

## Programmation C `printf` `fprintf` `sprintf`

Pour utiliser la fonction **printf** il faut inclure l'en-tête

`#include <stdio.h>` au début du code source du programme.

### 1 *printf*

Pour afficher un message sur la sortie standard (par défaut, l'écran).

```
printf ( "Bonjour tout le monde" );
```

Pour afficher le contenu d'une variable dans un message:

```
int a=10;
printf ( "valeur de a = %d" , a);
```

Pour afficher autant de contenus de variables que l'on veut, il suffit d'indiquer le nom des variables dans l'ordre, exemple :

```
long nombre1 = 400, nombre2 = 500;
printf("Le nombre 1 est egal a %ld et le nombre 2 a %ld",
      nombre1, nombre2);
```

Type	Lettre	commentaire
char	%c	<b>caractère 1 octet</b> (8 bits)
short	%hd	<b>entier 2 octets</b> (-32768,+32767)
int	%d	<b>entier 4 octets</b> (-2 <sup>31</sup> à +2 <sup>31</sup> -1)
long	%ld	<b>entier 4 octets</b>
float	%f	<b>réel sur 4 octets</b> (10 <sup>38</sup> )
double	%lf	<b>réel sur 8 octets</b> (10 <sup>308</sup> )
string (char*)	%s	<b>string</b> tableau de caractères
pointeur (void*)	%p	<b>adresse</b> d'une variable
int	%x	entier en hexa, lettres minuscules
int	%X	entier en hexa, lettres majuscules
int	%o	entier en octal

Pour afficher un réel avec plus de précision exemple 5 chiffres après la virgule

```
float x=0.12345;
printf("%.5f",x);
```

### 2 *fprintf*

fprintf permet d'écrire dans les fichiers de la même manière que printf écrit sur la sortie standard.

### 3 *sprintf*

sprintf renvoie le résultat dans une variable.

Le premier paramètre de cette fonction est la chaîne de caractères qui recevra le résultat de la conversion. Le deuxième paramètre est la chaîne de format qui servira de patron pour la conversion et il sera évidemment suivi par les variables à convertir.

```
char variable[256];
char prenom[] = "Anna";
char nom[] = "Dupond";
int age = 30;

sprintf(variable, "%s %s a %d ans", prenom, nom, age);

printf ("Resultat : %s\n", variable);
```