

NOTIONS FONDAMENTALES – 4 - FONCTIONS

RESUME DE COURS - EXERCICES

Version sept 2018

6. ++ Fonction et procédure

++ La notion de fonction – paramètre en entrée

Ecrire une **fonction** qui calcule la surface d'un rectangle et un programme qui permet de l'utiliser.

```
surface (larg , long) // Réel
// E : larg : Réel // la largeur
// E : long : Réel // la longueur
Début
    return larg * long;
Fin

Programme afficheSurface
    Lire (larg, long)
    Surf = surface(larg, long) ;
    Afficher (surf) ;
Fin
```

++ La notion de procédure – paramètre en sortie

Ecrire une **procédure** qui calcule la surface d'un rectangle et un programme qui permet de l'utiliser.

```

           ↓   ↓   ↑
surfaceRectangle (larg, long, surf) // pas de retour
// E : larg : Reel // largeur
// E : larg : Reel // longueur
// S : surf : Reel // surface
Début
    surf = larg * long;
Fin

Programme afficheSurface
    Lire (larg, long)
```

```

    surface(larg, long, surf)
    Afficher (surf)
Fin

```

Une procédure peut avoir plusieurs paramètres en sortie.

++ Fonction mixte

Ecrire une **fonction** qui calcule la surface et le périmètre d'un rectangle et un programme qui permet de l'utiliser.

```

                                ↓   ↓   ↑
perimetreEtSurfaceRectangle (larg, long, perimetre) // Retourne la surface
// E : larg : Reel // largeur
// E : larg : Reel // longueur
// S : perimetre: Reel // perimetre
Début
    perimetre = larg * long;
    return larg * long;
Fin

```

++ Utilisation de structure – paramètre en entrée et en sortie

On utilise structure « rectangle » en entrée-sortie contenant la largeur, la longueur et la surface.

```

↓↑
Procédure surfaceRectangle (rectangle) {
    rectangle.surf = rectangle.larg * rectangle.long;
}

```

En-tête de la fonction – paramètres formel - sortie standard – variables en entrée : valeur ne sera pas modifiée - Commentaires de l'en-tête - corps de la fonction – return - paramètres d'appels de la fonction – passage par valeur

En-tête de la procédure – paramètres en sortie - passage par adresse

Boîte noire, tests unitaires

Python de base

Fonction avec 2 paramètres réels en entrée

```
def surface(larg,long): # Réel
# E : larg : Réel // la largeur
# E : long : Réel // la longueur
    return larg * long

# programme
larg=float(input('entrez la largeur : '))
long=float(input('entrez la longueur : '))

surf = surface(larg, long)
print(surf)
```

Fonction avec une structure en entrée-sortie

```
def surfaceStruct(rectangle): # pas de retour
# E : structure rectangle (longueur et largeur)
# S : structure rectange (surface et perimetre)
    rectangle['surface']=rectangle['largeur'] * rectangle['longueur']
    rectangle['perimetre']=2*(rectangle['largeur'] + rectangle['longueur'])

# programme
larg=float(input('entrez la largeur : '))
long=float(input('entrez la longueur : '))

rect={'largeur':larg,'longueur':long}
print(rect)
surfaceStruct(rect)
print(rect)
```

EXERCICES – SERIE 4 – FONCTIONS

Méthode de base d'analyse algorithmique

La méthode de base pour écrire un algorithme suit les 4 étapes suivantes :

1. Comprendre le problème : bien lire le sujet et bien comprendre ce qu'il y a à faire.
2. Lister ce dont on a besoin pour résoudre le problème (les données) et ce qu'on va produire (les résultats) : préciser les Entrées et les Sorties.
3. Trouver un principe de résolution : se donner les grandes lignes, en français, de la méthode de résolution. Eventuellement, se donner des procédures ou des fonctions (des actions générales).
4. Ecrire l'algorithme en détail.

1 - Fonctions et test

Exercice 1 : double

Ecrire une procédure ou une fonction qui renvoie le double d'un entier.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 2 : TTC

Écrire une procédure ou une fonction qui calcule le prix TTC.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 3 : plus grand de 2 nombres

Écrire une procédure ou une fonction qui retourne le plus grand de deux nombres.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 4 : plus grand de 3 nombres

Écrire une procédure ou une fonction qui retourne le plus grand de trois nombres.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 5 : inverse d'un nombre

Écrire une fonction qui calcule l'inverse d'un nombre. Il faut traiter les cas particuliers.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 6 : inverser deux nombres

Écrire un programme qui inverse les valeurs de deux variables

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 7 : photocopieuse

Ecrire une procédure ou une fonction qui permet de calculer le prix d'un certain nombre de photocopies en appliquant le tarif suivant : les 10 premières coûtent 0,50 F pièce, les 20 suivantes 0,30 F pièce et toutes les autres 0,20 F pièce.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 8 : année bissextile

Ecrire une procédure ou une fonction qui renvoie le nombre de jours d'un mois d'une année donnée.

On rappelle qu'une année est bissextile si le millésime est divisible par 4 et n'est pas divisible par 100, ou si le millésime est divisible par 400.

Exemple : 2000 et 2008 étaient des années bissextiles, 1900 n'était pas une année bissextile.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 9 : menu

Ecrire un programme qui permette d'utiliser les fonctions ou procédures des exercices précédents (de 1 à 7). Le programme permettra de choisir parmi ces 7 possibilités. Il permettra aussi de recommencer ou de choisir d'arrêter. A chaque nouveau choix, on repartira avec un écran effacé en utilisant la procédure effaceEcran().

2 - Fonctions et boucles : base

Exercice 1

Ecrire une procédure ou une fonction qui affiche la table de multiplication par n'importe quel entier.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 2

Écrire une procédure ou une fonction qui calcule la somme des n premiers nombres (on utilisera une boucle).

Écrire une variante qui calcule la somme des n premiers nombres pairs.

Ecrire un programme qui teste la procédure ou la fonction écrite.

Exercice 3

Ecrire une procédure ou une fonction pour déterminer si un nombre est premier. Un nombre premier est un nombre qui ne se divise que par 1 et par lui même. Autrement dit, il n'a pas de diviseur. 17 est un nombre premier. 15 est divisible par 3 et n'est donc pas un nombre premier.

Ecrire un programme qui teste la procédure ou la fonction écrite.

3 - Fonctions et boucle : avancé

Reprendre les exercices avancés du chapitre sur les boucles :

- 1) Pour chaque exercice, écrivez une fonction qui traite le problème.
- 2) Ecrire un programme qui utilise cette fonction.

4-Branché maths

Reprendre les exercices « branchés maths » des chapitres sur les tests et sur les boucles.

- 1) Pour chaque exercice, écrivez une fonction qui traite le problème.
- 2) Ecrire un programme qui utilise cette fonction.