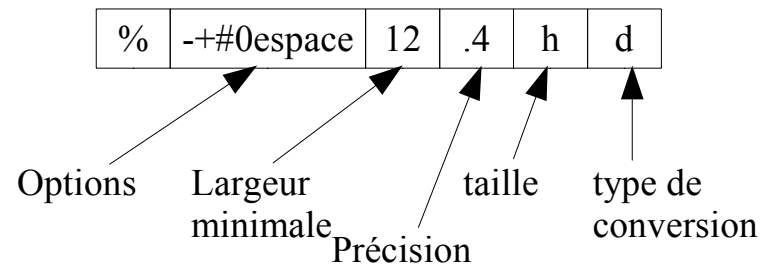


printf: formats de sortie

Ce qui suit est valable pour les fonctions de la famille printf (printf, fprintf, ...). printf affiche son premier argument qui est une chaîne de caractère en remplaçant chaque spécifications de format commençant par un % par l'argument correspondant (le premier argument correspond au premier %, le deuxième au deuxième %, ...). Une erreur dans la spécification de format d'un argument peut perturber le décodage des arguments suivants et provoquer l'affichage de choses sans sens.



Option : -	aligné à gauche	"12 " pour %-4d appliqué à 12
Option: +	une valeur numérique positive est précédée par +	" +12" pour %+4d appliqué à 12
Option: 0	bourré avec des zéros à gauche au lieu d'espaces	"0012" pour %04d appliqué à 12
Option: espace	si le nombre est positif et sans signe +, un espace est ajouté devant à la place du signe + absent.	" 12" pour % 4d appliqué à 12
Option: #	l'effet dépend du type de conversion	
Largeur minimale	largeur minimale d'affichage. des espaces sont ajoutés à gauche pour atteindre cette taille. Si la valeur à afficher fait plus que cette taille, elle est affichée complètement (c'est une largeur minimale d'affichage).	" 12" pour %3d appliqué à 12
précision	nombre de chiffres après la virgule. Le nombre est arrondi.	" 12.23" pour %6.2f appliqué à 12.231
taille	taille de l'argument (short, long)	voir tableau de types de conversions

lettres de conversions de types :

<i>Conversion</i>	<i>Options</i>	<i>Modificateur de taille</i>	<i>Type d'arguments</i>	<i>Précision par défaut</i>	<i>Exemple de sortie</i>	<i>Remarque</i>
d, i	- + 0 espace	sans	int	1	"-12", "00012", "+12"	"i" est conservé pour raison historiques. Il ne faut pas l'utiliser.
		h	short	1		
		l	long	1		
f	- + # 0 espace	sans	double	6	"12.36", "-0.0032"	si l'argument est un float, il sera converti en double.
		L	long double	6		
e,E	- + # 0 espace	sans	double	6	"12.34e-5" "12.34E-5"	e: exposant e, E: exposant E; si l'argument est un float, il sera converti en double.
		L	long double	6		
g,G	- + # 0 espace	sans	double	6	"12.34e-5" "12.34E-5" "12.3"	g: exposant e, G: exposant E
		L	long double	6		idem e, E mais si la valeur peut être affichée sans exposant, elle l'est.
c		sans			"a"	
		l	wint_t			caractères multi octets
u,o,x,X	- # 0	sans	unsigned int	1	"1e" "36" "036" "OX1E" (30 en hexa, octal, octal et hexa)	u: non signé, o: octal, x: hexadecimal, X : hexadécimal en majuscule
		h	unsigned short	1		#: préfixe 0x, 0X et O : "0x1e4"
		l	unsigned long	1		
s		sans	char *		"abcdef"	
		l	wchar_t *			caractères multi octets
%					%	%% représente le caractère %

