

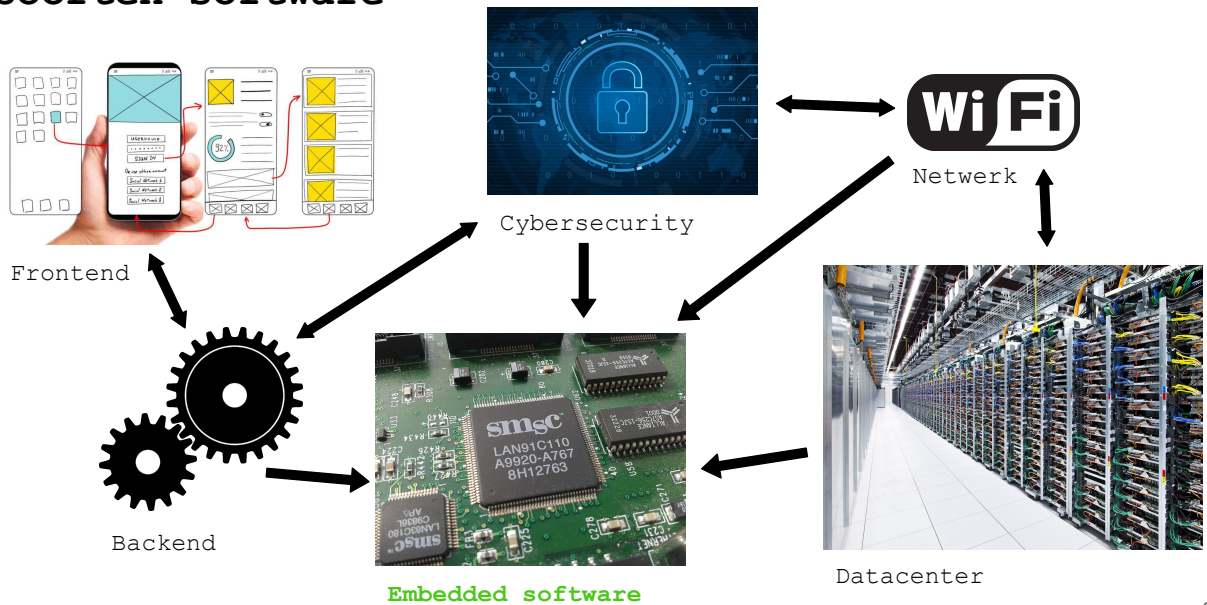
Les 0

- Wat is software
- Variabelen
- if-else statements
- while loops

Waar vind je software?



Soorten software



3

In deze lessen gaan we vooral op embedded software focussen met de Arduino.

Wat is software

Een tekst die aan de computer vertelt wat het moet doen.

Voorbeeld van software is het echte leven:

Jij vraagt aan mij hoe je naar de dichtstbijzijnde Jumbo gaat. Ik zeg:

1. Loop uit dit lokaal.
2. Loop naar buiten.
3. Loop naar het hek.
4. Maak het hek open.
5. Draai een kwartslag naar links.
6. Loop 330 meter rechtdoor.
7. Draai een kwartslag naar links.
8. Loop 25 meter rechtdoor.
9. Wacht tot de schuifdeur open is.
10. Loop 20 meter rechtdoor.

Maar:

- Weet je hoe je naar buiten moet?
- Weet je hoe je het hek open moet maken?
- Hoe meet je of je een kwartslag gedraaid hebt?
- Hoe meet je hoe ver je al gelopen hebt?
- Hoeven we niet op te letten tijdens het oversteken?
- Hoe weten we op de schuifdeur open is?

Een computer weet dit niet! Je moet het alles vertellen.

Vertel mij de stappen
(het programma) zodat ik
van punt A naar punt B loop.



Laat iemand op een plek in de kamer staan (punt A) en bedenk een ander punt in de kamer waar hij/zij naar toe moet gaan (punt B). Degene die op punt A staat moet nu de commandos van de programmeur opvolgend. Bijvoorbeeld “loop 3 stappen vooruit”, “draai 90 graden naar links”, etc.

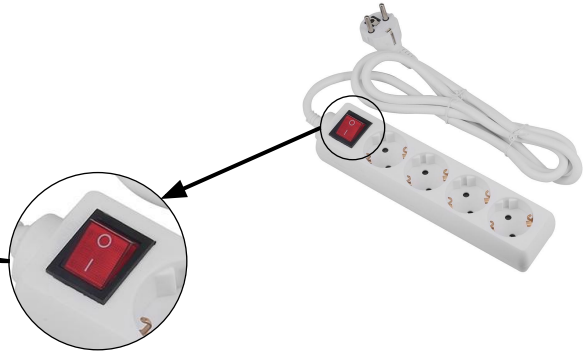
Basis van alle software

- Binair
- Data en variabelen
- Als(...) waar is, dan doe{...}. Anders doe{...}
- Zolang(...) waar is, doe {...}
- Functies

Met deze punten kan je alle software maken. De moeilijkheid van goede software maken ligt vooral in het kunnen vertalen van je ideeën naar stappen die in de computer uitgevoerd kunnen worden.

Binair

- Aan/Uit
- Waar/Niet-waar
- True/False
- 1/0



In programmeren noemen we “True/False” een **Boolean**. Dit wordt ook wel als **bool** afgekort in de code.

Binair (tweetallig stelsel)

Het telsysteem van een computer.

Tientallig stelsel (decimaal): 0,1,2,3,4,5,6,7,8,9

- Dit gebruiken we elke dag.

Tweetallig stelsel (binair): 0,1

- Dit gebruiken computers.

Het maximale getal is behaald. Dus begin weer vanaf 0 en hoog het linkse getal met 1 op.

**Deze informatie is niet nodig voor de lessen. Alleen voor de geïnteresseerden.*

Decimaal	Binair
00	0000
01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Voorbeeld: het getal 105 in:

- Decimaal: 105
- Binair : 1101001

1	0	5
100	10	1
$1 \times 100 = 100$	$0 \times 10 = 0$	$5 \times 1 = 5$
$100 + 0 + 5 = 105$		

1	1	0	1	0	0	1
64	32	16	8	4	2	1
$1 \times 64 = 64$	$1 \times 32 = 32$	$0 \times 16 = 0$	$1 \times 8 = 8$	$0 \times 4 = 0$	$0 \times 2 = 0$	$1 \times 1 = 1$
$64 + 32 + 0 + 8 + 0 + 0 + 1 = 105$						

8

Dit is extra informatie die niet per se nodig is om te leren voor nu.

Deze slide wilt duidelijk maken dat binair tellen eigenlijk hetzelfde werkt als tellen in ons tientallig stelsel, maar in binair gebruiken we alleen 0 en 1. Misschien heb je in de rekenles wel ooit geleerd welke waarde een getal heeft. In het getal 105 is de 1 (of eigenlijk 100) een honderdtal, de 0 (of eigenlijk 00) een tiental en de 5 een eenheid. In binair hebben we hetzelfde maar gebruiken we andere waarden. Deze zie je op de tweede rij van de onderste tabel (de 64, 32, 16, etc.).

Als je wilt weten hoe we aan deze wegingen komen dan is dat het volgende. In het tientallig stelsel hebben we voor elke positie 10 mogelijkheden (0 t/m 9). Op de plaats van de eenheid (positie 0) is de weging 10^0 (spreek uit als "10 tot de macht 0") en dit is gelijk aan 1 ($10^0 = 1$). Op de plaats van het tiental (positie 1) is de weging 10^1 (spreek uit als "10 tot de macht 1") en dit is gelijk aan 10 ($10^1 = 10$). Hetzelfde geldt voor het honderdtal op positie 2 wat een weging heeft van 10^2 wat gelijk is aan 100 ($10^2 = 10 \times 10 = 100$).

Voor de wegingen van het tweetallig stelsel doen we precies hetzelfde. Maar nu heeft elke positie van 2 mogelijkheden (0 en 1). De wegingen zijn dus $2^0 = 1$ (spreek uit van "2 tot de macht 0"), $2^1 = 2$ (spreek uit van "2 tot de macht 1"), $2^2 = 2 \times 2 = 4$ (spreek uit van "2 tot de macht 2"), $2^3 = 2 \times 2 \times 2 = 8$, etc,

Data en variabelen

Variabel: een opslagplaats met een naam.

Data: Wat je in een variabel opslaat.

In code ziet dit er zo uit:

```
int mijnVariabel = 103;
```

Het type van de variabele
(wat kan er in)

De naam van de variabele
(hoe heet het doosje)

De waarde van de variabele
(wat zit er in)

Afsluiting



Per programmeertaal verschilt het een beetje hoe een variabele wordt geschreven, maar in elke programmeertaal heeft een variabele dezelfde functie.

- In dit geval is een "int" een afkorting van "integer" wat betekent "geheel getal".
- De naam van je variabele kan je zelf kiezen
- De afsluiting ";" is nodig zodat het programma weet dat het het einde van een zin is. Hetzelfde als een "." in een boek.

Data en variabelen

- Je leeftijd
 - **type**: een getal
 - **naam**: leeftijd
 - **waarde**: je huidige leeftijd
 - In code: `int leeftijd = 12;`
- Een straatnaam
 - **type**: tekst
 - **naam**: straatnaam
 - **waarde**: de naam van de straat
 - In code: `String straatnaam = "Gerretsonlaan";`
- Om te weten of de computer aan staat
 - **type**: een boolean (true/false, waar/niet-waar, aan/uit)
 - **naam**: computerAan
 - **waarde**: of de computer aan staat of niet
 - In code: `bool computerAan = true;`

Als (...) waar is, dan doe{...}. Anders doe{...}
if (...) {...} else {...}

Het Engelse woord voor “als” is “if”.

Het Engelse woord voor “anders” is “else”.

De zin

“Als (er meer dan 4 regendruppels vallen), {pak dan de paraplu}.
Anders {pak de paraplu niet}.”


schrijf je in code zoals je hiernaast ziet.



```
int aantalRegendruppels = 5;  
bool pakParaplu = false;  
  
if (aantalRegendruppels > 4)  
{  
    pakParaplu = true;  
}  
else  
{  
    pakParaplu = false;  
}
```

Door de waarde in de variabel `aantalRegendruppels` aan te passen zetten we de variabel `pakParaplu` op `true` of `false`.

Eigenlijk is de het gedeelte hier helemaal niet nodig. We zetten `pakParaplu` namelijk in het begin al op `false`. We moeten het alleen naar `true` veranderen als er meer dan 4 regendruppels vallen.



De **while** loop blijft zichzelf herhalen totdat (...) niet meer waar is.

**“Zolang (er meer dan 4 regendruppels vallen),
{pak de paraplu}.”**

schrijf je in code zoals je hiernaast ziet.

```
int aantalRegendruppels = 5;
bool pakParaplu = false;

while(aantalRegendruppels > 4)
{
    pakParaplu = true;
}
```

We hebben hier hetzelfde voorbeeld als op de vorige slide met de if-else. Maar bij de if-else kijken we één keer of we **pakParaplu** op **true** of **false** moeten zetten. Met een while-loop blijven we de paraplu pakken zolang het genoeg regent. Wanneer we voor het eerst bij de while-loop komen kijken we op we er in moeten gaan of niet (we kijken of **aantalRegendruppels** > 4). Als dit niet het geval is dan slaan we de hele while-loop over. Als we er wel in gaan dan blijven we in de loop totdat **aantalRegendruppels** > 4 niet meer waar is. Dit is anders dan bij de if-else. In de if-else keken we ook of we in de if moesten gaan. Als we er in moesten dan voerden we wat er in de if stond maar één keer uit en gingen daarna door met de rest van het programma, ook al regende het nog meer dan 4 druppels.

Alles samen. Aftellen voor de start

Gewone tekst:

Ik tel van **3** naar **0**.

Als ik **nog niet bij 0** ben blijf ik **aftellen** en staat **het licht aan**.

Als ik **bij 0** ben dan mag de race starten en gaat het **licht uit**.

```
graph TD; A["int tel = 3;  
bool lichtAan = true;"] --> B["tel = 2;"]; B --> C["tel = 1;"]; C --> D["tel = 0;  
lichtAan = false;"];
```

```
int tel = 3;  
bool lichtAan = true;  
  
tel = 2;  
  
tel = 1;  
  
tel = 0;  
lichtAan = false;
```

13

We beginnen met twee variabelen te maken. De variabel `int tel` die in gehele getallen de tel bijhoud. En de variabel `bool lichtAan` die bijhoud of het licht aan moet staan of niet. Om af te tellen veranderen we de variabel `tel` simpelweg met onze volgende tel. Als we eenmaal bij `0` zijn gekomen dan mag het licht uit (dus `lichtAan = false;`).

Alles samen. Aftellen voor de start

Gewone tekst:

Ik tel van **3** naar **0**.

Als ik **nog niet bij 0** ben blijf ik **aftellen** en staat **het licht aan**.

Als ik **bij 0** ben dan mag de race starten en gaat het **licht uit**.

Code

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

while(tel > 0)
{
    tel = tel-1;
    if(tel == 0)
    {
        lichtAan = false;
    }
}
```

14

Hier zie je de vorige slide in code. We gebruiken hier zowel variabele, een while-loop en een if-statement. Waar we in vorige slide telkens `tel = 2;` en `tel = 1;` deden, doen we nu `tel = tel-1;`. We zeggen hiermee dus dat de nieuwe waarde van `tel`, de huidige waarde van `tel` is maar dan 1 minder. In de if-statement kijken we of `tel` al `0` is en dus of het licht uit mag. In de while-loop kijken we of we nog een keer moeten aftellen of dat we al bij `0` zijn en we dus uit de while-loop mogen gaan.

Zelf nog meer doen?

Websites en platformen:

- <https://www.arduino.cc/>
- <https://scratch.mit.edu/>
- <https://microbit.org/lessons/>
- <https://makeymakey.com/>
- <https://www.tinkercad.com/>



AUTODESK®
TINKERCAD®



Plekken om naartoe te gaan:

- MakersClub
 - <https://eindhoven.op-shop.nl/97/makersclub-also-in-english>
- Tech Playgrounds
 - <https://dynamojeugdwerk.nl/tech-playgrounds>
- De Ontdekfabriek
 - <https://www.deontdekfabriek.nl>



de Bibliotheek
Eindhoven



enzoeverers@hotmail.com