

# Funktionen in C

## Erklären Sie die Ausgabe - 1

```
#include <stdio.h>
#include <stdlib.h>
```

```
int x = 1;
```

```
int main()
{
```

```
int x = 2;
```

```
printf("%d ", x);
```

```
int x = 3;
```

```
printf("%d ", x);
```

```
getchar();
return 0;
```

```
}
```

### Ausgabe:

```
fdemo01.c: In function 'main':
fdemo01.c:14:7: error: redefinition of 'x'
fdemo01.c:9:7: note: previous definition of 'x' was here
```



**Keine doppelte Bezeichner  
im gleichen Gültigkeitsbereich  
möglich → **Eindeutigkeit verletzt!****



**Darf ich einen Bezeichner nie ein  
zweites Mal im QC verwenden?**

## Erklären Sie die Ausgabe - 2

```

int x = 1; ①

int main()
{
    int x = 2; ②
    printf("%d ", x++);
    {
        int x = 3; ③
        printf("%d ", x++);
    }
    printf("%d ", x++);
    getchar();
    return 0;
}

```

**Ausgabe:** 2 3 3

**global** Speicherstelle ①  
 zugreifbar sofern  
 nicht überschrieben

**global innerhalb main()**

② überschreibt ①

**lokal innerhalb { }**

③ überschreibt ②

③ Ist außerhalb der { }  
 nicht bekannt

## Sichtbarkeit und Lebensdauer

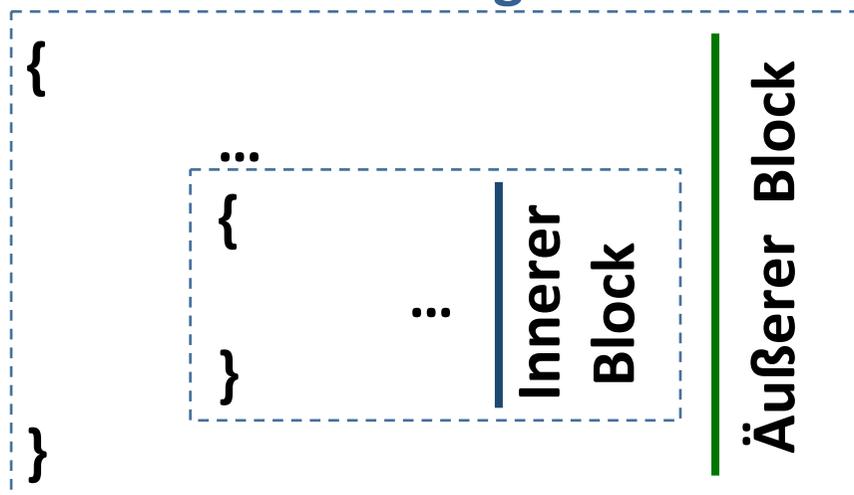
### Wo können Variablen deklariert werden?

Außerhalb von Funktionen

Innerhalb von Funktionen

Innerhalb von Blöcken → definiert durch `{ ... }`

### Blockschachtelung



Lokal definierte Variablen sind nur innerhalb ihres Blockes sichtbar.

## Sichtbarkeit und Lebensdauer

### Sichtbarkeit von Variablen

Variablen in inneren Blöcken sind nach außen nicht sichtbar.

Globale Variablen und Variablen in äußeren Blöcken sind in inneren Blöcken sichtbar.

Lokale Variablen verdecken in ihrem Bereich globale Variablen mit gleichem Bezeichner.

### Lebensdauer von Variablen

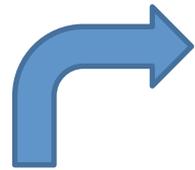
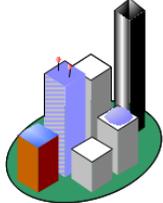
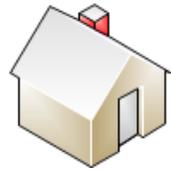
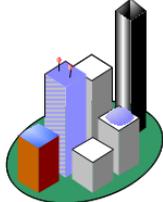
Globale Variablen existieren bis zum Ende des Programms.

Lokale Variablen werden zu Beginn des Blocks erzeugt und mit Ende des Blocks verworfen.

# Sichtbarkeit und Lebensdauer

Wer sieht wen?

Bei unterschiedlichen Bezeichnern

			
sieht	global	äußerer Block	innerer Block
			
			
			

## Berechnung einer Fakultät

```
int main()
{
    int eingabe = 0, fakultaet = 1;

    printf("Eingabe: ");
    scanf("%d", &eingabe);

    {
        int f = 1, n = eingabe;
        while( n > 1 ){
            f = f * n--;
        }
        fakultaet = f;
    }
    printf("Fakultaet: %d\n", fakultaet );

    getchar();
    return 0;
}
```

eingabe wird übernommen

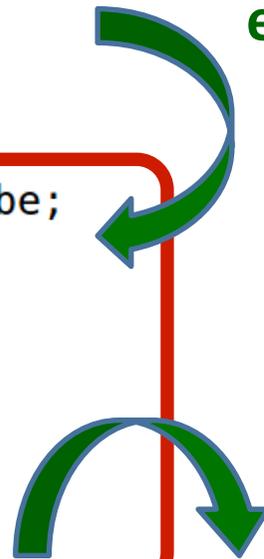
Innerer Block erzeugt

lokales int f

lokales int n

Ende innerer Block

Speicherplatz von  
f und n wird wieder  
freigegeben.



## Auslagern des inneren Blockes

```
int getFakultaet(int); // Prototyp
```

```
int main()
{
    int eingabe = 0;

    printf("Eingabe: ");
    scanf("%d", &eingabe);

    printf("Fakultaet: %d\n", getFakultaet(eingabe));

    getchar();
    return 0;
}
```

```
int getFakultaet(int n){
```

```
    int f = 1;
    while( n > 1 ){
        f = f * n--;
    }
    return f;
}
```

**Grundprinzip einer Funktion**

eingabe wird kopiert

int n = eingabe;

Wert von f wird kopiert

← Ende Hauptprogramm

**Innerer Block erzeugt**

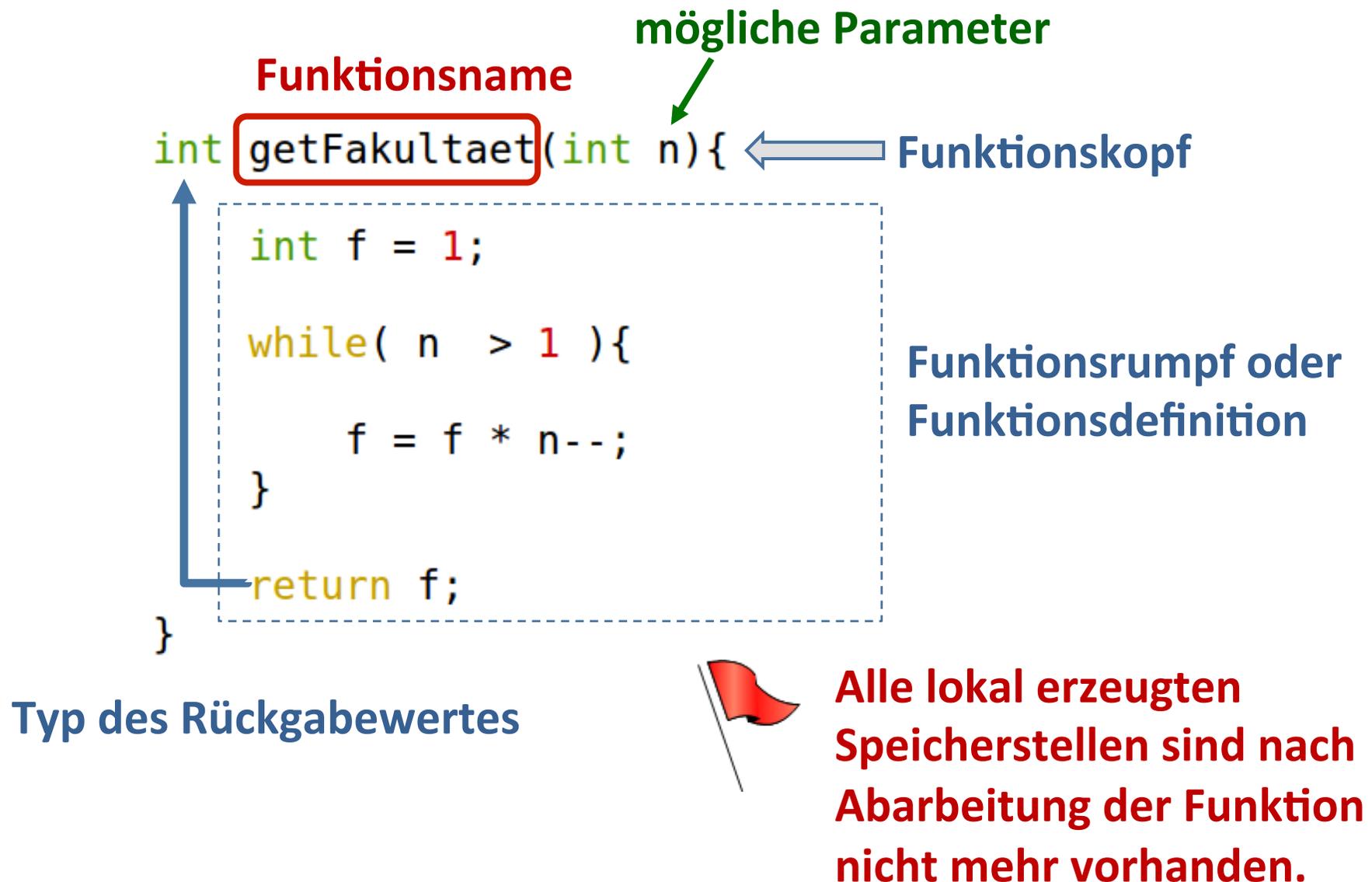
lokales int f

lokales int n

**Ende innerer Block (Ende f, n)**



# Aufbau einer Funktion



### Das Prinzip einer Funktion

**Eine Funktion kann maximal einen Rückgabewert besitzen**

**Der Prototyp informiert den Compiler über Name, Parametertyp und Rückgabetyt für den späteren Gebrauch**

**Funktionen können jeden gültigen Variablentyp als Rückgabewert besitzen. Gibt die Funktion keinen Wert zurück so lautet der Rückgabewert void.**

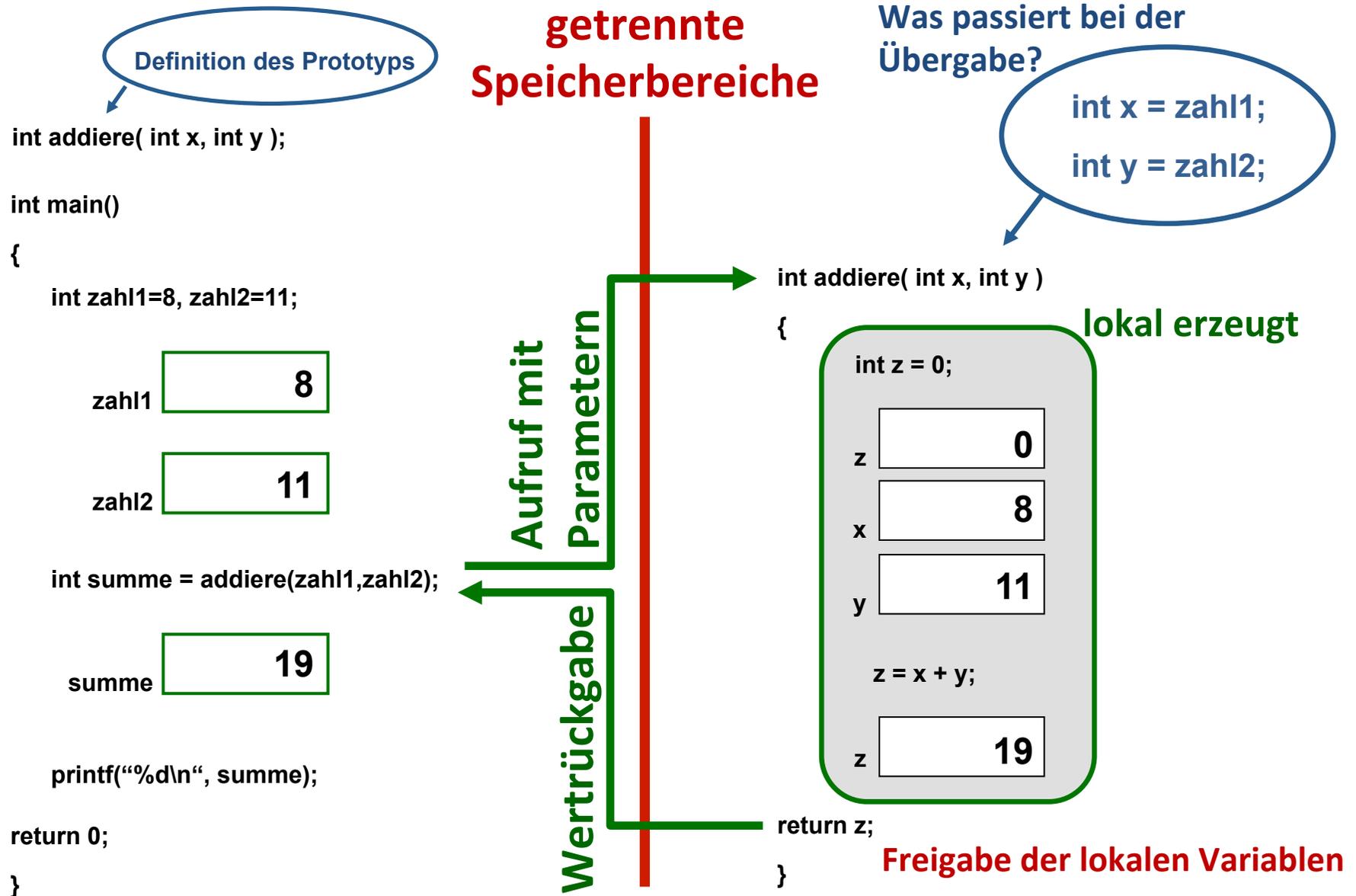
**Funktionen können sich gegenseitig aufrufen.**

**Funktionen können beliebig viele Parameter übernehmen.**

**Jeder Parameter wird bei dem Funktionsaufruf neu erzeugt.**

**Der Gültigkeitsbereich einer Funktion entspricht dem eines inneren Blocks**

## Beispiel: Berechnen einer Summe



## Beispiel: Funktion ohne Rückgabewert

```
#include <stdio.h>
#include <stdlib.h>
```

```
void ausgabe();
```

```
int main()
{
    ausgabe();

    getchar();
    return 0;
}
```

```
void ausgabe(){
    printf("Ich besitze keinen Rueckgabewert!\n");
}
```



*Weiter geht's mit Übungen ...*