

genereller Aufbau

Parameter	Beschreibung
r	
* NUR scanf	optional , Zuweisung zum korrespondierenden Zeiger unterdrücken
Flags NUR printf	optional , bestimmt über numerische Vorzeichen, Dezimalpunkte, link/rechtsbündigkeit und anderes
Breite	optional , minimal auszugebende Zeichenzahl. Der Rest wird mit Leerzeichen oder Nullen aufgefüllt.
F/N NUR scanf	optional , Größe des Adress-Parameters
.Präzision	optional , Angabe, wie präzise die Ausgabe sein soll
F/N/hh/h/l/l/L	optional , Angabe der Größe des Parameters
Typ	obligatorisch , Angabe des Typs der auszugebenden Variablen

printf:

% [Flags] [Breite] [.Präzision] [F/N/...] Typ

scanf:

% [*] [Breite] [F/N][hh|h||l|L] Typ

KEINE Leerzeichen! Sie dienen hier nur der Übersichtlichkeit!

Unterschiede der Präzision je nach Datentyp

Zeichenketten: maximal auszugebende Zeichen

ganze Zahlen: Minimalzahl von Zeichen

Fließkommazahlen: Maximalzahl der Nachkommastellen

Besonderheiten von scanf

Es dürfen nur Platzhalter und Leerzeichen verwendet werden

Anstatt %s kann auch %[Suchzeichen] genutzt werden.

Beispiel:

%[abcd] erwartet nur Eingaben, die aus den Zeichen a, b, c, d bestehen. Bei allen anderen wird die Eingabe beendet.

Wird als erstes Suchzeichen das Carret (^) verwendet, wird die Eingabe beendet, wenn eines der Suchzeichen eingegeben wird.

Beispiel:

%^[abcd] erwartet nur Eingaben, die **nicht** aus den Zeichen a, b, c, d bestehen. Die Eingabe wird auch nicht bei weißen Leerzeichen beendet.

Anstelle von Zeichenaufzählungen können auch Bereiche angegeben werden.

Beispiel:

% [0123456789] ist äquivalent mit %[0-9]

Bei manchen Compilern kann bzw. muss bei der Formatangabe das s angehängt werden.

Flags

Flag	Effekt
-	linksbündige Ausgabe
+	numerische Ausgabe immer mit Vorzeichen
<Leerzeichen>	Positiven Zahlen wird ein Leerzeichen voran gestellt
#	Alternative Darstellung

Alternative Darstellungen:

o: Es wird eine 0 vorangestellt.

x/X: es wird ein 0x bzw. 0X vorangestellt.

e/E/f: Es wird ein Dezimalpunkt ausgegeben, auch wenn es keine Nachkommastellen gibt.

g/G: wie bei e/E, außerdem werden folgende Nullen nicht unterdrückt.

ist nicht erlaubt bei Typ d/i/u/c/s/p

+ ist nicht erlaubt bei Typ c/s

Breite

Breite	Ausgabe
n	(n = Dezimalzahl) - Es werden mind. n Zeichen ausgegeben und notfalls führende Leerzeichen genutzt.
0n	(Dezimalzahl mit vorangestellter 0) - Es werden mind. n Zeichen ausgegeben und mit Nullen mit führenden Nullen aufgefüllt.
*n	Breite variabel bestimmbar

Beispiel *n

```
printf("%*d\n", Breite, Wert);
//Breite muss int sein!
```

Präzision

Präzision	Ausgabe
.0	Standardausgabe - keine Ausgabe von Dezimalpunkt und Nachkommastellen für Fließkommazahlen
.n	n = Dezimalzahl
.*	variable Präzision

Beispiel .*

```
printf("%.f\n", Breite, Präzision, Wert); //Breite und Präzision müssen int sein!
```

Unterschiede je nach Datentyp:

ganze Zahlen: Mindestanzahl von auszugebenden Ziffern

Fließkommazahlen (e/E/f/F): Ausgabe von n Nachkommastellen

Fließkommazahlen (g/G/a/A): Ausgabe von n Ziffern

Zeichenketten: maximale Anzahl von auszugebenden Zeichen



Größenangaben	
Größenangabe	Datentyp
hh ³	char
h ¹	short
l	long
ll ³	long long
L	long double
F ⁴	Far (Zeiger)
N ⁴	Near (Zeiger)

1: erst seit C89
 2: erst seit C95
 3: erst seit C99
 4: Nur in 16-Bit Compilern

Besonderheiten:
 Bei cl ist der Datentyp wint_t²
 Bei sl ist der Datentyp wchar_t*³

Typ	
Typ	Ausgabe
d / i	dezimaler Integer
u	vorzeichenloser dezimaler Integer
o	vorzeichenloser oktaler Integer
x / X	vorzeichenloser hexadezimaler Integer
f / F	dezimale Fließkommazahl
e / E	dezimale Fließkommazahl in Exponentendarstellung
g / G	wie e/E, aber in Kurzform
c	Zeichen
s	Zeichenkette bis zu einem Nullzeichen oder dem Erreichen der Präzision
p	Zeigeradresse

