

# WORKSHOP

Lussen (oftewel loops)

VISTA College ICT

VI  
STA  
COLLEGE

# Inhoudsopgave

- De basis van programmeren
- Wat zijn lussen?
- Soorten lussen
  - For loop
  - While loop
- Opdracht maken



# De basis van programmeren

Variabelen: var, let, \$, string, int, ....

Keuzes maken: if, else, while

Herhalingen programmeren: for, while, foreach, do, ....



# Wat is een *loop*?

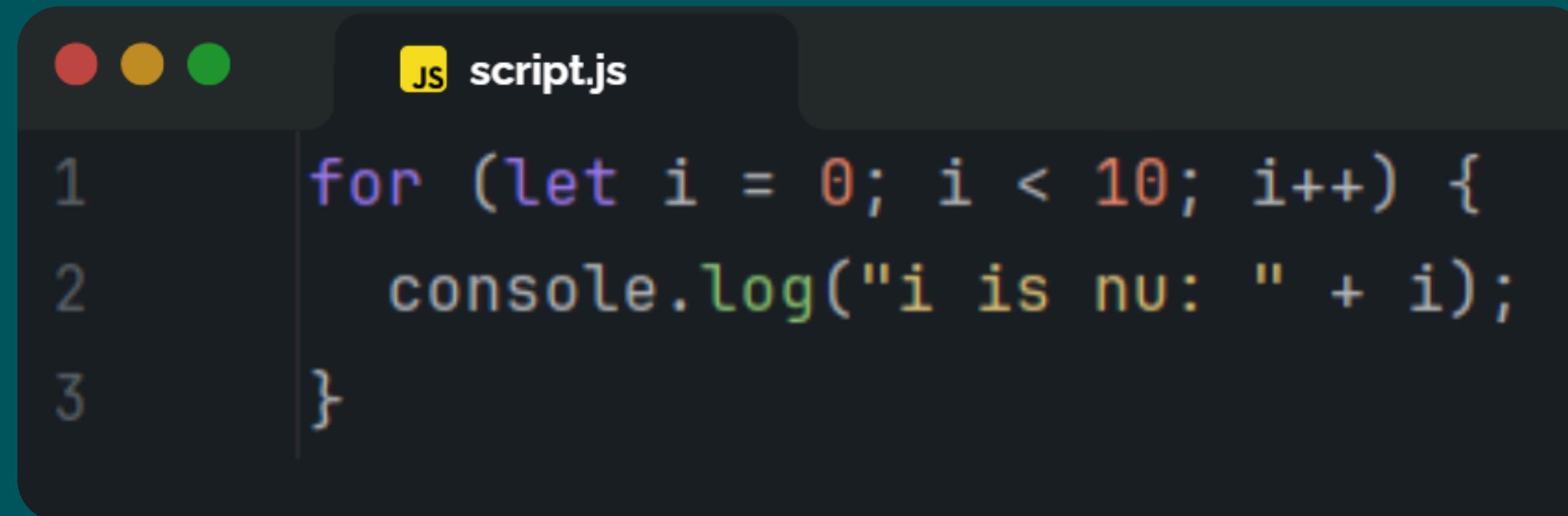
Een loop (of “lus”) is een manier om iets meerdere keren automatisch te herhalen...

- zonder dat je dezelfde code telkens opnieuw hoeft te schrijven;
- iets met de elementen uit een lijst wil doen.



# Soorten loops

## For loop in JavaScript



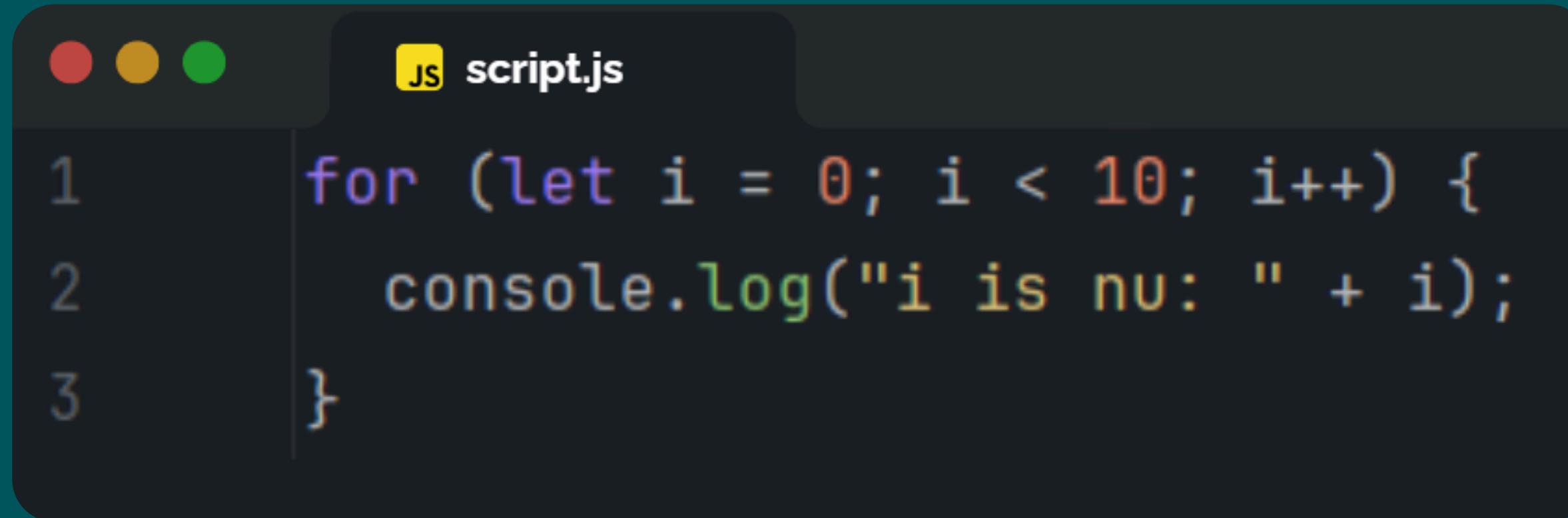
A screenshot of a code editor window titled "script.js". The window has three circular status indicators at the top left: red, yellow, and green. The code editor displays the following JavaScript code:

```
1 for (let i = 0; i < 10; i++) {  
2     console.log("i is nu: " + i);  
3 }
```



# Soorten loops

## For loop



The screenshot shows a code editor window titled "script.js". The code is a simple for loop:

```
1 for (let i = 0; i < 10; i++) {  
2     console.log("i is nu: " + i);  
3 }
```

The code editor has three status indicators at the top left: a red dot, a yellow dot, and a green dot.

Er staan drie statements tussen de haakjes:

1. We maken een variabele aan, i, en vullen i met waarde 0
2. We voeren de loop uit zolang i kleiner is dan 10
3. Elke keer dat we de loop doorlopen, verhogen we i met 1

code: Elke loop printen we de waarde van i

# Soorten loops

## While loop in JavaScript



A screenshot of a code editor window titled "script.js". The window has a dark theme with three circular status indicators at the top left. The code itself is a simple while loop:

```
1 let i = 0;
2
3 while (i < 10) {
4     console.log("i is nu: " + i);
5     i++;
6 }
```



# Soorten loops

## While loop

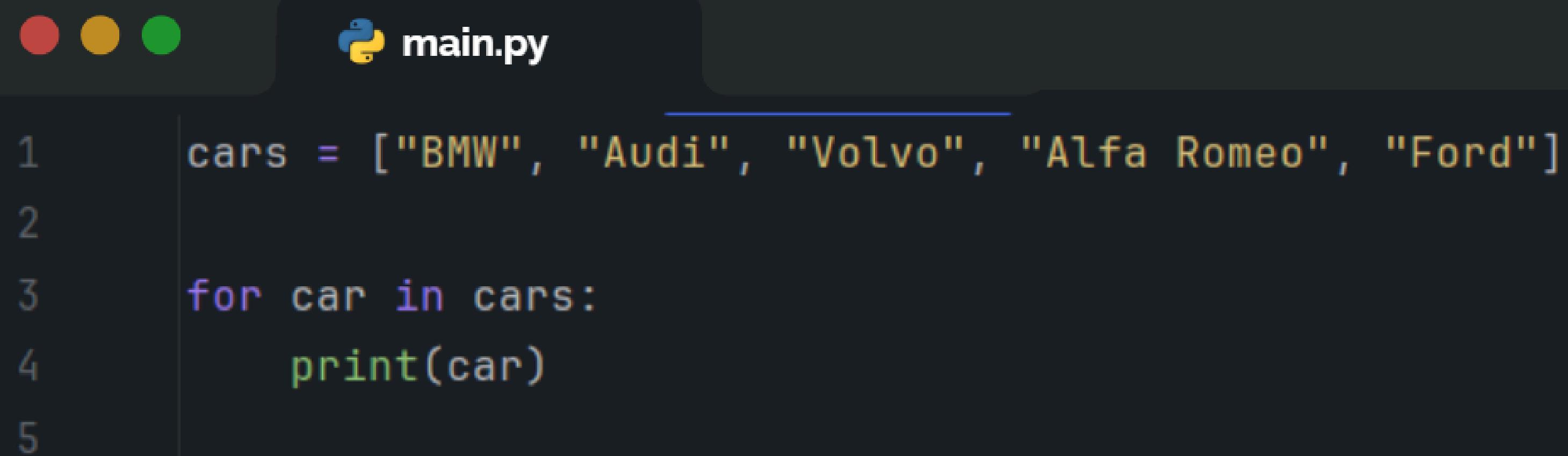
```
● ● ● JS script.js
1 let i = 0;
2
3 while (i < 10) {
4     console.log("i is nu: " + i);
5     i++;
6 }
```

- We maken een variabele aan, i, en vullen i met waarde 0
- We voeren de loop uit zolang i kleiner is dan 10
- Elke keer dat we de loop doorlopen, verhogen we i met 1
- Elke loop printen we de waarde van i



# Soorten loops

Over een lijst heen loopen in python



The image shows a dark-themed code editor window with three circular status indicators at the top left (red, yellow, green). The title bar on the right says "main.py". The code area contains the following Python script:

```
1 cars = ["BMW", "Audi", "Volvo", "Alfa Romeo", "Ford"]
2
3 for car in cars:
4     print(car)
5
```

A blue horizontal line highlights the word "print" in line 4.



# Jullie gaan... zometeen een zoekmachine maken!

- Kies een programmeertaal
- Maak een lijst van namen (alle studenten uit je klas)
- Zorg dat de gebruiker naar een van de studenten kan zoeken
- Staat de student in de lijst? Laat de gebruiker dit dan weten!

(Gevonden  / Niet gevonden )



# Aanpak

- Stap 1: Een lijst maken van namen
- Stap 2: De gebruiker vragen voor een naam
- Stap 3: Over elke naam in de lijst van namen *heen loopen*
- Stap 4: Checken of de ingevulde naam overeenkomt met een van de elementen uit de lijst van namen
- Stap 5: Feedback geven aan de gebruiker (gevonden of niet)



# Nu jullie!

- Kies een programmeer-/scripttaal (JS, Python, PHP, Java, ....)
- Maak een lijst (array, list, tupel, Dictionary) van namen (alle studenten uit je klas)
- Zorg dat de gebruiker naar een van de studenten kan zoeken (if else)
- Staat de student in de lijst? Laat de gebruiker dit dan weten!

(Gevonden  / Niet gevonden )



Type	Veranderbaar	Geordend	Unieke elementen	Toelaatbare types	Voorbeeld
<b>list</b>	✓ Ja	✓ Ja	✗ Nee	Verschillend	[1, 2, "a"]
<b>tuple</b>	✗ Nee	✓ Ja	✗ Nee	Verschillend	(1, 2, "a")
<b>array</b>	✓ Ja	✓ Ja	✗ Nee	Eén type (vaak num.)	array('i', [1,2,3])
<b>set</b>	✓ Ja	✗ Nee	✓ Ja	Eén of meerdere	{1, 2, 3}
<b>dict</b>	✓ Ja	✓ Ja (vanaf 3.7)	Sleutels uniek	Sleutel/waarde	{"a": 1, "b": 2}