

# La dernière fois ...

- ▶ tri d'un tableau d'entiers

# Les fonctions

# Retour sur les fonctions

Déclaration d'une fonction :

```
Fonction  nomDeFonction  (arg1,arg2 : type1) :      type2
```

# Retour sur les fonctions

Déclaration d'une fonction :

Fonction    nomDeFonction    (arg1,arg2 : type1) :    type2



on dit que  
c'est une  
fonction

on lui donne  
un **nom**

la liste des  
**arguments** avec  
leur nom et leur  
type

le type du  
**résultat** renvoyé  
par la fonction

# Afficher et Renvoyer

A la fin d'une fonction, il faut toujours **renvoyer** le résultat :

```
Fonction truc(x:Entier) : Entier
```

```
Debut
```

```
Renvoyer(4*x+2)
```

```
Fin
```

et le résultat renvoyé doit être du bon type !

# Afficher et Renvoyer

A la fin d'une fonction, il faut toujours **renvoyer** le résultat :

```
Fonction truc(x:Entier) : Entier
```

```
Debut
```

```
Renvoyer(4*x+2)
```

```
Fin
```

et le résultat renvoyé doit être du bon type!

**Attention ! Renvoyer  $\neq$  Afficher**

# Petits exemples

Fonction affine( $x, a, b$ : Réel) : Réel

Debut

Renvoyer( $a*x+b$ )

Fin

## Petits exemples

Fonction affine( $x, a, b$ : Réel) : Réel

Debut

Renvoyer( $a*x+b$ )

Fin

Fonction polynome( $x, a, b, c$ : Réel) : Réel

Debut

Si ( $a=0$ )

Alors Renvoyer( $a*x+b$ )

Sinon Renvoyer( $a*x*x+b*x+c$ )

Fin

## Appel d'une fonction par une autre

```
Fonction mult(x,y:Réels) : Réel
```

```
Debut
```

```
Renvoyer(x*y)
```

```
Fin
```

## Appel d'une fonction par une autre

```
Fonction mult(x,y:Réels) : Réel
```

```
Debut
```

```
Renvoyer(x*y)
```

```
Fin
```

```
Fonction puiss(x: Réel;n: Entier) : Réel
```

```
Var i: Entier; temp : Réel
```

```
Debut
```

```
temp ← 1
```

```
  Pour i=1 à n faire
```

```
    temp ← mult(temp,x)
```

```
  FinPour
```

```
Renvoyer(temp)
```

```
Fin
```

*mult(temp,x)* se comporte comme un Réel

# Fonction qui ne renvoie rien

Exemple :

```
Fonction echange(tab : Tableau d'Entiers; i,j : Entiers)
```

```
Debut
```

```
...
```

```
Fin
```

## Fonction qui ne renvoie rien

Exemple :

```
Fonction echange(tab : Tableau d'Entiers; i,j : Entiers)
```

```
Debut
```

```
...
```

```
Fin
```

Le tableau est **modifié** :

```
tab ← [4,2,1,6,5];
```

```
echange(tab,0,2);
```

```
Afficher(tab);
```

mais la fonction ne renvoie pas de résultat. On ne peut pas écrire :

```
tab ← [4,2,1,6,5];
```

```
newtab ← echange(tab,0,2);
```

## Fonction qui ne renvoie rien

Exemple :

```
Fonction echange(tab : Tableau d'Entiers; i,j : Entiers)
```

```
Debut
```

```
...
```

```
Fin
```

Le tableau est **modifié** :

```
tab ← [4,2,1,6,5];
```

```
echange(tab,0,2);
```

```
Afficher(tab);
```

mais la fonction ne renvoie pas de résultat. On ne peut pas écrire :

```
tab ← [4,2,1,6,5];
```

```
newtab ← echange(tab,0,2);
```

**Fonction sans sortie = Procédure**

# Récurtivité

# Principe

- ▶ récursivité = technique de définition de fonction
- ▶ une fonction récursive fait appel à elle-même dans sa définition (=appel récursif)

# Principe

- ▶ récursivité = technique de définition de fonction
- ▶ une fonction récursive fait appel à elle-même dans sa définition (=appel récursif)

Exemple :

Fonction `test(n:Entier) : Booléen`

Debut

    Si ( $n < 0$ )

        Alors Renvoyer(Faux)

    Sinon Renvoyer `test(n-1)`

Fin

## Et ça ne pose pas de problème?

Que fait cette fonction récursive?

```
Fonction essai(n:Entier) : Entier
```

```
Debut
```

```
    Renvoyer (essai(n-1))
```

```
Fin
```

## Et ça ne pose pas de problème ?

Que fait cette fonction récursive ?

```
Fonction essai(n:Entier) : Entier
```

```
Debut
```

```
    Renvoyer (essai(n-1))
```

```
Fin
```

Attention à l'initialisation des appels récursifs !!

## Et ça ne pose pas de problème? (bis)

Que fait cette fonction récursive?

```
Fonction essai(n:Entier) : Entier
```

```
Debut
```

```
  Si (n<0)
```

```
    Alors Renvoyer(0)
```

```
  Sinon Renvoyer (essai(n)+3)
```

```
Fin
```

## Et ça ne pose pas de problème? (bis)

Que fait cette fonction récursive?

```
Fonction essai(n:Entier) : Entier
```

```
Debut
```

```
  Si (n<0)
```

```
    Alors Renvoyer(0)
```

```
  Sinon Renvoyer (essai(n)+3)
```

```
Fin
```

Attention à l'argument de l'appel récursif !!

## Exemple : nombres de Fibonacci

Il peut même y avoir **plusieurs** appels récursifs dans la définition de la fonction :

```
Fonction fibo(n: Entier): Entier
```

```
Debut
```

```
  Si (n<2)
```

```
    Alors Renvoyer (1)
```

```
  Sinon Renvoyer( fibo(n-1)+fibo(n-2) )
```

```
Fin
```

## Exemple : nombres de Fibonacci

Il peut même y avoir **plusieurs** appels récursifs dans la définition de la fonction :

```
Fonction fibo(n: Entier): Entier
```

```
Debut
```

```
  Si (n<2)
```

```
    Alors Renvoyer (1)
```

```
  Sinon Renvoyer( fibo(n-1)+fibo(n-2) )
```

```
Fin
```

⇒ Exercices!