Procédures et fonctions

Introduction

La programmation utilise des procédures et des fonctions pour organiser et structurer le code. Ces concepts permettent de diviser un programme en morceaux plus petits et plus gérables, facilitant ainsi la lecture, la maintenance et la réutilisation du code.

Principes

Abstraction : Les procédures et fonctions masquent les détails non pertinents, permettant aux développeurs de se concentrer sur des tâches à un niveau plus élevé.

Réutilisation : Une fois définies, les procédures et fonctions peuvent être réutilisées dans différents programmes ou parties du code.

Modularité : Elles favorisent un code organisé, où différentes parties du programme peuvent être développées et testées indépendamment.

Arguments - Paramètres

Arguments : Ce sont les valeurs passées à une fonction ou une procédure lors de son appel.

Paramètres : Ce sont les variables définies dans la déclaration de la fonction ou de la procédure qui reçoivent les valeurs des arguments.

Exemple :

```
def ajouter(a, b):  // a et b sont des paramètres
  return a + b
```

```
resultat = ajouter(5, 3)  // 5 et 3 sont des arguments
```

Fonctions

Les fonctions sont des blocs de code qui effectuent une tâche spécifique et retournent une valeur. Elles peuvent prendre des paramètres et sont souvent utilisées pour effectuer des calculs ou manipuler des données.

Exemple de fonction :

```
def multiplier(x, y):
  return x * y
```

Retour et type de retour

Le retour est la valeur que la fonction renvoie après son exécution. Le type de retour est le type de données de la valeur retournée (par exemple, entier, chaîne, liste).

Exemple :

```
def obtenir_nom():
  return "Alice"  # retourne une chaîne de caractères
```

Procédures

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
https://wiki.siochaptalqper.fr/doku.php?id=bloc1:prog:sousprogrammes&rev=1730644954

Last update: **2024/11/03 15:42**