

# Programmieren in C

## Grundlagen

thoto

/dev/tal e.V.

16. März 2013

# Agenda für Heute

- 1 Variablen
  - Was ist das?
  - Typen
  - weiteres zu Zahlen
- 2 Funktionen und Methoden
  - Grundfunktionen
  - Methoden

Variablen

# Kurzer Nachtrag:

- Kommentare

# Kurzer Nachtrag:

- Kommentare

```
ganzzeitig int foo;// ganzzeiliger Kommentar
```

## Kurzer Nachtrag:

- Kommentare

```
ganzzeilig int foo;// ganzzeiliger Kommentar
```

```
normal int /*ganzzahl!*/ foo;
```



# Wir erinnern uns ...

- Bombenbauanleitung

## Wir erinnern uns ...

- Bombenbauanleitung
- `char helloworld[]=helloworld;`



## Wir erinnern uns ...

- Bombenbauanleitung
- `char helloworld[]=helloworld;`
- `int sprache;`

## Wir erinnern uns ...

- Bombenbauanleitung
- `char helloworld[]=helloworld;`
- `int sprache;`
- `sprache=1;`



Was ist das?

# Deklaration

- `int foo,bar;`

# Deklaration

- `int foo,bar;`
- Muster: `typ name1,name2,name3;`

# Deklaration

- `int foo,bar;`
- Muster: `typ name1,name2,name3;`
- Konstante: `const int foo,bar;`

# Deklaration

- `int foo,bar;`
- Muster: `typ name1,name2,name3;`
- Statische Variable: `static int foo,bar;`

# Deklaration

- `int foo,bar;`
- Muster: `typ name1,name2,name3;`
  
- Zeiger: `int *foo, *bar;`



Was ist das?

# Konstanten



# Konstanten: Beispiel

## Beispiel

```
const int foo=3;  
printf("%d\n", foo); // gibt foo -- also 3 -- aus
```

# Konstanten: Gegen-Beispiel

**Falsch!**

```
const int foo;  
foo=3;
```

# Konstanten: Gegen-Beispiel

**Falsch!**

```
const int foo;  
foo=3;
```

```
datei.c:2:1: error: assignment of read-only variable 'foo'
```

Typen



# Ganzzahlen

- Integer
- Negativ und Positiv bei signed
- Keine Kommazeichen darstellbar
- 32bit: [-16777216|16777215]
- 64bit: mehr ...



# Fließkommazahlen

- float
- Kommadarstellung
- sehr große und sehr kleine Zahlen
- Auch Darstellung von Nichtzahlen
- Details unnötig



# Buchstaben

- char
- *einzelner* Buchstabe
- ASCII gewöhnlich
- 7/8-Bit Zahl

# Zeichenketten

- mehrere chars
- mehr nach Pointern!
- `char foo []=asdf;`



weiteres zu Zahlen



# Zahlnotationen

- Hexadezimal: `0xF3`
- Oktal: `0127`
- Dezimal: `123`
- Binaer: `0b0010`



# Überläufe

- <http://xkcd.com/571>
- bei sehr großen/kleinen Zahlen
- Beispiel ...

# Funktionen und Methoden

## Grundfunktionen

# printf

- Ausgabe.
- Syntax: `printf(format, quelle1, quelle2, ...);`
- Formate: z.B. `%s %d %x ...`

# scanf

- Eingabe.
- Syntax: `scanf(format,ziel1,ziel2,...);`
- Formate: z.B. `%s %d %x ...`

# Rechnen

```
foo=1-bar
```

Methoden



# Methoden

```
void methode(typ parameter1,typ parameter2,...){  
    ...  
}
```

# Beispiel

```
[language=C]
void addieren(int zahl1,int zahl2){
    printf("Summe von %d und %d ist %d\n",zahl1,zahl2,\
        zahl1+zahl2);
}
...
addieren(2,3);
```

## Beispiel 2

```
void addieren(int zahl1,int zahl2){  
    int zahl3;  
    zahl3=zahl1+zahl2;  
    printf("Summe von %d und %d ist %d\n",zahl1,zahl2,zahl3);  
}  
  
...  
addieren(2,3);
```

# Statische Variablen

**Vorsicht!**

Zugriff auf zahl3 aus main() nicht möglich!