

Einführung in C

April 29, 2025

Übersicht

1. Was ist C?
2. Was sind Pointer?
3. Was sind Structs?
4. Wie lese ich Argumente ein?
5. Wie verwendet man .h Dateien?

Was ist C?

- Imperative, prozedurale Sprache
- Entwickelt in den 1970er Jahren (Dennis Ritchie)
- Grundlage vieler moderner Sprachen (C++, Rust, Go)
- Direkter Speicherzugriff möglich
- Häufig in Systemprogrammierung verwendet

Pointer – Was ist das?

- Ein Pointer speichert eine Speicheradresse

```
1 int x = 10;  
2 int *p = &x; // p zeigt auf x
```

- Zugriff mit *p (Dereferenzierung)

Pointer – Verwendung

```
1  #include <stdio.h>
2
3  void verdoppeln(int *z) {
4      *z = *z * 2;
5  }
6
7  int main() {
8      int a = 5;
9      verdoppeln(&a);
10     printf("%d\n", a); // Ausgabe: 10
11     return 0;
12 }
```

Structs – Benutzerdefinierte Typen

```
1 struct Person {  
2     char name[50];  
3     int alter;  
4 };  
5  
6 struct Person p1 = {"Max", 25};
```

- Nützlich für komplexe Datenstrukturen

Structs – Verwendung

```
1 #include <stdio.h>
2
3 struct Person {
4     char name[50];
5     int alter;
6
7 void druckePerson(struct Person p) {
8     printf("Name: %s, Alter: %d\n", p.name, p.alter);
9 }
```

```
1 int main() {
2     struct Person p1 = {"Anna", 30};
3     druckePerson(p1);
4     return 0;
5 }
```

Kommandozeilenargumente einlesen

```
1 #include <stdio.h>
2
3 int main(int argc, char *argv[]) {
4     if (argc > 1) {
5         printf("Argument 1: %s\n", argv[1]);
6     } else {
7         printf("Kein Argument angegeben\n");
8     }
9     return 0;
10 }
```


Header-Dateien in C

datei.h:

```
1 void sagHallo();
```

datei.c:

```
1 #include <stdio.h>
2 #include "datei.h"
3
4 void sagHallo() {
5     printf("Hallo!\n");
6 }
```

Zusammenfassung

- Pointer: Zugriff auf Speicher, Call-by-Reference
- Structs: Eigene Datentypen
- Kommandozeilenargumente: `argc`, `argv[]`
- Header-Dateien: Trennung von Deklaration und Implementierung