

# Flitsuitleg 2: IfStatementConditional

© 2015 Richel Bilderbeek 

```
/*  
  Conditionals - If statement
```

This example demonstrates the use of if() statements.  
It reads the state of a potentiometer (an analog input) and turns on an LED  
only if the LED goes above a certain threshold level. It prints the analog value  
regardless of the level.

The circuit:

- \* potentiometer connected to analog pin 0.

Center pin of the potentiometer goes to the analog pin.

side pins of the potentiometer go to +5V and ground

- \* LED connected from digital pin 13 to ground

- \* Note: On most Arduino boards, there is already an LED on the board  
connected to pin 13, so you don't need any extra components for this example.

created 17 Jan 2009

modified 9 Apr 2012

by Tom Igoe

This example code is in the public domain.

<http://arduino.cc/en/Tutorial/IfStatement>

```
*/
```

```
/*  
Conditionals - If statement
```

This example demonstrates the use of if() statements.  
It reads the state of a potentiometer (an analog input) and turns on an LED  
only if the LED goes above a certain threshold level. It prints the analog value  
regardless of the level.

The circuit:

- \* potentiometer connected to analog pin 0.

Center pin of the potentiometer goes to the analog pin.

side pins of the potentiometer go to +5V and ground

- \* LED connected from digital pin 13 to ground

- \* Note: On most Arduino boards, there is already an LED on the board  
connected to pin 13, so you don't need any extra components for this example.

created 17 Jan 2009

modified 9 Apr 2012

by Tom Igoe

This example code is in the public domain.

<http://arduino.cc/en/Tutorial/IfStatement>

```
*/
```

# Meerlijns commentaar

- Ook wel: C stijl commentaar
- Aantekeningen voor jezelf
- Begint met /\*
- Eindigt met \*/

```
/*
```

```
Mijn programma
```

```
(C) 2015 Richel Bilderbeek
```

```
*/
```

```
// These constants won't change:
const int analogPin = A0;    // pin that the sensor is attached to
const int ledPin = 13;       // pin that the LED is attached to
const int threshold = 400;    // an arbitrary threshold level that's

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communications:
  Serial.begin(9600);
}
```

```
// These constants won't change:
```

```
const int analogPin = A0;    // pin that the sensor is attached to
const int ledPin = 13;       // pin that the LED is attached to
const int threshold = 400;    // an arbitrary threshold level that's
```

```
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communications:
  Serial.begin(9600);
}
```

# Enkelijns commentaar

- Aantekeningen voor jezelf
- Gaat vaak over de volgende regel
- Begint met //
- Eindigt aan het eind van de regel

```
// Bereken de tijd in seconden
```

```
// These constants won't change:
const int analogPin = A0;    // pin that the sensor is attached to
const int ledPin = 13;       // pin that the LED is attached to
const int threshold = 400;    // an arbitrary threshold level that's

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communications:
  Serial.begin(9600);
}
```



# Definities van variabelen

- Een variabele met een beginwaarde
- De variable heeft een data type
- Een definitie is een soort statement (= een regel waarin iets gebeurt)

```
int dozijn = 12;
```

```
const double pi = 3.14159265359;
```

```
// These constants won't change:
const int analogPin = A0;    // pin that the sensor is attached to
const int ledPin = 13;       // pin that the LED is attached to
const int threshold = 400;   // an arbitrary threshold level that's

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communications:
  Serial.begin(9600);
}
```

# Variabele naam

- De naam van een stukje geheugen om waarden in te stoppen
- Mag niet beginnen met een getal
- Kan geen spaties hebben

```
int mijn_variabele = /* */;
```

```
bool is_led_aan = /* */;
```

```
const double pi = /* */;
```

```
// These constants won't change:
const int analogPin = A0;    // pin that the sensor is attached to
const int ledPin = 13;       // pin that the LED is attached to
const int threshold = 400;   // an arbitrary threshold level that's

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communications:
  Serial.begin(9600);
}
```

# Data type

- Het soort gegevens dat een variabele is:
  - int: hele getallen
  - double: gebroken getallen
  - LiquidCrystalDisplay
  - void: geen data type
  - ...

```
int dozijn = 12;
```

```
const double pi = 3.14159265359;
```

```
// These constants won't change:
const int analogPin = A0;    // pin that the sensor is attached to
const int ledPin = 13;       // pin that the LED is attached to
const int threshold = 400;   // an arbitrary threshold level that's

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communications:
  Serial.begin(9600);
}
```

# Modifier

- Het aanpassing van het data type:
  - const: alleen lezen
  - Andere modifiers: mutable, static, volatile

```
const double pi = 3.14159265359;
```

```
// These constants won't change:
const int analogPin = A0;    // pin that the sensor is attached to
const int ledPin = 13;       // pin that the LED is attached to
const int threshold = 400;   // an arbitrary threshold level that's

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communications:
  Serial.begin(9600);
}
```



# Definitie van een variabele

- Een variabele met een beginwaarde
- De variable heeft een data type
- Een definitie is een soort statement (= een regel waarin iets gebeurt)

```
int dozijn = 12;
```

```
const double pi = 3.14159265359;
```

```
// These constants won't change:
const int analogPin = A0;    // pin that the sensor is attached to
const int ledPin = 13;       // pin that the LED is attached to
const int threshold = 400;    // an arbitrary threshold level that's
```

```
void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize serial communications:
  Serial.begin(9600);
}
```

# Definitie van de setup functie

- De setup functie wordt bij het opstarten een keer gedaan

```
void setup() { /* ... */ }
```

- Niet te verwarren met een functie declaratie:

```
void setup();
```

```
void loop() {  
    // read the value of the potentiometer:  
    int analogValue = analogRead(analogPin);  
  
    // if the analog value is high enough, turn on the LED:  
    if (analogValue > threshold) {  
        digitalWrite(ledPin, HIGH);  
    }  
    else {  
        digitalWrite(ledPin, LOW);  
    }  
  
    // print the analog value:  
    Serial.println(analogValue);  
    delay(1);           // delay in between reads for stability  
}
```

```
void loop() {  
  // read the value of the potentiometer:  
  int analogValue = analogRead(analogPin);  
  
  // if the analog value is high enough, turn on the LED:  
  if (analogValue > threshold) {  
    digitalWrite(ledPin, HIGH);  
  }  
  else {  
    digitalWrite(ledPin, LOW);  
  }  
  
  // print the analog value:  
  Serial.println(analogValue);  
  delay(1);          // delay in between reads for stability  
}
```

# Definitie van de loop functie

- De loop functie wordt na de setup oneindig vaak uitgevoerd

```
void loop() { /* ... */ }
```

- Niet te verwarren met een functie declaratie:

```
void loop();
```

```
void loop() {  
  // read the value of the potentiometer:  
  int analogValue = analogRead(analogPin);  
  
  // if the analog value is high enough, turn on the LED:  
  if (analogValue > threshold) {  
    digitalWrite(ledPin, HIGH);  
  }  
  else {  
    digitalWrite(ledPin, LOW);  
  }  
  
  // print the analog value:  
  Serial.println(analogValue);  
  delay(1);          // delay in between reads for stability  
}
```

# If statement

- Doe iets als iets waar is
- 'else' is niet verplicht

```
if (/* iets */) {  
    //iets is waar, doe dan dit  
}  
else {  
    //doe anders dit  
}
```



```
void loop() {  
  // read the value of the potentiometer:  
  int analogValue = analogRead(analogPin);  
  
  // if the analog value is high enough, turn on the LED:  
  if (analogValue > threshold) {  
    digitalWrite(ledPin, HIGH);  
  }  
  else {  
    digitalWrite(ledPin, LOW);  
  }  
  
  // print the analog value:  
  Serial.println(analogValue);  
  delay(1);          // delay in between reads for stability  
}
```

# Conditie

- iets dat waar of onwaar is

```
if (a == b) { /* a is gelijk aan b */ }
```

```
if (a != b) { /* a is ongelijk aan b */ }
```

```
if (a < b) { /* a is kleiner dan b */ }
```

```
if (a > b) { /* a is groter dan b */ }
```

```
void loop() {  
  // read the value of the potentiometer:  
  int analogValue = analogRead(analogPin);  
  
  // if the analog value is high enough, turn on the LED:  
  if (analogValue > threshold) {  
    digitalWrite(ledPin, HIGH);  
  }  
  else {  
    digitalWrite(ledPin, LOW);  
  }  
  
  // print the analog value:  
  Serial.println(analogValue);  
  delay(1);          // delay in between reads for stability  
}
```

# Functie aanroep

- De functie 'analogRead' word aangeroepen
  - Argumenten: analogPin (== A0)
  - Return type: int
- Het return type hoeft niet gebruikt te worden (maar dat zou in dit geval raar zijn)
- Een functie aanroep is een soort statement (= een regel waarin iets gebeurt)

```
analogRead(analogPin);
```

```
const int x = analogRead(A0);
```