

Cheat Sheet für C-Anfänger¹

Ein Zeichen als Eingabe / Ausgabe

```
#include <stdio.h>
int main() {
    int c;
    printf("Geben Sie ein Zeichen ein:");
    c = getchar();

    printf("\n Ihre Eingabe: ");
    putchar(c);

    return 0;
}
```

Zeilen-
umbruch

Die Funktionen `printf`,
`getchar` und `putchar`
verfügbar machen

Ein Zeichen von der
Tastatur lesen, ASCII-Wert
als Ergebnis

Das Zeichen für einen
ASCII-Wert auf dem
Bildschirm ausgeben

Formatierte Ausgabe

```
#include <stdio.h>
int main() {
    int zeichen;
    zeichen = printf("Hallo Welt");
    printf("%d Zeichen ausgegeben\n", zeichen);
    return 0;
}
```

Die Funktion `printf`
verfügbar machen

`printf` gibt die Anzahl der
ausgedruckten Zeichen
zurück

Platzhalter für dezimalen (`%d`) Wert
der Variablen `zeichen`

Formatierte Eingabe

```
#include <stdio.h>
int main(void) {
    char a;
    int b, check;

    printf("Bitte Eingabe (Zeichen/Zahl): ");
    check = scanf("%c %d",&a, &b);
    printf("check = %d \n",check);
    return 0;
}
```

Die Funktionen `printf` und
`scanf` verfügbar machen

Ein Zeichen (`%c`) und eine
Dezimalzahl (`%d`) einlesen und
in den Variablen `a` und `b`
abspeichern. Rückgabewert ist
die Anzahl der gelesenen
Werte, in `check` abgelegt.

¹ Quelle und weitere Beispiele: http://openbook.rheinwerk-verlag.de/c_von_a_bis_z/

Makefile erstellen

Das Programm `make` ist auf Linux- bzw. macOS X-Systemen typischerweise bereits verfügbar.

Es sucht im aktuellen Verzeichnis nach einer Datei mit dem Namen `Makefile` und interpretiert dessen Inhalt auf folgende Art und Weise:

```
helloworld: hello.c
    gcc -o helloworld hello.c

clean:
    rm helloworld
```

Das Arbeitsziel `helloworld`. Wird ausgewählt, wenn `make` ohne Argumente oder mit `make helloworld` aufgerufen wird.

Arbeitsziel hängt von der Datei `hello.c` ab.

Dieses Kommando wird im Terminal ausgeführt, wenn sich die Abhängigkeiten (also hier `hello.c`) seit der letzten Ausführung von `make` geändert haben. Einrückung mit TAB-Taste.

Arbeitsziel `clean` ohne Abhängigkeiten. Wird ausgewählt, wenn `make clean` aufgerufen wird.

Dieses Kommando wird im Terminal ausgeführt, wenn dieses Arbeitsziel ausgewählt wurde.

C-Programm kompilieren

Das Programm `gcc` ist auf Linux- bzw. macOS X-Systemen typischerweise bereits verfügbar. Es übersetzt ein oder mehrere C-Quelltextdateien in ein ausführbares Programm.

```
gcc -o helloworld hello.c
```

Eingabedatei mit C-Quelltext.

Programm `gcc` ausführen.

Nächstes Argument ist der gewünschte Name der Ausgabedatei.

Gewünschter Name der Ausgabedatei.

Weitere interessante Optionen für den `gcc`:

- `-Wall`: Anzeige von allen Warnungen bzgl. der Korrektheit des Quelltextes.
- `-g`: Ausführbares Programm erzeugen, dass mit einem Debugger untersucht werden kann.
- `-std=c99`: Neuere Version der Programmiersprache C unterstützen (siehe Skript).

Der GCC unterstützt die Übersetzung mehrerer Quelltextdateien zu einem Programm:

```
gcc -o helloworld main.c funktionen.c
```

Argumente von der Kommandozeile entgegennehmen

Die Funktion `printf`
verfügbar machen

Alternative Signatur für die `main()` Funktion, um
Kommandozeilenargumente übergeben zu bekommen

```
#include <stdio.h>
int main(int argc, char** argv) {
    printf("%d Kommandozeilenargumente\n", argc);
    printf("Name des Programms: %s\n", argv[0]);
    printf("Erstes Argument: %s\n", argv[1]);
    return 0;
}
```