

Syntaxblad Arduino C++

(dit blad wordt op de toets erbij gegeven)

Commentaar

```
// commentaarregel  
/* commentaarblok */
```

Variabelen en datatypen

```
datatype naam = waarde;  
Voorbeelden:  
int i = 0;  
bool isAan = false;  
float f = 5.1;  
String melding = "Fout!";
```

Arrays

```
datatype naam[lengte]  
Voorbeeld:  
int mijnIntegers[6];  
mijnIntegers[0] = 15;  
  
datatype naam[] = {waarden}  
Voorbeeld:  
int mijnPinnen = {3, 5, 6, 13};  
Serial.println(mijnPinnen[1]);  
// output is 5
```

Operators

```
rekenkundig: +, -, *, /  
vergelijking: ==, !=, <, >, <=, >=  
logisch: || "or", && "and", ! "not"
```

Selectie, herhaling

```
if (logische expressie) {  
    // uitvoeren als logische  
    // expressie waar is  
}  
else {  
    // else is optioneel.  
    // uitvoeren als logische  
    // expressie onwaar is  
}  
  
for (start; logische expressie; stap) {  
}  
  
Voorbeeld:  
for (int i = 0; i < 10; i++) {  
    // dit blok wordt 10 keer uitgevoerd  
}  
  
while (logische expressie) {  
    // uitvoeren zolang logische  
    // expressie waar is  
}
```

Functies

```
terugkeerwaarde naam(parameters) {}
```

Voorbeeld:

```
int geefKwadraat(int x) {  
    return x * x;  
}
```

Voorbeeld zonder terugkeerwaarde:

```
void printKwadraat (int x) {  
    Serial.println(x * x);  
}
```

Arduino functies

```
Serial.print(waarde of string)  
Stuurt een waarde tekst naar de seriële  
port van de aangesloten computer
```

```
Serial.println(waarde of string)  
Als Serial.print(), maar dan met  
een regeleinde
```

```
digitalWrite(pin, waarde: HIGH of LOW)  
zet de gegeven pin op de gegeven waarde
```

```
digitalRead(pin)  
geeft de waarde van de gegeven digitale  
pin, HIGH of LOW
```

```
analogWrite(pin, waarde)  
zet een PWM signaal op de gegeven  
digitale pin. De pin moet PWM  
aankunnen. Waarde is een getal  
tussen 0 en 255
```

```
analogRead(pin)  
geeft waarde van genoemde analoge pin  
in getal tussen 0 en 1023
```

```
pinMode(pin, waarde: INPUT of OUTPUT)  
Stelt de gegeven pin in als INPUT  
of OUTPUT
```

```
delay(waarde)  
Pauzeert de uitvoering van het  
programma met waarde milliseconden
```

```
millis()  
Geeft het aantal milliseconden sinds de  
microcontroller aanstaat
```