

**KEEP
CALM
AND
<CODE/>**

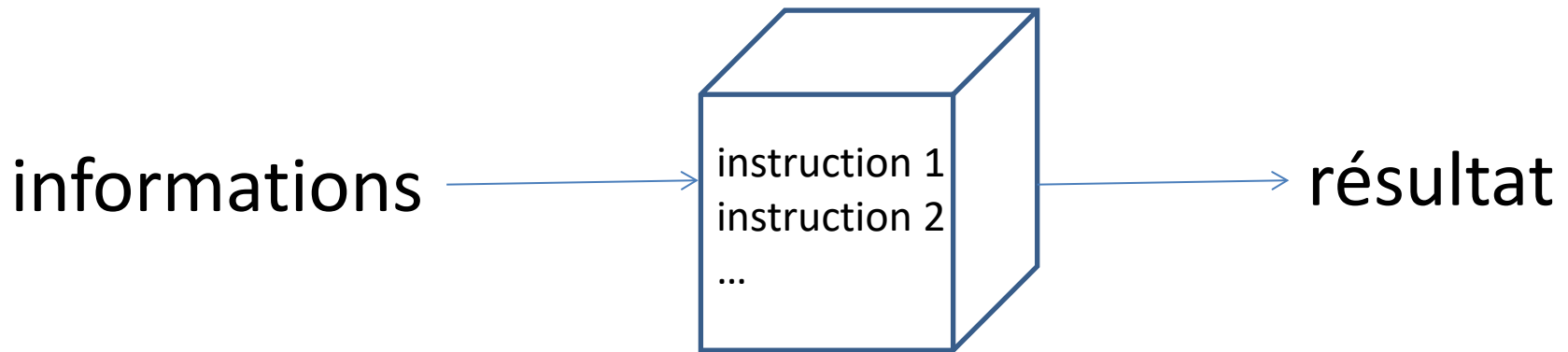
Algorithmique : les fonctions

Sylvie Trouilhet

www.irit.fr/~Sylvie.Trouilhet

Qu'est-ce qu'une fonction ?

- Une fonction est une boîte qui effectue un traitement et retourne (éventuellement) un résultat. Pour effectuer le traitement, une fonction peut avoir besoin d'informations.



Qu'est-ce qu'une fonction ?

- Bonne pratique : une fonction ne fait qu'un traitement (responsabilité unique)
 - Par exemple : fonction pour calculer un prix
- Si l'on doit faire plusieurs paramètres, on écrit plusieurs fonctions
 - Par exemple : calculer un prix, faire une remise, encaisser le paiement...

Définir une fonction

- Une fonction est constituée d'une signature et d'un corps
 - **Signature** : nom de la fonction et paramètres formels.
 - **Paramètres** formels : informations à passer pour que la fonction puisse faire son traitement. Le nombre de paramètres peut être 0, 1, 2, ...
- La signature est la première ligne de la définition
- def** nomFonction(par1, par2, ...):

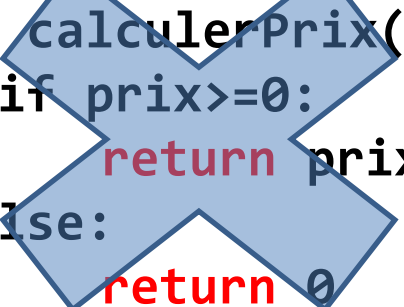
Définir une fonction

- **Corps** = ce sont les instructions (organisées à l'aide de structures de contrôle) qui seront exécutées pour faire le traitement
- Si la fonction fournit un résultat, il y a une instruction **return**.

```
def nomFonction(par1, par2, ...):  
    instruction 1  
    instruction 2  
    ...  
    return résultat
```

Définir une fonction

- Bonne pratique : séparer le traitement du résultat
→ un seul return à la fin de la fonction



```
def calculerPrix(prix):  
    if prix >= 0:  
        return prix * 1,2  
    else:  
        return 0
```

```
def calculerPrix(prix):  
    rep = 0  
    if prix >= 0:  
        rep = prix * 1,2  
    return rep
```

- À retenir : une information a TOUJOURS des parenthèses, même sans paramètre

```
def nomFonction()
```

Utiliser une fonction

- Pour utiliser une fonction, on l'appelle en lui indiquant les valeurs avec lesquelles elle doit faire le traitement.
- Ces valeurs sont appelées les **paramètres réels**
- **Appel** de fonction :
 nomFonction(parRéel1, parRéel2, ...)
- On peut appeler une fonction plusieurs fois
 rep1=calculerPrix(**12**)
 rep2=calculerPrix(**12+6**)
 rep3=calculerPrix(**rep1**)

Exemple

#définition

```
def calculerPrixTTC(prix):  
    TVA=20/100  
    ttc=0  
    if prix>0: ttc=prix+prix*TVA  
    return ttc
```

#appel

```
resultat=calculerPrixTTC(13)  
print("le prix ttc est "+ str(resultat)+" euros")
```

autre appel

```
resultat=calculerPrixTTC(int(input("entrer un prix : ")))
```