C Handbuch

Inhaltsverzeichnis:

Hello World

```c
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

int main() die **main**() Funktion ist der Eintrittspunkt zu einem Programm. **#include <stdio.h>** die Funktion wird benutzt um den Output zu generieren, dieser ist mit **stdio.h** definiert. Geschweifte Klammern markieren den Anfang und das Ende einer Funktion.

Daten Typen

```c
#include <stdio.h>

int main() {
    printf("int: %ld \n", sizeof(int));
    printf("float: %ld \n", sizeof(float));
    printf("double: %ld \n", sizeof(double));
    printf("char: %ld \n", sizeof(char));

    return 0;
}
```

Int: Heißt ,,Integer“ also Ganzzahl

Float: Heißt ,,Floating Point“ ist eine Zahl mit fraktionierten Teil.

Double: Heißt ,,Double-Precision“ hat die doppelte Präzision von float.

Char: Bedeutet ,,Single Character“ also alleinstehender Charakter.

sizeof legt/zeigt die benötigte Speichermenge für einen bestimmten Daten Typen.

Das **printf** statements hat **zwei Argumente**. Das erste ist der output string mit dem **format specifier** (%ld), während das nächste die **sizeof** Menge wiedergibt. Im finalen output, wird das **%ld** (for long decimal) durch den Wert vom zweiten Argument ersetzt.

```c
#include <stdio.h>

int main() {
    int a, b;
    float salary = 56.23;
    char letter = 'Z';
    a = 8;
    b = 34;
    int c = a+b;

    printf("%d \n", c);
    printf("%f \n", salary);
    printf("%c \n", letter);

    return 0;
}
```

Input & Output

```c
#include <stdio.h>

int main() {
    char a = getchar();

    printf("You entered: %c", a);

    return 0;
}
```

getchar() sagt C, dass es einen Input öffnen soll und diesen ,,speichert“. Dieser kann dann z.B. im printf statement wiedergegeben werden.

```c
#include <stdio.h>

int main() {
    char a[100];

    gets(a);

    printf("You entered: %s", a);

    return 0;
}
```

Die **gets()** Funktion wird genutzt, um inputs in geordneter Sequenz von ,,Charakteren“. Diese werden auch **string** genannt.

```c
#include <stdio.h>

int main() {
    int a, b;
    printf("Enter two numbers:");
    scanf("%d %d", &a, &b);

    printf("\nSum: %d", a+b);

    return 0;
}
```

Das **&** Zeichen vor der Variable ist der **address operator**. Er gibt die Adresse, im Speicher einer Variablen an. Das wird benötigt weil **scanf** einen Inputwert an einer Variablen Adresse platziert.

Wichtig: **scanf()** hört auf zu lesen sobald es auf ein Leerzeichen trifft. Also ist "Hello World" für **scanf()** zwei verschieden Inputs.

```c
#include <stdio.h>

int main() {
  char a = getchar();

  printf("You entered: ");
  putchar(a);

  return 0;
}
```

putchar sagt C, dass es den ersten ,,character“ von der Variablen a auslesen soll. printf mit putchar kombiniert sagt C, dass es den ersten ,,character“ vom input in den output schreiben soll.

```c
#include <stdio.h>

int main() {
  char a[100];

  gets(a);

  printf("You entered: ");
  puts(a);

  return 0;
}
```

puts() wird genutzt um ein ,,string” als Display output anzuzeigen.
Ein string ist im char array gespeichert.

```c
#include <stdio.h>

int main() {
    int x;
    float num;
    char text[20];
    scanf("%d %f %s", &x, &num, text);
}
```

Schreib 10 22.5 abcd und dann Enter drücken weißt 10 zu x zu, 22.5 zu num und abcd zu text.
& muss benutzt werden um die Variablen Adresse zu erreichen. Das **&** wird bei einem string nicht gebraucht da er sich wie ein pointer verhält.

```c
#include <stdio.h>

int main() {
    int x, y;
    char text[10];

    scanf("%2d %d %*f %5s", &x, &y, text);
    /* input: 1234  5.7  elephant */
    printf("%d  %d  %s", x, y, text);
    /* output: 12  34  eleph */

    return 0;
}
```

Wichtig: **d** decimal

c character

s string

f float

e scientific notation

x hexadecimal