

04 verslagje 19 mrt 2019 If en if ... else  
Ronde haakjes  
Switch-case  
break

Voorbeeld: als het geld op is, ga pinnen.

```
if (GeldInPortemonnaie == 0){  
    GeldGaanPinnen;  
}  
else {  
    ErIsVoldoendeGeld  
}
```

*Het “else” statement wordt altijd aan het einde van een “if ... then ...” geplaatst, dus als de laatste “conditie” wat wordt uitgevoerd als alle condities falen ...*

Else komt alleen in beeld, als alle andere condities onwaar zijn.

Else hoeft dus niet perse aanwezig te zijn, zoals hieronder kan het ook.

```
if ( conditie ) {  
    // wat gedaan moet worden als de conditie waar (true) is  
}
```

De vergelijkingen zijn booleaans:

- == is gelijk aan
- != is NIET gelijk aan
- < is kleiner dan
- > is groter dan
- <= is kleiner of gelijk aan
- >= is groter of gelijk aan

*De meest voorkomende fout bij vergelijking Operators is dat men het symbool “=” gebruikt (toewijzing!) in plaats van het “==” teken als we kijken of twee waarden gelijk zijn.*

```
if ( true ) {  
    // doen als de conditie waar is  
}
```

Bovenstaande if zal altijd worden uitgevoerd, dit is altijd waar.

Een werkend voorbeeld:

```
void setup() {  
    Serial.begin(9600);          //monitor snelheid  
    int ZakGeld = 4;  
    int SpaarGeld = 12;  
    int AlMijnGeld = ZakGeld + SpaarGeld;  
    Serial.print("Geld op zak = ");  
    Serial.println(ZakGeld);  
    Serial.print("Spaargeld = ");  
    Serial.println(SpaarGeld);  
    Serial.print("Al mijn Geld = ");  
    Serial.println(AlMijnGeld);  
    if(AlMijnGeld<5) {  
        Serial.println("Oh oh,... we hebben maar weinig geld!!");  
    }  
    else {  
        Serial.println("Geen probleem, we hebben genoeg geld.");  
    }  
}
```

```
void loop() {
  // even leeg laten
}
```

De uitkomst, dat wat wordt afgedrukt is:

```
Geld op zak = 4
Spaargeld = 12
Al mijn Geld = 16
Geen probleem, we hebben genoeg geld.
```

Het -if- gedeelte is overgeslagen, omdat er voldoende geld in kas is.

Als we het spaargeld op -0- zetten dan zien we:

```
Geld op zak = 4
Spaargeld = 0
Al mijn Geld = 4
Oh oh,... we hebben maar weinig geld!!
```

Dit komt omdat de totale hoeveelheid geld (AlMijnGeld) kleiner dan € 5,- is.

Nu wordt de handeling na het -if- statement WEL uitgevoerd.

Het deel na de -else- wordt overgeslagen.

Je kan ook meerdere IF's achter elkaar zetten, de laatste ELSE is zonder IF.

```
if(AlMijnGeld==0) {
  Serial.println("PANIEK!!");
} else if(AlMijnGeld<5) {
  Serial.println("Oh oh,... we hebben maar weinig geld!!");
} else if(AlMijnGeld>10) {
  Serial.println("Woohoo - we zijn rijk!!");
} else {
  Serial.println("Geen probleem, we hebben genoeg geld.");
}
```

Deze code zal:

- "PANIEK!!" weergeven als AlMijnGeld = 0, **OF**
- "Oh oh,... we hebben maar weinig geld!!" weergeven als AlMijnGeld kleiner is dan 5, **OF**
- "Woohoo - we zijn rijk!!" weergeven als AlMijnGeld groter is dan 10, **OF**
- in alle andere situaties "Geen probleem, we hebben genoeg geld." weergeven.

Omdat het zakgeld € 4,- en het spaargeld € 12,- is krijg je onderstaande tekst te zien:

```
Geld op zak = 4
Spaargeld = 0
Al mijn Geld = 4
Oh oh,... we hebben maar weinig geld!!
```

**Zodra er een else if waar is, worden de overige else if's overgeslagen.**

Veranderen we zakgeld = 0 en spaargeld = 0

```
We zien dan: Geld op zak = 0
Spaargeld = 0
Al mijn Geld = 0
PANIEK!!
```

Maar de conditie: *AlMijnGeld<5* is ook waar, toch wordt die overgeslagen!

Bij overlappende condities wordt na de eerste waarheid, de rest overgeslagen.

**Houdt hier rekening mee!**

Je kan meerdere keren else achter elkaar gebruiken maar de HELE routine wordt verlaten zodra er een (de eerste) waarheid is gevonden. De rest wordt niet meer gecontroleerd. Dit kan je ondervangen door elke situatie van een eigen if .... else te voorzien.

Wil je toch alle condities weergeven, dan moet je de if's uit elkaar halen.

```
if(AlMijnGeld==0) {  
    Serial.println("PANIEK!!");  
}  
  
if(AlMijnGeld<5) {  
    Serial.println("Oh oh,... we hebben maar weinig geld!!");  
}  
  
if(AlMijnGeld>10) {  
    Serial.println("Woohoo - we zijn rijk!!");  
}
```

Je krijgt dan: Geld op zak = 0  
Spaargeld = 0  
Al mijn Geld = 0  
PANIEK!!  
Oh oh,... we hebben maar weinig geld!!

Dan krijg je de twee voorwaarden, die waar zijn, te zien.  
Is dat niet de bedoeling, dan moet je bij elke if een else if gebruiken.

Wanneer je met dat doet:

```
if(AlMijnGeld==0) {  
    Serial.println("PANIEK!!");  
} else {  
    Serial.println("Geen probleem");  
}  
  
if(AlMijnGeld<5) {  
    Serial.println("Oh oh,... we hebben maar weinig geld!!");  
} else {  
    Serial.println("Geen probleem");  
}  
  
if(AlMijnGeld>10) {  
    Serial.println("Woohoo - we zijn rijk!!");  
} else {  
    Serial.println("Geen probleem");  
}  
}
```

Je krijgt nu dit te zien: Geld op zak = 0  
Spaargeld = 0  
Al mijn Geld = 0  
PANIEK!!  
Oh oh,... we hebben maar weinig geld!!  
Geen probleem

Vooraf dat laatste is niet waar, maar het is de voorwaarde achter de 'else' die geprint wordt, zodra de voorwaarde achter de 'if' niet voldoet. Hier past alleen de weergave: PANIEK!!

Als de if niet waar is, dan wordt de else uitgevoerd, maar bij meerdere keren 'niet waar' wordt telkens de achterliggende else uitgevoerd. Bij overlappende condities geeft dat problemen.

Als 'almijngeld' de waarde 8 heeft (bijvoorbeeld door zakgeld en spaargeld op 4 te zetten.

```
Geld op zak = 4
Spaargeld = 4
Al mijn Geld = 8
Geen probleem
Geen probleem
Geen probleem
```

Je krijgt nu drie maal 'geen probleem' te zien, dat kan ook niet de bedoeling zijn.

Achter elke voorwaarde waaraan niet wordt voldaan, wordt de else uitgevoerd

**Dat overlappen is op te vangen door booleaanse kreten and-or-not** And **&&** or **||** en not !

```
voorbeeld:  if(almijngeld >= 5 && almijngeld <= 10){
             Serial.println("geld tussen 5 en 10");
             }
```

Waarheidstabellen van de booleaanse functies zijn bij les -3- gegeven.

Wat wordt eerst uitgevoerd? De **&&** of **>=** of **<=**?

Net als Mijnheer Van Dalen Wacht Op Antwoord, weet je het nog? Ook hier gelden er regels.

Voor de duidelijkheid kan je het best haakjes gebruiken, er is dan geen verwarring.

De nette notatie is dus: **(AlMijnGeld >= 5) && (AlMijnGeld <= 10)**

```
if(AlMijnGeld<5) {
    Serial.println("Oh oh,... we hebben maar weinig geld!!");
}

if(AlMijnGeld>10) {
    Serial.println("Woohoo - we zijn rijk!!");
}

if( (AlMijnGeld>=5) && (AlMijnGeld<=10) ){
    Serial.println("Geen probleem, we hebben genoeg geld.");
}
}
```

Je krijgt dan wat gewenst is: Geld op zak = 0

Spaargeld = 4

Al mij geld = 4

Oh oh,... we hebben maar weinig geld!!

Vooraf voor complexe vergelijkingen is het gebruik van haakjes zeer nuttig!

Dit heet geneste condities, de condities die het diepst liggen zullen, net als bij de normale reken regels, het eerst worden afgehandeld.

In de les worden daar een heleboel (rekenkundige) voorbeelden van gegeven, maar wie de lagere school/basis school goed heeft doorlopen, weet dit al.

## Switch ... case ...

Deze functie is een soort 'super if'

```
void setup() {
    Serial.begin(115200);
    int A = 6;

    if(A==1) {
        Serial.println("A = 1");
    }
}
```

```

else if(A==2) {
    Serial.println("A = 2");
}
else if(A==3) {
    Serial.println("A = 3");
}
else if(A==4) {
    Serial.println("A = 4");
}
else if(A==5) {
    Serial.println("A = 5");
}
else if(A==6) {
    Serial.println("A = 6");
}
else if(A==7) {
    Serial.println("A = 7");
}
else if(A==8) {
    Serial.println("A = 8");
}
else if(A==9) {
    Serial.println("A = 9");
}
else if(A==10) {
    Serial.println("A = 10");
}
else {
    Serial.println("A is kleiner dan 1 of groter dan 10");
}
}

void loop() {
    // even leeg laten
}

```

Het resultaat is: **A = 6** en dat klopt!

```

switch (variabele) {
    case waarde1:
        // doe dit als variable gelijk is aan waarde1
        break;
    case waarde2:
        // doe dit als variable gelijk is aan waarde2
        break;
    ... // etc.

    default:
        // optioneel: doe dit als geen van de stellingen waar is
        break;
}

```

De switch vergelijkt A met elke waarde, is geen waarde geldig dan geeft hij de default waarde

```

void setup() {
  Serial.begin(115200);
  int A = 6;

  switch(A) {
    case 1: Serial.println("A = 1");
    case 2: Serial.println("A = 2");
    case 3: Serial.println("A = 3");
    case 4: Serial.println("A = 4");
    case 5: Serial.println("A = 5");
    case 6: Serial.println("A = 6");
    case 7: Serial.println("A = 7");
    case 8: Serial.println("A = 8");
    case 9: Serial.println("A = 9");
    case 10: Serial.println("A = 10");
    default: Serial.println("A is < 1 of > 10");
  }
}

void loop() {
  // even leeg laten
}

```

Hier is iets bijzonders aan de hand, kijk maar naar het resultaat:

```

A = 6
A = 7
A = 8
A = 9
A = 10
A is < 1 of > 10

```

Als één case voldoet, wordt alles na deze voldoening uitgevoerd!

Soms is dat handig, maar wil je dat niet, dan moet je deze eigenschap onderbreken met **break**.

Dus na elke case moet je break typen!

Dat ziet er zo uit:

```

switch(A) {
  case 1: Serial.println("A = 1");
          break;
  case 2: Serial.println("A = 2");
          break;
  case 3: Serial.println("A = 3");
          break;
  case 4: Serial.println("A = 4");
          break;
  case 5: Serial.println("A = 5");
          break;
  case 6: Serial.println("A = 6");
          break;
  case 7: Serial.println("A = 7");
          break;
  case 8: Serial.println("A = 8");
          break;
  case 9: Serial.println("A = 9");
          break;
  case 10: Serial.println("A = 10");
           break;
  default: Serial.println("A is < 1 of > 10");
}

```