

Die folgende Tabelle gibt alle möglichen Typen an.

Typ	optionale Größenangabe	Datentyp des Parameters	Ausgabe	erlaubte Flags
d i	(keine) hh h l ll	int char short long long long	dezimaler Integer	- + Leerzeichen
u	(keine) hh h l ll	unsigned int unsigned char unsigned short unsigned long unsigned long long	dezimaler Integer (nur positive Zahlen)	- + Leerzeichen
o	(keine) hh h l ll	unsigned int unsigned char unsigned short unsigned long unsigned long long	oktaler Integer (nur positive Zahlen)	- + # Leerzeichen
x X	(keine) hh h l ll	unsigned int unsigned char unsigned short unsigned long unsigned long long	hexadezimaler Integer (nur positive Zahlen) bei Typ x: mit Buchstaben a...f bei Typ X: mit Buchstaben A...F	- + # Leerzeichen
f F	(keine) L	float, double long double	dezimale Fließkommazahl; vorzeichenbehafteter Wert der Form [-]d.dddddd, die Anzahl der Nachkommastellen kann durch die Angabe .Präzision festgelegt werden. Ausgabebeispiele: -.900000, 3.141500	- + # Leerzeichen
e E	(keine) L	float, double long double	dezimale Fließkommazahl; vorzeichenbehafteter Wert der Form [-]d.dddddd e [+ -]dd (Exponentendarstellung), es steht grundsätzlich eine Ziffer vor dem Dezimalpunkt, die Anzahl der Nachkommastellen kann durch die Angabe .Präzision festgelegt werden, der Exponent hat immer zwei Ziffern (notfalls mit führender Null). Ausgabebeispiele: -1.900000e+00, 2.550000e-03	- + # Leerzeichen

Typ	optionale Größenangabe	Datentyp des Parameters	Ausgabe	erlaubte Flags
<b>g G</b>	(keine) L	float, double long double	dezimale Fließkommazahl; vorzeichenbehafteter Wert im e- oder f- bzw. im E- oder F-Format. Nullen am Ende, ein Dezimalpunkt und ein Vorzeichen werden nur ausgegeben, wenn es notwendig ist. Das e-Format wird nur verwendet, wenn das Resultat im f-Format mehr als .Präzision Stellen ergibt oder mehr als vier führende Nullen erfordert.	- + # Leerzeichen
<b>a A</b>	(keine) L	float, double long double	hexadezimale Fließkommazahl; sonst wie Typ g bzw. G, anstelle des Buchstabens e bzw. E für den Exponenten wird das p bzw. P verwendet, da das e / E zu den hexadezimalen Ziffern gehört.	- + # Leerzeichen
<b>c</b> <b>s</b>	(keine) (keine)	int char *	Zeichen Zeichenkette (Array von char) bzw. Zeichenkette (Array von Multibytes) bis zu einem Nullzeichen oder dem Erreichen der durch .Präzision vorgegebenen Zeichenzahl.	- -
<b>p</b>	(keine) F N	void *	Compilerabhängig; Far (F) und Near (N) -Zeiger sind nur in 16-Bit- Compilern enthalten.	Compiler- abhängig
<b>%</b>	(keine)	-	Ausgabe des Zeichens %	(keine)

Flags	Ausgabe
- +	linksbündige Ausgabe (im Normalfall werden Zahlen rechtsbündig ausgegeben) numerische Ausgabe immer mit Vorzeichen (im Normalfall wird nur bei negativen Zahlen ein Vorzeichen ausgegeben)
Leerzeichen	Positiven Zahlen wird ein Leerzeichen vorangestellt; wird dieses zusammen mit dem + verwendet, wird das Leerzeichen ignoriert!
#	Alternative Darstellung Typ o: es wird eine 0 vorangestellt Typ x/X: es wird "0x" bzw. "0X" vorangestellt Typ e/E/f: es wird ein Dezimalpunkt ausgegeben, auch wenn es keine Nachkommastellen gibt Typ g/G: wie bei e und E, zusätzlich werden folgende Nullen nicht unterdrückt
Breite	Ausgabe
n	(n: Dezimalzahl) Es werden mind. n Zeichen ausgegeben und notfalls Leerzeichen vorangestellt.
0n	(0n: Dezimalzahl mit vorangestellter 0) Es werden mind. n Zeichen ausgegeben und notfalls Nullen vorangestellt. Anstelle einer Zahl kann auch ein Stern (*) eingesetzt werden. Es muss dann aber in der Argumentenliste die Breite als int angegeben werden. Beispiel: <pre>int Breite = 5, Wert = 1; printf("%5d\ n", Wert); printf("%*d\ n", Breite, Wert);</pre>
.Präzision	Ausgabe
.0	Standardvorgabe für ganze Zahlen (Typ d, i, u, o, x und X); keine Ausgabe von Dezimalpunkt und Nachkommastellen für Fließkommazahlen (Typ e, E, f, F, g, G, a und A)
.n	(n: Dezimalzahl) Mindestanzahl von auszugebenen Ziffern für ganze Zahlen (Typ d, i, u, o, x und X); Ausgabe von n signifikanten Nachkommastellen für Fließkommazahlen (Typ e, E, f und F); Ausgabe von n signifikanten Ziffern für Fließkommazahlen (Typ g, G, a und A); maximale Anzahl von auszugebenen Zeichen für Zeichenketten (Typ s). Auch für die Präzision kann anstelle einer Zahl ein Stern (*) eingesetzt werden. Es muss dann in der Argumentenliste die Präzision als int angegeben werden. Wenn für beide, Breite und Präzision, ein Stern eingesetzt wird, muss in der Argumentenliste erst die Angabe der Breite und dann die der Präzision erfolgen. Beispiel: <pre>int Breite = 5, Praezision = 3; float Wert = 1.234567; printf("%5.3f\ n", Wert); printf("%*.*f\ n", Breite, Praezision, Wert);</pre>