

Empowering Creative Thinking Through Programming

Les 5: Functions deel 2

Wat gaan we doen

Scope

Functions met meerdere parameters

Return

Variabelen vullen met waardes die returned
worden uit functions

Functions aanroepen in andere functions

Vorige week functies met parameters

- De vraag was eigenlijk: waarom zou ik parameters gebruiken? Het werkt toch ook met allemaal globale variabelen?
- Parameters zijn maar de helft van het verhaal (deze week de andere helft)
- Maar eerst nog even iets over variabelen.

Waar is mijn variabele?

Ik kan 'm niet vinden..

Scope van variabelen:

Gelden ze voor het hele programma of maar voor
een klein stukje

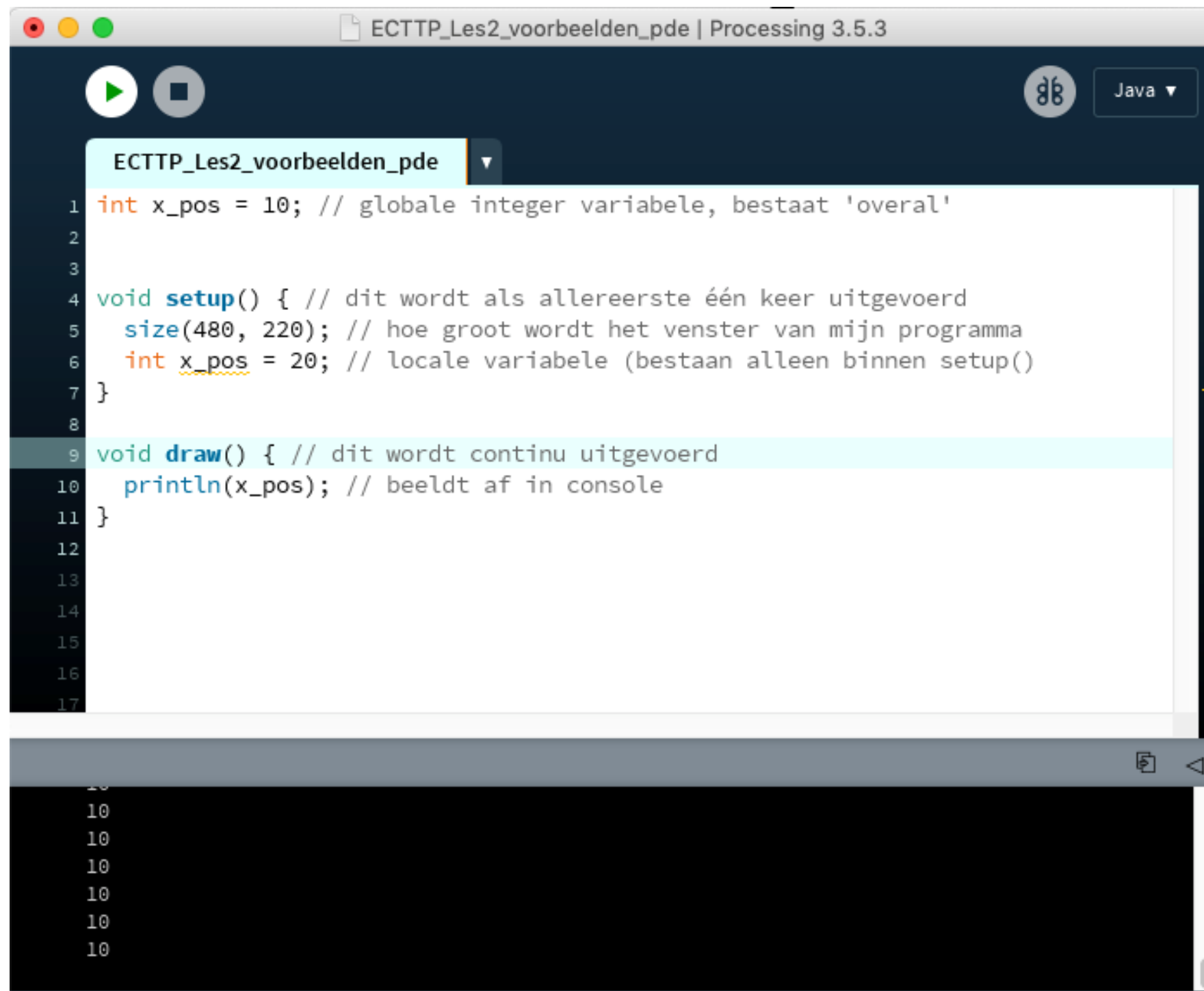
Global: geldt voor hele programma

Local: Geldt voor een klein stukje

Afhankelijk van waar je ze aanmaakt

Waar is mijn variabele?

Ik kan 'm niet vinden..



The screenshot shows the Processing IDE interface. The title bar reads "ECTTP_Les2_voorbeelden_pde | Processing 3.5.3". The code editor displays the following Java code:

```
1 int x_pos = 10; // globale integer variabele, bestaat 'overal'
2
3
4 void setup() { // dit wordt als allereerste één keer uitgevoerd
5   size(480, 220); // hoe groot wordt het venster van mijn programma
6   int x_pos = 20; // locale variabele (bestaan alleen binnen setup())
7 }
8
9 void draw() { // dit wordt continu uitgevoerd
10  println(x_pos); // beeldt af in console
11 }
12
13
14
15
16
17
```

Below the code editor, the console output shows the value of `x_pos` being printed repeatedly as `10`.

Scope van variabelen versus parameters

- Variabelen kunnen local of global zijn (zie vorige slide)
- Parameters zijn alleen maar local. Ze bestaan altijd alleen maar binnen de function waarin ze aangemaakt worden
- Aanmaken van parameters van een function tussen de haakjes achter de naam() van de function

In code

```
voorbeeld_1
1 // voorbeeld functions: local vs global
2 // door Ton Markus En Vincent Booman, september 2019
3 int x_pos=0;
4 int y_pos=0;
5 int x_snelheid=10;
6 int y_snelheid=20;
7
8 void setup() {
9     size(100, 100);
10 }
11
12 void draw() {
13     berekenNieuwePositie(x_pos,x_snelheid);
14     println(x_pos);
15 }
16
17 void berekenNieuwePositie(int oudePositie, int snelheid) {
18     x_pos = oudePositie + snelheid;
19 }
20
21
```

1. Globale variabelen (buiten functions aangemaakt)

2. Stop waarde van x_pos in 1e parameter

3. Stop waarde van x_snelheid in 2e parameter

4. Waardes van globale var's worden
doorgegeven aan parameters van
function

5. Lokale parameter waardes zorgen voor
nieuwe waarde van globale variabele

6. Print globale
variabele

Maar als ik die function nou voor zowel de x als de y positie wil gebruiken?

Want de rekensom is in beide gevallen hetzelfde:

$$\text{nieuwePositie} = \text{oudePositie} + \text{snelheid}$$

```
15 }  
16  
17 void berekenNieuwePositie() {  
18     x_pos = x_pos + x_snelheid;  
19     y_pos = y_pos + y_snelheid;  
20 }  
21
```

En ja, dit kan ook, maar we waren bezig geen code te herhalen, en dit ziet er toch wel erg hetzelfde uit...

Het antwoord is Return

Ofwel:

Waarom staat er void voor de naam van mijn function?

Functions kunnen een waarde terug sturen naar de plek waar ze aangeroepen worden.

Dit gebeurt met de opdracht return.

Als er geen waarde terug gestuurd wordt moet dat expliciet met void aangegeven worden.

```
12 void draw() {  
13     berekenNieuwePositie(x_pos,x_snelheid);  
14     println(x_pos);  
15 }  
16  
17 void berekenNieuwePositie(int oudePositie, int snelheid) {  
18     x_pos = oudePositie + snelheid;  
19 }  
20
```

Er wordt alleen een globale variabele aangepast

Waarom moet er int voor de naam van mijn function?

Als je return gebruikt moet je aangeven wat voor soort waarde er terug gegeven wordt (net als bij variabelen: Void als er niks terug komt, int als het een cijfer is etc.

Met de waarde die terug komt van de function kan je een variabele vullen



Waarom is dit zinvol?

Je kan nu 1 function gebruiken om meerdere globale variabelen te manipuleren

```
11
12 void draw() {
13     x_pos = berekenNieuwePositie(x_pos,x_snelheid);
14     y_pos = berekenNieuwePositie(y_pos,y_snelheid);
15     z_pos = berekenNieuwePositie(z_pos,z_snelheid);
16     println("x-positie = " + x_pos);
17 }
18
19 int berekenNieuwePositie(int huidigePositie, int snelheid) {
20     int nieuwePositie = huidigePositie + snelheid;
21     return nieuwePositie;
22 }
23
24
```

Maar dan heb ik nog steeds voor iedere positie een aparte println() nodig...

Function aanroepen binnen andere function

- Voor opdelen complexe code

```
11
12 void draw() {
13     x_pos = berekenNieuwePositie(x_pos,x_snelheid,"x");
14     y_pos = berekenNieuwePositie(y_pos,y_snelheid,"y");
15 }
16
17 int berekenNieuwePositie(int huidigePositie, int snelheid,String bewegingRichting) {
18     int nieuwePositie = huidigePositie + snelheid;
19     laatZienInConsole(nieuwePositie,bewegingRichting);
20     return nieuwePositie;
21 }
22
23 void laatZienInConsole(int positie, String richting) {
24     println("De nieuwe " + richting + "-positie is " + positie);
25 }
26
```

Verwacht tekst als parameter

Hier wordt de functie aangeroepen binnen een andere functie

Lesopdracht

A

- Maak een function die 2 parameters (int value1 en int value2) als input heeft.
- Tel binnen de function value1 en value2 bij elkaar op.
 - return het resultaat.

B

- Maak een globale variabele aan met als beginwaarde 0
- in de setup(): Vermeerder met behulp van de function uit het A deel de waarde van de variabele met 10
- in de draw(): Vermeerder met behulp van de function uit het A deel de waarde van de variabele met 3
- Gebruik println() om de resultaten in de console af te beelden. Je mag in het hele script maar 1 println() opdracht gebruiken, dus plaats 'm strategisch.