

Initialization

```
pinMode();
pinMode(pin, 1);
pinMode(pin, OUTPUT);
pinMode(pin, 0);
pinMode(pin, INPUT);
```

Digital Read/Write

```
digitalWrite(pin, 1);
digitalWrite(pin, HIGH);
digitalWrite(pin, 0);
digitalWrite(pin, LOW);
digitalRead(pin);
```

Analog Read/Write

```
analogRead(sensorPin);
analogWrite(sensorPin);
```

#define

```
#define LEDL3 6 //Kein "=" oder ";" am
schluss!
#define LED3ON digitalWrite(L3,HIGH)
//Led3 einschalten
```

Servo

```
#include <Servo.h>
Servo myservo;
myservo.attach(<Pin>); //setup
myservo.write(<0 - 180>);
```

Serial Monitor

Serial.begin(9600);	im setup() Geschwindigkeit 9600
Serial.print("Text");	Text ohne Zeilensprung
Serial.print(wert);	Ausgabe einer Dezimalzahl
Serial.print(wert,DEC);	Ausgabe einer Dezimalzahl
Serial.print(wert,HEX);	Ausgabe eines Wertes in HE
Serial.print(wert,BIN);	Ausgabe eines Binärwertes
Serial.println();	Anzeige mit Zeilenumbruch

Variable

byte	8 bits
char	8 bits
int	16 bits
float	32 bits

Toggle-Funktion

```
!(digitalRead(L1))
digitalWrite(L1,!digitalRead(L1);
#define Led1Toggle digitalWrite(L1, !digitalRead(L1))
```

Functions

```
delay(Millisekunden);
delayMicroseconds(Mikrosekunden);
tone(pin, frequency, duration);
millis();
random(min, max);
```

Vergleichsoperatoren

>	größer als
<	kleiner als
>=	größer gleich
<=	kleiner gleich
!=	ist ungleich
==	identisch mit
=	Zuweisung Variable

Schreibweisen Variablen

varname	Variable
varName	Variable
Varname	Konstante
VarName	Konstante
VARNAME	Konstante



while Loop

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

For Loop

```
void loop () {
    for (int i=0; i<100;
i++) {
        ...;
    }
}
```

do while Loop

```
void loop () {
    do {
        ...;
    } while(...);
}
```

if Statement

```
void loop () {
    if (...) {
        ...
    }
    else {
        ...
    }
}
```

switch Statement

```
void loop () {
    switch(...) {
        case a:
            ...;
            break;
        case b:
            ...;
            break;
        case c:
            ...;
            break;
        default:
            ...;
            break;
    }
}
```

counting loops

```
void loop() {
    // put your main code here, to
run repeatedly:
    Leds();
    delay(1);
}
void Leds(){
    loopCounterRed--;
    loopCounterYellow--;
    if(loopCounterRed <= 0){
        RedLedToggle;
        loopCounterRed = RedLedFre-
quenz;
    }
    if(loopCounterYellow <= 0){
        YellowLedToggle;
        loopCounterYellow = Yellow-
LedFrequenz;
    }
}
```

millis()-Funktion

```
void setup() {
    startTime = millis(); //
initialisation using actual time
}
void loop() {
    if (millis() - startTime >=
period) {
        startTime = millis();
    }
}
```

