Hallo Welt!

== Was brauchen wir? ==
Wir benötigen folgendes:

- Compiler (Z.b. GCC, MVC, ...)
- Lust und Laune
- Einen Nutzbaren PC

== Wie fange ich an? ==
Zu aller erst erstellen wir ein neues Konsolen-Projekt. Nun fügen wir diesem Projekt eine neue Datei hinzu. Diese könnt ihr benennen wie ihr wollt, wichtig ist, dass sie die Endung *.cpp besitzt. Bei MVC darf es auch auf *.c enden, aber bei GCC zum Beispiel nicht, denn GCC nimmt dann an, wir würden ein C-Programm und kein C++-Programm schreiben. Also, auch der ordnungsgemäßen Vorgehensweise, eine *.cpp-Datei.

== Wie geht es weiter? ==
Wir binden zuerst die Datei "iostream" ein. Dies erreichen wir so:
Quellcode

```
1. #include <iostream>
```

Nun erweitern wir den Code um diese Zeile:
Quellcode

```
1. using namespace std;
```

Sie bewirkt, dass wir nicht vor jedem "Begriff" 'std' gefolgt von '::' schreiben müssen.
Nun öffnen wir die "main-Funktion". Es ist erstmal unwichtig für den Anfang mehr über sie zu wissen, als dass man sie immer öffnen sollte, da sie beim starten des Programms ausgeführt wird.
Quellcode

```
1. int main()
2. {
```

Nun unser eigentliches Programm:
Quellcode

```
1. cout << "Hallo Welt!";
```

cout gibt immer dort aus, wo sich der Schreibcursor befindet. Um einen Zeilenumbruch zu starten, müssen wir den Cursor "verschieben". Dazu später mehr.
cout ist Teil des std-Namespaces. Bedeutet, wenn die Zeile "using namespace std;" fehlen würde, müsse man das hinschreiben:
Quellcode

```
1. std::cout << "Hallo Welt!";
```

Ihr seht, man spart sich Schreibarbeit damit.
Fahren wir fort, der Operator "<<" bewirkt, dass genau das, was dahinter steht zu cout geschickt wird. In einfachen Worten ausgedrückt, um jetzt niemanden zu verwirren.
Das umgekehrte tut der ">>" Operator, aber das kommt später.

Wir fügen noch folgendes ein:

Quellcode

```
1. system("PAUSE"); //Sieht erstmal unschön aus, später wird gezeigt wie man Ordnung rein bringt.
2. return (0); //In dem Fall kann man die Klammern auch weglassen
```

Okay, alles in allem schließen wir ab mit:

Quellcode

```
1. }
```

Fertig! F5 Drücken, kompilieren im release/debug-Modus und ihr seht euren Erfolg. (Gilt nur in MVC/MVCE, bei anderen einfach normal Debuggen. Meistens durch einen Pfeil symbolisiert)

== Nochmal alles in allem etwas umfangreicher: ==
Hier seht ihr alles erwähnte noch einmal nur etwas umfangreicher gestaltet.

Quellcode

```
1. #include <iostream>
2. using namespace std;
3. int main()
4. {
5. cout << "Ha";
6. cout << "llo";
7. cout << " ";
8. cout << "Welt";
9. cout << "!" << endl; //endl versetzt den Cursor in die nächste Zeile und "leert" den Buffer.
10. cout << endl << "Tach!";
11. cout << endl; //Wir wollen nicht, dass der Satz von system("PAUSE") gleich dahinter steht.
12. system("PAUSE");
13. return(0);
14. }
```

Alles anzeigen

== Schlusswort ==
Alles in einem könnt ihr doch jetzt schon schöne Sachen machen. Ihr könnt Texte ausgeben lassen und diese auch Ordnen.
Zumal könnt ihr nun auch auf irgendeinen Tastendruck des Benutzers warten und dann das Programm via return 0 beenden.
Zu return:
return sendet einen Wert aus einer Funktion zu der Funktion, die die Funktion gerade nutzt. Das jetzt auch etwas vereinfacht ausgedrückt, aber mehr müsst ihr erstmal nicht wissen.
Es reicht zu wissen, dass wenn man in der Funktion main den Wert "0" sendet, dass sich das Programm schließt.

MfG
Check