



Informatique S1 Programmation C

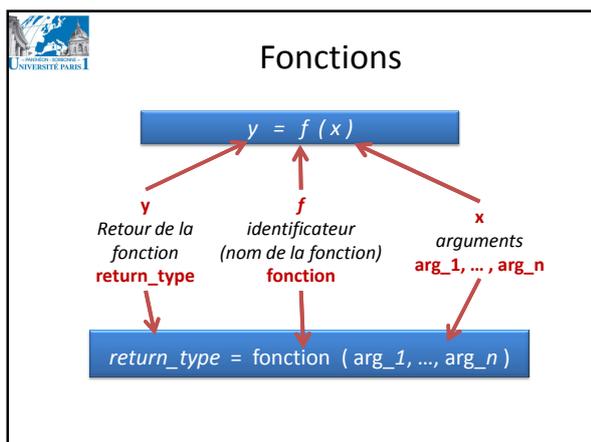
- Objectifs :
 - Usage des fonctions
 - Définition des constantes



Fonctions

- Définition
 - Une fonction est un module constitué d'une *suite d'instructions*
 - Une fonction est un morceau de code réutilisable
- Exemples


```
main ()
printf ("Hello world !")
c = getchar()
```




Usage des fonctions

- On *appelle* une fonction par son nom, en indiquant les *arguments* appropriés
- Exemples :
 - ```
racine = sqrt (x);
```

Fonction : « sqrt »  
Argument : « x »  
Retour sur la variable « racine »
  - ```
printf ("x = %f ", x);
```

Fonction : « printf »
Arguments : « x = %f » et « x »
Retour ignoré
 - ```
c = getchar ();
```

Fonction : « getchar »  
Argument : pas d'argument  
Retour sur la variable « c »

### Exemple

```

5 /* cm7 : swith (devinette) */
6 int main () {
7 int i,x;
8 srand (time(NULL)); //initialise le generateur
9 x = rand()%11;
10
11 printf ("Entrez i [0-10] : ");
12 scanf ("%d", &i);
13 ...
14 }

```

### Fonction main

```

int main (int argc, char argv[])
{
...
}

```

### Bibliothèques

- Les fonctions peuvent être organisées dans les bibliothèques
- Bibliothèques → ensemble de fonctions
- Bibliothèques standard
 

|                 |                 |                     |
|-----------------|-----------------|---------------------|
| <i>stdio.h</i>  | <i>stdlib.h</i> | <i>math.h</i>       |
| <i>time.h</i>   | <i>ctype.h</i>  | <i>limits.h</i>     |
| <i>string.h</i> | <i>errno.h</i>  | <i>float.h</i>      |
| <i>locale.h</i> | <i>stdarg.h</i> | <i>stddef.h</i> ... |

### Bibliothèques

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 /* cm7 : swith (devinette) */
6 int main () {
7 int i,x;
8 srand (time(NULL)); //initialise le generateur
9 x = rand()%11;
10
11 printf ("Entrez i
12 scanf ("%d", &i);
13 }

```

Les fonctions de *stdlib.h* sont disponibles :

- > *srand (...)*
- > *rand ()*
- > *abs ()*
- > ...

Pour compiler  
gcc -lm -o prog prog.c

### Quelques fonctions utiles...

- **stdio.h**
  - int printf (char\* format, ...)
  - int scanf (char\* format, ...)
  - int getchar (void)
  - int putchar (int c)
- **stdlib.h**
  - int abs (int n)
  - long labs (long n)
  - int rand (**void**)
  - void srand (unsigned int seed)
- **ctype.h**
  - int isalpha (int c)
  - int isupper(int c)
  - int islower(int c)
  - int iscntrl (int c)
  - int isdigit (int c)
  - int isalnum (int c)
  - int tolower(int c)
  - int toupper(int c)
- **math.h**
  - double pow(double x, double y)
  - double sqrt(double x)
  - double log10(double x)
  - double fabs(double x)
  - double sin(double x)

**void**  
Type de donnés  
indique pas d'argument ou  
pas de retour

### Définition d'une constante

- Les bibliothèques peuvent contenir la définition des constantes
- **constante** → valeur ne change pas
  - M\_PI                    x = pow (M\_PI, 2);
  - NULL                    srand (time(NULL));
  - FLT\_MAX
  - DBL\_MAX
  - SHRT\_MAX
  - INT\_MIN
  - ...

### Définition d'une constante

- Définition
 

**#define** CTE valeur

Directive pré-compilation

gcc remplace tous les « CTE » par « valeur »

Habituellement, on utilise des MAJUSCULES pour les noms des constantes
- Exemples
 

```
#define MAX 10
#define FAUX 0
#define VRAI 1
```

### Exemple

```
1 #include <stdio.h>
2 #define MIN -15
3 #define MAX 45
4
5
6 /* Conversion C-F */
7 main () {
8 float fahr;
9 int cels;
10
11
12 printf ("Celsius\tFahrenheit \n");
13
14 for (cels=MIN; cels <= MAX; cels+=5) {
15 fahr = ((double)cels * 9)/5 + 32;
16 printf (" %d \t %2.2f\n", cels, fahr);
17 }
18
19 }
```

**Définition**  
#define MIN -15  
#define MAX 45

**Usage**  
cels = MIN;  
cels <= MAX;