Methoden in C#

Methoden (oder Funktionen in nicht Objekt-Orientierten Sprachen) machen den Hauptteil von Quellcode aus. Alle "Statements" eines Programms müssen in einer Methode stehen(zwischen den "{", "}"). Ausgenommen sind Deklarationen von Klassen, Methoden und Klassenvariablen.

Die beiden Methoden die wir bisher kennen heißen "setup" und "loop".

**// Setup, wird immer einmal zu Beginn eines Screens aufgerufen**

**override public void setup()**

**{**

**// Euer Code**

**}**

**// Loop, wird jeden Frame (60 mal die Sekunde) aufgerufen**

**override public void loop()**

**{**

**// Euer Code**

**}**

Diese sind aber sehr simple Methoden, da sie keine Parameter (die Klammern "()" sind leer) und keinen Rückgabetypen besitzen("void" vor dem Namen der Methode).

Aber Methoden sind mehr als nur Container für unseren Quellcode. Man könnte ein Programm ja nur in einer einzigen Methode schreiben, die das gesamte Programm ausmacht. Aber wenn man Methoden richtig benutzt, machen sie vieles sehr viel leichter.

**Der Allgemeine Aufbau eine Methode-Definition:**Eine Methode besteht aus 4 Bestandteilen:

Den **Bezeichner:** Namen der Methode. Damit ruft man sie später auf. Ist vollkommen beliebig, wie bei Variablenamen.

Die **Parameter**: Beliebig viele Variablen, die mit Typ und Namen angegeben werden. Werden beim Aufruf gesetzt durch das setzen von Werten in die "()" festgelegt.

Den **Programmcode**: Beschreibt was die Methode tut, sobald sie aufgerufen wird.

Den **Rückgabetypen**: Ein Datentyp, der angibt welche Art von Werten die Methode zurückgibt. "void" Bedeutet die Methode gibt nichts zurück.  
Vor dem Rückgabetypen können noch "modifier" stehen, diese werden erst später behandelt. Beispiele: override, public, private, protected, static

Bezeichner

Rückgabetyp

Parameter

**int summe(int zahl1, int zahl2, int zahl3)**

**{**

**int zahl4 = zahl1 + zahl2 + zahl3;**

Programmcode

**return zahl4;**

**}**  
Hierbei ist es wichtig zu verstehen, was "return" macht. Der "return" Befehl gibt einen Wert zurück, der vom Rückgabetyp sein muss.(Ist Rückgabetyp = int, so muss nach return ein Int-Wert stehen)Eine Methode mit dem Rückgabetyp "void" gibt nichts zurück, braucht also auch kein "return".

Wenn eine Methode definiert ist, kann man sie **aufrufen**.

**int x = summe(13,7,20);**

# **// x = 13+7+20 = 40**

Man kann sich eine Methode wie eine klassische Mathematische Funktion vorstellen:

**f(x) = x²**

Hierbei ist x der Parameter,   
f der Name der Funktion und   
x² der Programmcode.

(Es wird kein Rückgabetyp angegeben, da in der Mathematik alle Werte Zahlen sind)

Wenn man jetzt nach f(2) fragt, setzt man für den Parameter x die Zahl 2 ein und führt den Programmcode aus:

**a = f(2) = 2² = 4**In C# könnte man diese Funktion jetzt so programmieren:

**int f(int x)**

**{**

**return x\*x;**

**}**Und aufrufen:

**int a = f(2);**

# **// a = 2\*2 = 4**