

LES PROCÉDURES

Principes généraux et compléments

Déclaration d'une procédure

procedure <i>Nom de la procédure</i> <i>Liste de paramètres</i> ;	Entête
var déclarations des variables locales	optionnel
begin <i>Instructions</i> end;	Corps

Liste de paramètres (optionnel)

```
( nom : type , nom : type, ....., nom : type )
```

alternative plus compacte:

```
( liste de noms : type ; liste de noms : type ; .....; liste de noms : type )
```

liste de noms:

```
nom , nom, ....., nom
```

APPEL DE PROCÉDURE

Nom de la procédure **paramètres effectifs** ;

Paramètres effectifs:

(**expression , expression, , expression**)

- Nombre de paramètres effectifs = nombre de paramètres formels
- type paramètre effectif compatible avec type paramètre formel

EXEMPLE

```

procedure FormerMail (p, n , o : string);
begin
    mail := p+'.'+n+'@'+o+'.fr';
end;

```

Appel	Résultat
FormerMail ('Jean' + '-' + 'Luc', 'Sermo', 'free');	mail := 'Jean-Luc.Sermo@free.fr;
FormerMail ('Jean','Luc','Sermo','free');	erreur de compilation !
FormerMail ('Jean-Luc','free');	erreur de compilation !
FormerMail ('Jean-Luc','Sermo', 9);	erreur de compilation !

Passage des paramètres et exécution

Variable	Prenom1	Prenom2	Nom
Valeur	'Jean'	'Sébastien'	'Bach'

Entête	procedure FormerMail (p, n, o : string);		
Appel	FormerMail (Prenom1 +'-' Prenom2, Nom, 'free');		
1) Evaluation des paramètres effectifs		2) Allocation mémoire pour les paramètres formels et affectation des valeurs	
Paramètre effectif	Valeur	Paramètre formel	Valeur
Prenom1 +'-' Prenom2	'Jean-Sébastien'	p	'Jean-Sébastien'
Nom	'Bach'	n	'Bach'
'free'	'free'	o	'free'
3) Exécution du corps avec ces valeurs: mail := p+'.'+n+'@'+o+'.fr';		4) Libération de la place mémoire occupée par p, n et o.	