

# Les 1

- Sensoren & Actuatoren
- Functies
- I/O functies van de Arduino
- Servo aansturing met een Arduino

# Kahoot!

[www.kahoot.it](http://www.kahoot.it)

## Vraag 0

Bij welk getal beginnen we in programmeren met tellen?



Antwoord: 0

## Vraag 1

Wat voor soort data kan de variabele hebben?

```
int mijnVariabel;
```

Antwoord: positieve en negatieve hele getallen

Op de Arduino bestaat een integer (`int`) uit 16 bits. Dit betekent dat het getallen van -32.768 tot 32.767 op kan slaan. Op de meeste andere computers bestaat een integer uit 32 bits. Dit betekent dat het getallen van -2.147.483.648 tot 2.147.483.647 op kan slaan.

## Vraag 2

Wat is de nieuwe waarde van de variabele `tel`?

```
int tel = 3;
bool plus1 = false;
bool plus5 = true;

if(plus1 == true)
{
    tel = tel+1;
}
else if(plus5 == true)
{
    tel = tel+5;
}
```

Antwoord: 8

Het is 8 omdat alleen `plus5 true` is. De eerste `if` wordt dus niet uitgevoerd omdat `plus1` niet `true` is. De `else if` wordt wel uitgevoerd omdat `plus5 true` is. In deze `else if` tellen we 5 op bij de waarde die in `tel` staat (3). Dit nieuwe getal (8) schrijven we weer terug in de variabele `tel`.

## Vraag 3

Wordt de variabele `tel` ooit 2?

```
int tel = 3;
bool blijfDoorgaan = true;

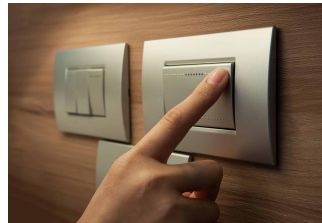
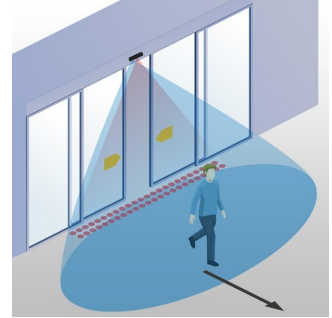
while(blijfDoorgaan == true)
{
    tel = 5;
}
tel = 2;
```

Antwoord: Nee

Wanneer we voor de eerste keer kijken of de `while` loop uitgevoerd moet worden is `blijfDoorgaan true`. Eenmaal in de `while` loop schrijven we het getal 5 in de variabele `tel`. De variabele `blijfDoorgaan` passen we niet aan. De variabele `blijfDoorgaan` blijft dus `true`. Elke keer wanneer we op het einde van de `while` loop zijn kijken we op we het nog een keer moeten herhalen. Met deze code komen we dus nooit uit de `while` loop en zal de regel `tel = 2;` dus ook nooit worden uitgevoerd.

# Wat zijn sensoren

Alles wat signalen kan opvangen/meten van zijn omgeving (zintuigen)

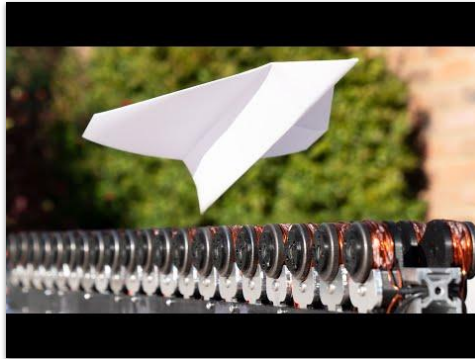


# Wat zijn actuatoren

Alles wat invloed kan uitoefenen op zijn omgeving



<https://youtu.be/uwMSUANqI-o?t=402>



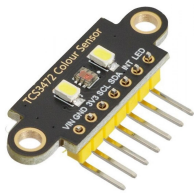
[https://youtu.be/\\_4TGb3MsSIE](https://youtu.be/_4TGb3MsSIE)





# Sensoren en actuatoren voor Arduino

Een paar voorbeelden



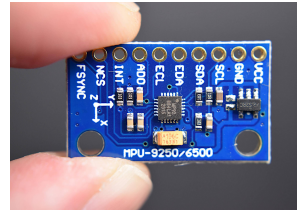
Kleurensensor



Servomotor



Afstandssensor



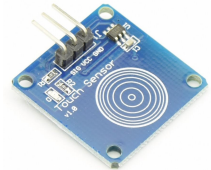
Acceleratiesensor + gyroscoop



Geluidssensor



DC motor



Touchsensor

Deze gaan we gebruiken voor de sorteerder.

# Functionies

Een klein stukje programma dat je kan aanroepen wanneer je wilt

- **Net zoals** een variabel heeft het een type.
- **Net zoals** een variabel heeft het een naam.
- **Anders dan** een variabel slaat het een stukje code op die je uit kan voeren.
- **Anders dan** een variabel kan het parameters hebben.

In code ziet dit er zo uit:

```
int mijnFunctie(int par1, bool par2)
{
    // Wat code
    return 3;
}
```

The diagram illustrates the components of a C++ function definition with the following annotations:

- Het type van de functie** (wat kan de functie teruggeven): Points to the `int` return type.
- De naam van de functie** (waar je de functie mee aan roept): Points to the function name `mijnFunctie`.
- Parameters die je meegeeft aan de functie** (een soort variabelen): Points to the parameters `int par1` and `bool par2`.
- De return statement van een functie** (je eindigt de functie en geeft hetgeen achter "return" terug): Points to the `return 3;` statement.

10

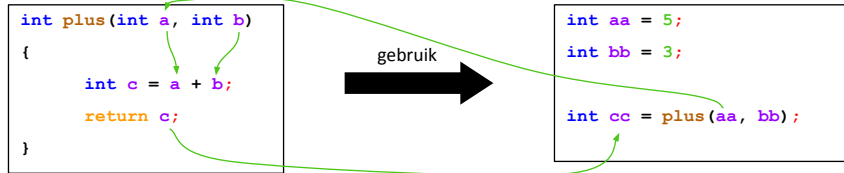
Op de volgende slide zie je voorbeelden van hoe je functies gebruikt.

Als het type van de functie `void` (void betekent "leeg") is dan kan je kiezen om of alleen `return;` te schrijven, of om helemaal geen return te gebruiken. In het laatste geval eindigt de functie wanneer het bij de laatste regel van de functie is.

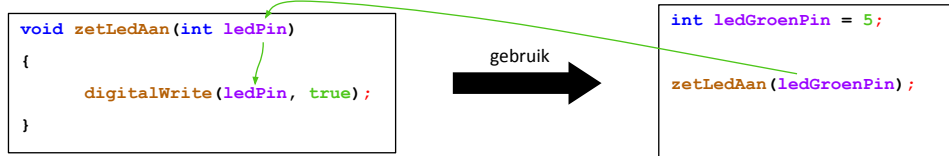
# Functies

Een klein stukje programma dat je kan aanroepen wanneer je wilt

- Een functie die twee getallen (integers) kan optellen en het resultaat teruggeeft.



- Een functie die een LED aan zet.
  - **void** betekent dat er niks wordt teruggegeven.
  - Je kan functies in functies aanroepen.
  - **digitalWrite** is een functie die bij de Arduino hoort.



# Arduino I/O functies

I/O: Input / Output

## Een pin configureren:

- `pinMode(pin, mode);`
  - Zet de pin in output of input mode.
    - Voor input: `pinMode(pin, INPUT);`
    - Voor output: `pinMode(pin, OUTPUT);`
  - <https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/>

## Input:

- `digitalRead(pin);`
  - Lees de digitale waarde (0/1) van de pin.
  - <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>

## Output:

- `digitalWrite(pin, waarde);`
  - Schrijf de digitale waarde (0/1) naar de pin.
  - <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>

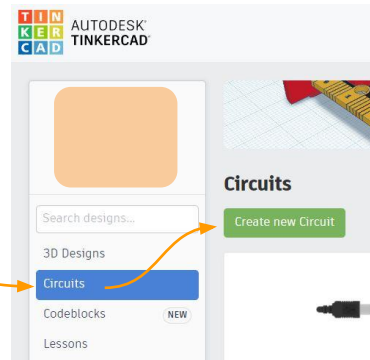
## Met een echte Arduino aan de gang

Volg **Arduino omgeving installeren**.  
Maak **Arduino knipperlicht**.

Dit kan je met **echte hardware** doen of in  
**Tinkercad** circuits: [www.tinkercad.com](https://www.tinkercad.com)

De projecten zijn te vinden op:

<https://gitlab.com/enzoever/lerenprogrammeren/-/tree/main/Projecten>



## YouTube kanalen



<https://www.youtube.com/user/makemagazine>



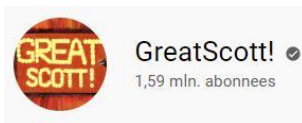
<https://www.youtube.com/c/MakeKids>



<https://www.youtube.com/c/8BitsandaByte>



<https://www.youtube.com/user/thebenheckshow>



<https://www.youtube.com/c/greatscottlab>



<https://www.youtube.com/c/Electroboom>