

### Grundlagen

Dieser Spickzettel erklärt ein paar Dinge, die auf jeden Fall da sein müssen oder die man oft brauchen kann.

Befehl	Beschreibung	Beispiel
<pre>void setup() { }</pre>	<p>Innerhalb der geschweiften Klammern { und } kannst Du das Programmieren, was <b>einmal beim Einschalten</b> des Arduinos passieren soll. Hier werden oft Einstellungen vorgenommen, die danach immer gleich bleiben.</p> <p>Englisch: void = Leere / Lücke *) Englisch: setup = Einrichtung</p>	<pre>void setup() {   Serial.begin(9600); }</pre>
<pre>void loop() { }</pre>	<p>Innerhalb der geschweiften Klammern { und } kannst Du das Programmieren, was <b>immer wieder hintereinander</b> passieren soll, nachdem <i>setup</i> fertig ist.</p> <p>Englisch: void = Leere / Lücke *) Englisch: loop = Kringel / Kreislauf / Überschlag</p>	<pre>void loop() {   Serial.println("Hallo");   Serial.flush();   delay(1000); }</pre>
<pre>// Kommentar</pre>	<p>Hinter den zwei schrägen Strichen kannst Du bis zum Ende der Zeile schreiben, was ihr wollt. Es wird einfach ignoriert. Hier kann man z.B. aufschreiben, warum man etwas programmiert hat, oder eine Erklärung abgeben.</p>	<pre>// Warte eine Sekunde delay(1000);</pre>
<pre>/* Kommentar */</pre>	<p>Zwischen /* und */ kannst Du schreiben, was Du willst. Es wird einfach ignoriert. Das kann ein Teil einer Zeile sein aber auch mehrere Zeilen umfassen, darf aber keinen Befehl unterbrechen.</p>	<pre>/* Das Ergebnis war mal 9, jetzt 5 */ Serial.println(2+3/*+4*/);</pre>
<pre>delay(Millisekunden);</pre>	<p>Wartet die angegebene Zeit und tut währenddessen einfach nichts. Damit kannst Du das Programm verlangsamen, was manchmal ziemlich nützlich ist.</p> <p>1000 = 1 Sek, 60000 = 1 Min, 3600000 = 1 Std Englisch: delay = verzögern / verspäten / hinausschieben</p>	<pre>delay(2500);</pre>

\*) beim Programmieren gibt es viele Funktionen, die etwas ausrechnen und das Rechenergebnis zurückgeliefert wird. *void* bedeutet, dass nichts zurückgeliefert wird.