

Arduino Cheat Sheet

Aerospace Lab Herrenberg

Mehr Informationen auf <http://arduino.cc/en/Reference/extended>



Struktur

```
void setup() {...} void loop() {...}
```

Kontrollstrukturen

```
if(x < 5) {...} else {...}
for(int i=0; i < 255; i++) {...}
while(x < 5) {...}
return x; //Oder return; für void
```

Weitere Syntax

```
// Einzeiliges Kommentar
/* Mehrzeiliges Kommentar */
#define DUTZENT 12;
```

Grundlegende Operatoren

```
= (Zuweisungsoperator)
+ (Addition) - (Subtraktion)
* (Multiplikation) / (Division)
% (Modulo)
== (Ist Gleich) != (Ist nicht gleich)
< (Kleiner als) > (Größer als)
<= (Kleiner gleich)
>= (Größer gleich)
&& (Und) || (Oder) ! (Nicht)
```

Verbindungsoperatoren

```
++ (Erhöhen) -- (Erniedrigen)
+= (Verbindungsaddition)
-= (Verbindungssubtraktion)
*= (Verbindungsmultiplikation)
/* (Verbindungsdivision)
```

Konstanten

```
HIGH | LOW
INPUT | OUTPUT
true | false
142 // Dezimalzahl
0b11011111 // Binär
0x7F // Hexadezimalzahl
7U // Gezwungenes unsigned
10L // Gezwungenes long
10.0 // Gezwungene Kommazahl
2.4e5 // 240000
```

Datentypen

```
void
boolean // 0, 1, true oder false
char // z.B. 'a', -128 bis 127
unsigned char // 0 bis 255
byte // 0 bis 255
int // -32768 bis 32767
unsigned int // 0 bis 65535
long // -2147483648 bis 2147483647
float // -3.4028235E+38 bis 3.4028235E+38
double // das gleiche wie float
```

Strings

```
char S1[15];
char S2[8] = {'a','r','d','u','i','n','o'};
char S3[] = "arduino";
char S4[8] = "arduino";
char S5[15] = "arduino";
```

Arrays

```
int myInts[6];
int myPins[] = {2,4,5,9};
int myValues[3] = {-4,122,8};
```

Typumwandlungen

```
char()    byte()
int()     long()
float()   double()
```

Kennzeichner

```
static // bleibt zwischen Aufrufen bestehen
volatile // Nutze den RAM
const // nur Lesezugriff
```

Digitaler I/O

```
pinMode(pin, [INPUT,OUTPUT])
digitalWrite(pin, wert)
int digitalRead(pin)
```

Analoger I/O

```
int analogRead(pin)
analogWrite(pin, value) // PWM
```

Fortgeschrittener I/O

```
tone(pin, freqHz)
tone(pin, freqHz, dauerMs)
noTone(pin)
shiftOut(datPian, clockPin, [MSBFIRST, LSBFIRST], value)
```

Zeit

```
unsigned long millis() // 50 Tage
unsigned long micros() // 7 Min.
delay(ms)
delayMicroseconds(us)
```

Mathe

```
min(x,y) max(x,y) abs(x)
constrain(x, minVal, maxVal)
map(x, vonL, vonH, bisL, bisH)
pow(zahl, exponent) sqrt(x)
sin(rad) cos(rad) tan(rad)
```

Zufallszahlen

```
randomSeed(seed) // long od. int
long random(max)
long random(min, max)
```

External Interrupts

```
attachInterrupt(interrupt, function, [LOW, CHANGE, RISING, FALLING])
detachInterrupt(interrupt)
interrupts()
noInterrupts()
```

Bibliotheken:

Serial.

```
begin([300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200])
end()
int available()
int read()
flush()
print()
println()
write()
```

Servo. (#include <Servo.h>

```
attach(pin)
write(winkel) // 0 bis 180 Grad
read() // 0 bis 180
attached() // Wahrheitswert
detach()
```

LCD. (#include <LiquidCrystal.h>

```
begin(reihen, spalten)
print(wert)
setCursor(reihe, spalte)
```

	ATMega 168	ATMega 328	ATMega 1280
Flash	16kB	32kB	128kB
SRAM	1kB	2kB	8kB
EEPROM	512B	1kB	4kB

	Duemilanove/ Nano/ Pro / ProMini	Mega
# of IO	14 + 6 Analog (Nano hat 14 + 8)	54 + 18 Analog
Serielle Pins	0 - RX 1 - TX	0 - RX1 19 - RX2 17 - RX3 15 - RX4 1 - TX1 18 - TX2 16 - TX3 14 - TX4
Ext. Interrupts	2 - (Int 0) 1 - (Int 1)	2, 3, 21, 20, 19, 18
PWM	5, 6 - Timer 0 9, 10 - Timer 1 3, 11 - Timer 2	0 - 13
SPI	10 - SS 11 - MOSI 12 - MISO 13 - SCK	53 - SS 51 - MOSI 50 - MISO 52 - SCK
I2C	A4 - SDA A5 - SCK	20 - SDA 21 - SCK