekenTotaal`:
 - De functie bevat in het codeblok, de code van Opdracht 1.
 - Verwijder de `console.log` in de `while` loop.
 - De functie geeft het berekende totaal terug (na de `while` loop).
- Initialiseer een variabele genaamd `resultaat`, roep de functie `berekenTotaal` aan en ken het resultaat toe aan de variabele `resultaat`.
- Voeg een `console.log` toe, die uitprint "Het totaal is: " met daarachter de waarde van de variabele `resultaat`.

Deel 3

Gebruik de oplossing van Opdracht 2 als basis:

- Voeg een parameter toe aan de functie `berekenTotaal` genaamd `aantalLussen`.
- Zorg ervoor dat de `while` loop niet 10 keer herhaald wordt, maar dat de parameter `aantalLussen` bepaalt hoe vaak de `while` loop herhaald wordt.
- Verwijder de regel code waar de variabele `resultaat` geïnitieerd wordt.
- Vervang in de `console.log`, de variabele `resultaat` door `berekenTotaal(10)`.
- Voeg volgende `console.log` toe:
 - `console.log("Het totaal is: ", berekenTotaal(25))`
 - `console.log("Het totaal is: ", berekenTotaal(100))`
 - `console.log("Het totaal is: ", berekenTotaal(666))`
 - `console.log("Het totaal is: ", berekenTotaal(10000))`

Deel 4

Gebruik de oplossing van Opdracht 3 als basis:

- Definieer een functie genaamd `printTotaal`:
 - De functie accepteert één parameter, genaamd `aantalLussen`
 - Het codeblok bevat volgende code: `console.log("Het totaal is: ", berekenTotaal(aantalLussen));`
- Vervang de vijf `console.log` onderaan door:

```
printTotaal(10);
printTotaal(25);
printTotaal(100);
printTotaal(666);
printTotaal(10000);
```

Opdracht 2

```
const naam = "John Duck";

if (naam === "John Duck") {
  console.log("Hallo John Duck");
}
```

Schrijf bovenstaande `if` om naar een `switch`.

Opdracht 3

```
const naam = "John Duck";

if (naam === "John Duck") {
  console.log("Hallo John Duck");
} else {
  console.log("Hallo vreemdeling");
}
```

```
}
```

Schrijf bovenstaande `if-else` om naar een `switch`.

Opdracht 4

```
const naam = "John Duck";

if (naam === "John Duck") {
  console.log("Hallo John Duck");
} else if (naam === "Janine Duck") {
  console.log("Hallo Janine Duck");
} else {
  console.log("Hallo vreemdeling");
}
```

Schrijf bovenstaande code om naar een `switch`.

Opdracht 5

Deel 1

```
const neefjesDonaldDuck = ["Kwik", "Kwek", "Kwak"];
```

Schrijf code die elk element van de array `neefjesDonaldDuck` uitprint. Maak gebruik van een `for` loop.

Deel 2

Extra informatie: Objecten kunnen complexe data voorstellen. Een object kan bijvoorbeeld een persoon voorstellen. Een object kan properties/eigenschappen hebben. Als voorbeeld, een variabele genaamd `persoon` kan een object toegekend krijgen.

```
const persoon = {};
```

Aangezien dit object een persoon voorstelt, kan dit object properties hebben, bijvoorbeeld `voornaam`, `achternaam`, `leeftijd`, `lengte`.

```
const persoon = {
  voornaam: "Bart",
  achternaam: "Duisters",
  leeftijd: "29",
  lengte: "174",
};
```

Om de lengte te weten van deze persoon, kan de lengte opgevraagd worden met de `puntnotatie`.

```
console.log(persoon.lengte); // Dit print: 174
```

Een array is van het type "object". Dit object heeft ook bepaalde properties waar informatie over de array in bijgehouden wordt. Één van die properties is de property `length`. Met de property `length` kan opgevraagd worden hoeveel elementen aanwezig zijn in de array.

Voorbeeld:

```
const neefjesDonaldDuck = ["Kwik", "Kwek", "Kwak"];
console.log(neefjesDonaldDuck.length); // Dit print: 3
```

```
const neefjesDonaldDuck = ["Kwik", "Kwek", "Kwak", "Kwok"];
console.log(neefjesDonaldDuck.length); // Dit print: 4
```

Donald Duck krijgt een extra neefje, genaamd "Kwok". Voeg "Kwok" toe aan de array `neefjesDonaldDuck`.

Print opnieuw alle elementen van de array `neefjesDonaldDuck` uit. Een oplossing is om het tweede statement van de for loop te wijzigen van `i < 3` naar `i < 4`.

Maar wat als Donald Duck nog een vijfde neefje krijgt? Dan moet de code opnieuw gewijzigd worden. Dat is te veel werk! Maak gebruik van bovenstaande kennis om te zorgen dat de for loop altijd alle elementen van de array uitprint, ongeacht de lengte van de array.