ALGORITHMES ET PROGRAMMES

https://www.php.net/manual/fr/function.phpinfo.php

AFCI
CENTRE DE FORMATION

Qu'est-ce que la programmation ?

La programmation est avant tout une méthode d'analyse et non l'apprentissage d'un langage.

Pour bien comprendre le fonctionnement d'un programme, il convient de le décomposer en modules indépendants et élémentaires qui se contentent d'effectuer une action spécifique.

En utilisant cette technique, la conception devient aisée, vous faites moins d'erreurs et la maintenance du code est simplifiée.

AFCI
CENTRE DE FORMATION

Ne pas tout réécrire

- Lorsque vous écrivez un code informatique, vous pouvez bien souvent vous appuyer sur des "bibliothèques" pour éviter de tout réécrire.

- Tous les langages proposent des bibliothèques complémentaires pour simplifier vos développements. N'hésitez pas à les utiliser. Votre temps de développement sera réduit d'autant ...

AFCI
CENTRE DE FORMATION

Algorithme :

Le mot algorithme vient d'Al-Khwarizmi, nom d'un mathématicien persan du lX siècle.

Un algorithme est une suite finie et non ambiguë d'instructions et d'opérations permettant de résoudre une problématique.

Un algorithme n'est pas forcément destiné à décrire la solution d'un problème pour la programmation et l'informatique…

- Un algorithme en cuisine s'appelle une recette
- Un algorithme en musique s'appelle une partition
- Un algorithme en tissage s'appelle un point

AFCI
CENTRE DE FORMATION

Algorithme : L’analyse

L’analyse est une phase de réflexion préalable qui permet d’identifier précisément le problème :

- Données à traiter
- Résultats attendus
- Cas particuliers
- Traitements à effectuer
- Etc….

L’analyse permet également de découper le problème en une succession de taches simples à enchaîner pour arriver jusqu’à la résolution.

L’analyse est fondamentale. En effet, la résolution d’un problème ne peut être dissociée de sa compréhension. Mieux vous comprendrez le problème, plus facilement vous le résoudrez.

AFCI
CENTRE DE FORMATION

Algorithme : La structure

La structure d'un algorithme est la suivante :

```
Algorithme nom-de-l'algorithme // partie en-tête
Debut                          // partie traitement
    bloc d'instructions;
Fin
```

Chaque ligne comporte une seule instruction. L'exécution de l'algorithme correspond à la réalisation de toutes les instructions, ligne après ligne, de la première à la dernière, dans cet ordre.

Les commentaires : // ou /* */

// Commentaire

/* j'écris
des commentaires */

AFCI
CENTRE DE FORMATION

Algorithme : La structure

Algorithme permettant d'afficher « Hello World », pour cela , nous avons besoin d'une fonction écrire ("phrase") permettant d'écrire dans le résultat la phrase écrite entre parenthèse.

```
Algorithme : algo-hello
        début
                écrire ("Hello World" );
        fin
```

```
Algorithme : algo-hello2
        début
                écrire ("Hello");
                écrire ("World");
        fin
```

AFCI
CENTRE DE FORMATION

Algorithme : Déclaration et utilisation des variables

La définition d'un algorithme consiste en la mise en œuvre d'actions élémentaires à l'aide d'une notation dédiée :

- Déclaration de variables
- Lecture / Écriture
- Tests
- Boucles

Définition : Variable
Une variable désigne un emplacement mémoire qui permet de stocker une valeur.
Une variable est définie par :

- Un nom unique qui la désigne
- Un type de définition unique
- Une valeur attribuée et modifiée au cours du déroulement de l'algorithme

AFCI
CENTRE DE FORMATION

Algorithme : La syntaxe

```
Algorithme nom-de-l'algorithme      // partie en-tête
variables: indice: entier           // partie des déclarations des variables

Debut                               // partie traitement
    bloc d'instructions;
Fin
```

Ici, une variable nommée indice de type Entier a été déclarée, et pourra donc être initialisée et utilisée dans le bloc d'instructions.

Nom d'une variable – identifiant d'une variable

Le nom d'une variable permet de l'identifier de manière unique au cours de l'algorithme.
Par convention, il convient de respecter des règles (inspirées du langages Java) pour

Nommer les variables.

- Le nom d'une variable commence par une minuscule
- Le nom d'une variable ne comporte pas d'espace
- Si le nom de la variable est composé de plusieurs mots, il faut faire commencer
- chacun d'eux par une majuscule (exemple : laVitesse, valeurMaxOuMin)
- Il faut également faire attention à bien donner aux variables un nom explicite

AFCI
CENTRE DE FORMATION

Algorithme : L'affectation

L'affectation est une instruction qui permet de changer la valeur d'une variable.
Elle modifie le contenu de la variable.

Le symbole de l'affectation est ←

Exemple :carré ← 0 : se lit carré reçoit la valeur 0

Avertissement :

- L'affectation est une instruction qui dite « destructrice »
- L'ancienne valeur de la variable est détruite, écrasée, effacée par la nouvelle valeur !

  n ← 12

  carré ← n < == > Copie de la variable n dans la variable carré qui n'est plus égale à 0

AFCI
CENTRE DE FORMATION

Algorithme : L'affectation

Exemple : Calculer et afficher le double d'un nombre réel donné

Analyse du problème : On nous informe qu'un nombre réel est donné(exemple 7) et qu'un autre nombre réel sera calculé, puis affiché.

La structure de l'algorithme est alors la suivante :

```
Algorithme : double
Variables : nombre, resultat : réel;
    début
        nombre ← 7;
        resultat ← nombre * 2;
        écrire(resultat);
    fin
```

AFCI
CENTRE DE FORMATION

Algorithme : L'affectation

L'algorithme se déroule de manière séquentielle et ligne après ligne, les variables changent parfois de valeurs à mesure du déroulement. Au départ, lors de la déclaration, les valeurs sont inconnues : leur valeur est indiquée par " ? ".

Algorithme : double	
Variables : nombre, resultat : réel;	**Déclaration des variables**
début	nombre = ? resultat=?
nombre ←7;	nombre = 7 resultat=?
resultat ← nombre * 2;	nombre = 7 resultat=14
écrire(resultat);	nombre=7 resultat=14
fin	Les variables n'existent plus

AFCI
CENTRE DE FORMATION

Variables : Les différents types

Les types alphanumériques :

- Le type caractère annoter entre simple quote : ‘c’
- Le type chaine de caractère (String)

Il est possible de concaténer deux ou plusieurs variable de type alphanumériques :

A ← « hello »
B ← « World »
C ← A & B →Résultat: C aura pour valeur « hello World »

Les valeurs types alphanumériques String sont mise en guillemets : « Hello World »

Il est possible de déclarer un ou une série de chiffres comme alphanumériques :
« 1234 » la valeur sera de type alphanumériques et non de type numérique

AFCI
CENTRE DE FORMATION

Variables : Les différents types

Le type booléen : Ne prenant comme valeur uniquement **vrai (TRUE)** ou **faux (FALSE).**
Les opérateurs admissibles sur les éléments de ce type sont réalisées à l'aide de tous les connecteurs logiques.

- **ET** : Pour le « et logique »
- **OU :** pour le « ou logique inclusif » (il est vrai si l'un des deux booléens testés vaut vrai)
- **NON :** pour le « non logique »,inverse une condition < == >**NON**(condition1) est **VRAIE** si condition1 est **FAUX**, et il sera **FAUX** si condition1 est **VRAIE**

La table de vérité donne la réponse **« Vrai »** ou **« Faux »** des opérations logiques

Opération ET	Faux	Vrai
Faux	Faux	Faux
Vrai	Faux	Vrai

Opération OU	Faux	Vrai
Faux	Faux	Vrai
Vrai	Vrai	Vrai

Variables : Constante

Une constante, comme une variable, peut représenter un entier, un réel,
une chaîne de caractères, etc…Toutefois, contrairement à une variable dont la valeur peut-être
Modifiée au cours de l'exécution de l'algorithme, la valeur d'une constante ne varie pas.

Les noms de CONSTANTE doivent être écris en respectant le **SNAKE_CASE**, à savoir :

- Le nom d'une constante doit être en **Majuscule**
- Pas de caractère accentués ou de ponctuation
- Toujours nommer ses Constantes de façon précise
- Ne pas commencer les noms de Constante par des chiffres ou n'utiliser que des chiffres comme nom de Constante
- Si le nom d'une constante est composé de plusieurs mots alors on séparera chaque mots par un underscore : **MA_CONSTANTE**

AFCI
CENTRE DE FORMATION

Les opérateurs

Les opérateurs arithmétiques

Opérateur	Opération
+	L'addition
-	La soustraction
*	La multiplication
:	La division entière
/	La division réelle

Pour calculer le reste du division, il existe l'opérateur modulo : **%**

Les opérateurs de comparaison

Opérateur	Opération
==	Egal à
!=	Différent de
<	Inférieur à
<=	Inférieur ou égal à
>	Supérieur à
>=	Supérieur ou égal à

AFCI
CENTRE DE FORMATION

Les opérateurs

Les opérateurs logiques

Opérateur	Opération
ET - &&	ET logique
OU - ||	OU logique inclusif
NON - !	Négation logique
XOR	OU logique exclusif

Les opérateurs divers

Opérateur	Opération
<-	L'affectation
&	La concaténation

AFCI
CENTRE DE FORMATION

Les opérateurs

- Le test d'égalité « = » s'écrit **==** (ex : x ==5)
- Différent « ≠ » s'écrit **!=** (ex : x != 5)
- Inférieur ou égal « ≤ » et supérieur ou égal « ≥ » s'écrivent **<=** et **>=** (ex : x <= 5)

Ordre des instructions

- L'ordre va de ligne en ligne
- Même si c'est logique, on commence par la première ligne, on l'exécute,
- puis on passe à la suivante, jusqu'à la fin.
- Cela s'appelle la synchronicité

AFCI
CENTRE DE FORMATION

Algorithme

EXERCICES-1

AFCI
CENTRE DE FORMATION