**C-Checklist**

**Grundlagen**:

* Pointer
* Arrays (+Strings)
* Dynamischer Speicher
* Kommandozeilenargumente/Parameterübergabe
* Ein-und-Ausgabe von Dateien (Bsp.: Beschreiben von Config-Datei)
* Makefiles
* Headerdateien
* Strukturen
* Konstanten und globale Variablen

**Interprozesskommunikation**:

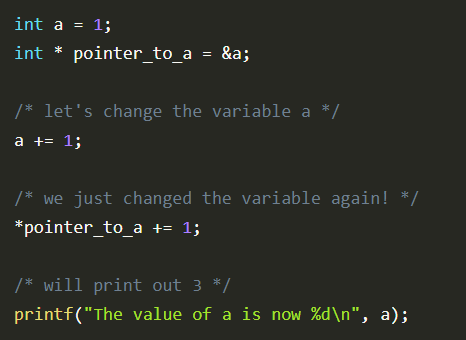
* Erzeugen von Prozessen (fork, exec)
* Shared Memory
* Semaphoren
* Signale (kill)
* Sockets
* Pipes

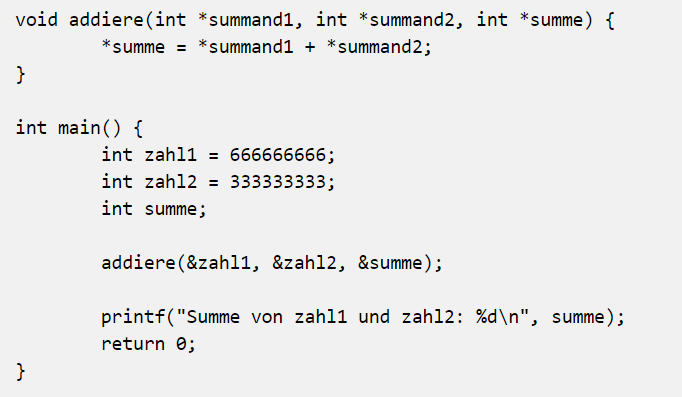
**Grundlagen**

**Pointer**

Erstellen: <Datentyp> name = <wert> ;

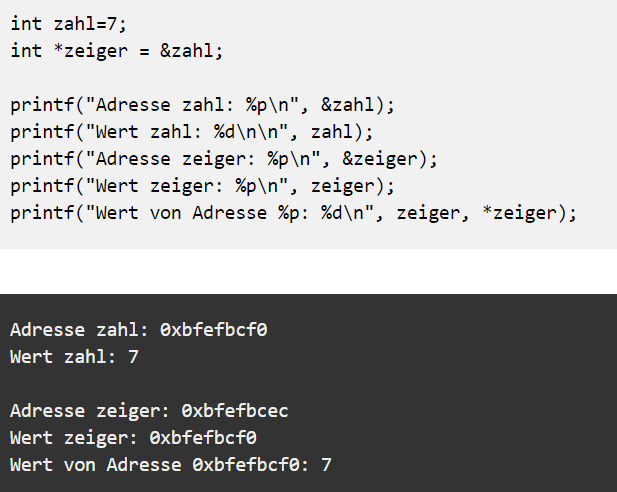
Referenzieren: name = &<Referenz> ;

Beispiele:

Addieren

Mittels

Pointer



Ausdrucken

Von Werten

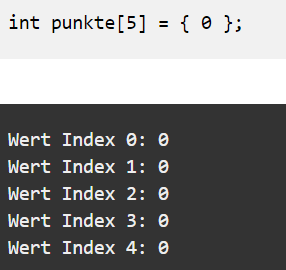
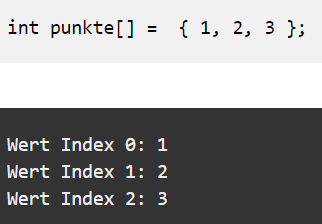
**Arrays und Strings**

Erstellen: <Datentyp> Name[Größe];

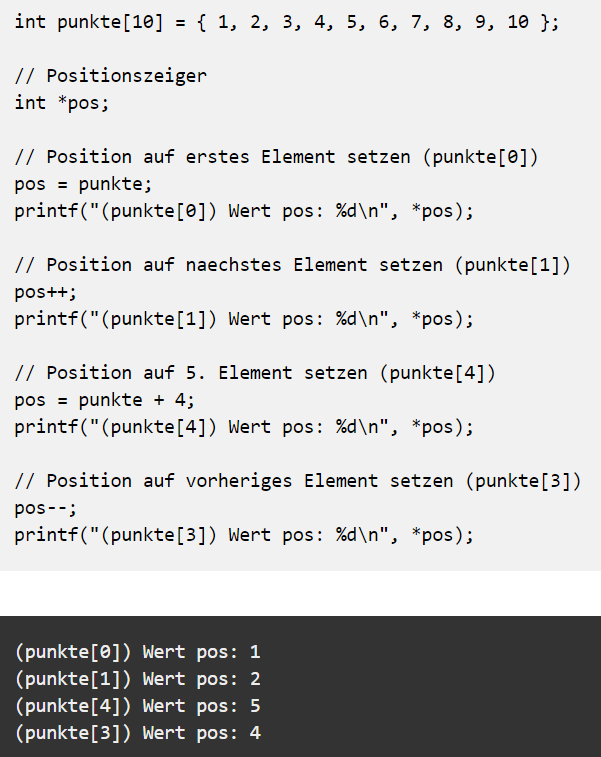
Char\* string[Länge];

Initialisieren: Name[Größe] = {wert1, wert2, wert3…}

Char\* string = „Text“;

Beispiele:

Initialisierung

Zeigerarithmetik

Mit Arrays

**Dynamischer Speicher**

Erstellen: void \*malloc(size\_t size);

Rückgabe: void-Zeiger auf Anfang des Speicherbereichs

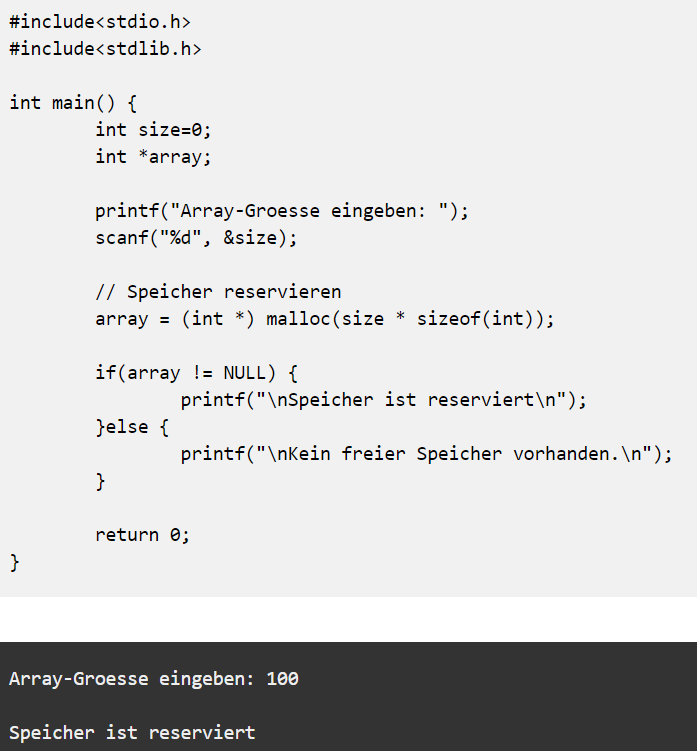
Null-Zeiger, falls kein freier

Alternative: void \*calloc(size\_t n, size\_t size);

Unterschied: calloc initialisiert den Speicher mit 0

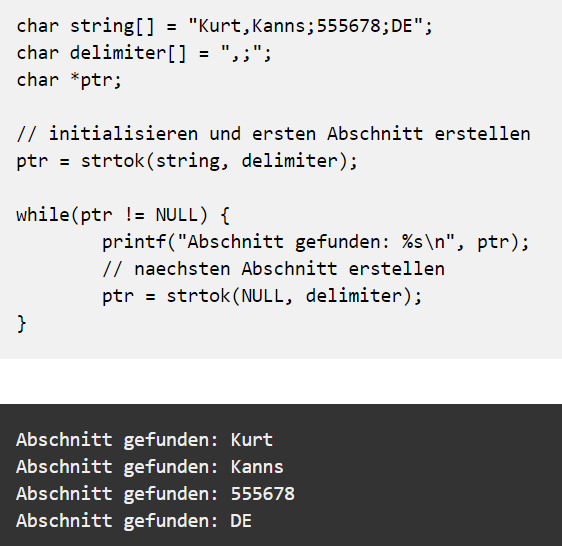
n gibt die Anzahl der Elemente an

Freigeben: free(speicher);

Beispiel:

**Wichtigsten String-Funktionen**:

* **int strlen(char \*string):** Berechnet die Länge des strings
* **char \* strcpy(char \*dest, char \*src):** Kopiert Zeichenkette in src nach dest
* **char \* strcat(char \*dest, char \*src**): Verkettet zwei strings und speichert sie in dest
* **int strcmp(char \*str1, char \*str2):** Vergleich Zeichen miteinander
  + 0 -> Strings sind gleich
  + >0 das erste ungleiche Zeichen in str1 ist größer als in str2
  + <0 das erste ungleiche Zeichen in str1 ist kleiner als in str2
* **Int strcmp(char \*str1, char \*str2, int n):** Vergleicht n Zeichen miteinander
* **Char \* strstr(char \*string, char \*needle):** Schaut ob needle im string vorkommt
  + Rückgabe ist die Adresse des vorkommenden Strings
  + Oder NULL falls er nicht darin vorkommt
* char \* strtok(char \*string, char \*delimiters): Zerlegt den string anhand delimiters

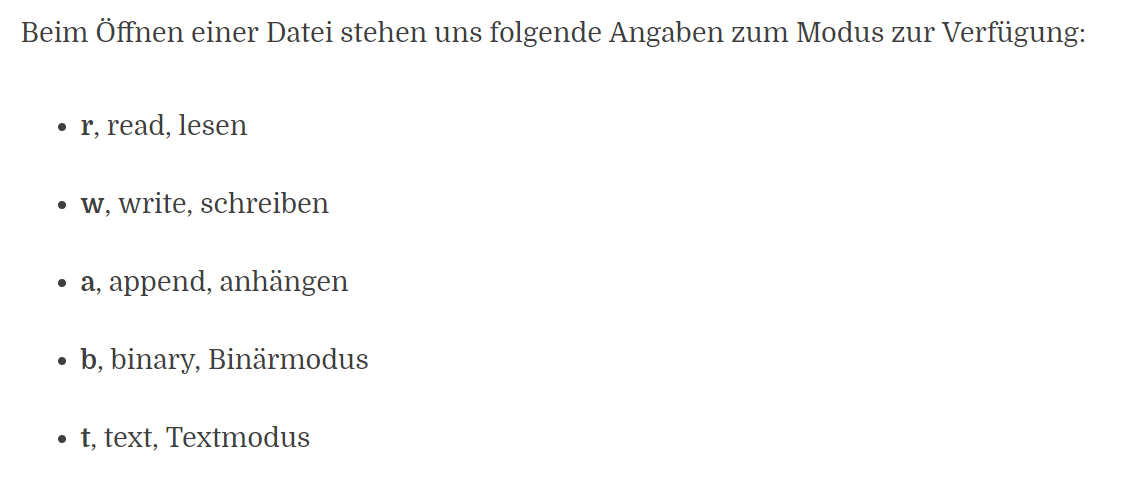


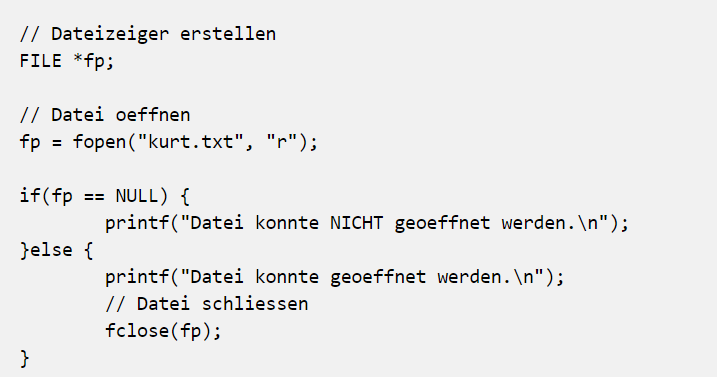
**Typumwandlungen**:

* int atoi(const char \*string); string -> int
* double atof(const char \*string); string -> double
* long atol(const char \*string); string -> long

**Kommandozeilenargumente**: Siehe hier <http://openbook.rheinwerk-verlag.de/c_von_a_bis_z/013_c_kommandozeilenargumente_001.htm#mjf6a9e42fa3bc25b2b1e06270e7bf8f68>

**Ein-und-Auslesen von Dateien**

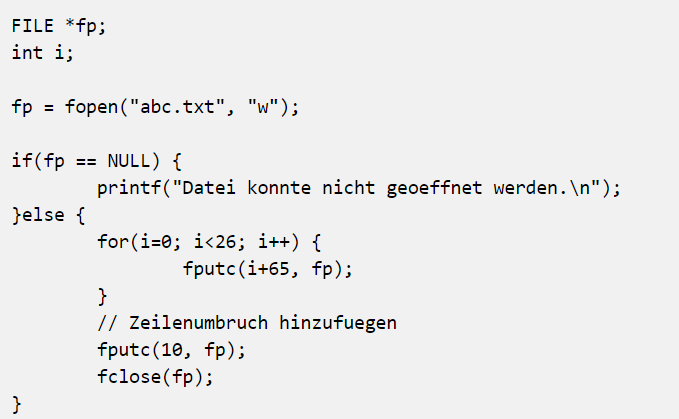
Öffnen und schließen: **FILE \* fopen(const char \*filename, const char \*mode);**

Beispiel:

Schreiben: **int fputc(int c, FILE \*fp);**

Lesen: **int fgetc(FILE \*fp);**

Beispiele:





Schreiben formatiert: **int fprintf(FILE \*fp, const char \*format [, arguments]);**

Lesen formatiert: **int fscanf(FILE \*fp, char \*format [, arguments]);**

Beispiele:

