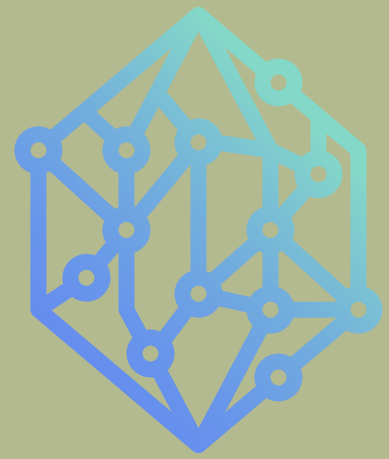


Les sous- algorithmes



Définition



Un sous-algorithme est une séquence d'instructions nommée et éventuellement paramétrée, que l'on définit pour la réutiliser dans un algorithme principal.

Son but est de rendre les algorithmes plus lisibles, modulaires et réutilisables.

Les paramètres



Un paramètre formel: utilisée dans la définition du sous-algo

Un paramètre effectif : utilisée lors de l'appel du sous-algorithme

Type de paramètres	Sens	mode de passage
Donnée(entrée)	donnée fournie au sous-algo	Par valeur
Résultat (sortie)	donnée renvoyée après exécution	Par référence
Donnée/Résultat	donnée d'entrée modifiée puis renvoyée	Par référence

Interet des sous-algorithmes :

Eviter les répétitions



Améliorer la lisibilité

Faciliter la maintenance et la réutilisation

Introduire la modularité et l'abstraction

Structure d'un sous algorithme

Il y a le profil (=entête) qui donne le nom, les paramètres et le type de retour éventuel et puis il y a le corps qui contient les déclarations locales et les instructions.

Portée des variables

Eviter les variables globales, cela réduit la réutilisation du sous algorithme



(Les variables globales sont des variables utilisables dans tout l'algorithme principal ex: ne pas utiliser la variable a dans le sous algo et dans l'algo)

Seul et unique condition pour appeler le sous algo

Pour appeler le sous algo il faut que la variable dans laquelle on utilise le sous algo soit du meme type que le sous algo



ex: (on va supposer que le sous algo renvoie un entier et que son nom est sousalg et que a est un entier)

`a<- sousalg(x)`

À retenir

Un sous algo a une seule tâche claire.



Les paramètres peuvent être en valeur (copie) ou référence (modifié).

Toujours commenter et structurer les sous algorithmes

favoriser la réutilisation et la modularité