

PSEUDO-CODE

Jason CHAMPAGNE

(<https://www.youtube.com/formationvideo8>)

AVANT-PROPOS

- Aucune règle « officielle » ou norme n'est définie pour écrire du pseudo-code. Il n'y a donc pas de « mauvaise » syntaxe, ni d'écriture meilleure qu'une autre.

Mais dans un soucis de compréhension, je vous invite à adopter les notations proposées sur ce PDF, surtout lorsque vous partagerez des algorithmes avec les autres abonnés, comme sur notre Discord → <https://discord.gg/JfFaWXD>

RÈGLES GÉNÉRALES

- Chaque mot-clé sera écrit en majuscules :
 - FONCTION
 - TANT QUE
 - EST UN ENTIER
- Vous pouvez écrire les mots en français ou en anglais :
 - FONCTION ou FUNCTION
 - TANT QUE ou WHILE
 - FIN SI ou END IF
- Un commentaire s'écrit ainsi :
 - //Ceci est un commentaire

ALGORITHME

ALGORITHME <nom_algorithme>

 <déclarations_variables>

DEBUT

 <instructions>

FIN

AFFICHAGE ET SAISIE

- Afficher du texte, des variables, des données :

`AFFICHER("Bonjour tout le monde")`

`AFFICHER("Tu as ", age_utilisateur, " ans")`

- Demander des informations :

`LIRE(prenom)`

`prenom ← LIRE("Comment t'appelles-tu ?")`

VARIABLES (1/2)

- Types possibles de variables :
 - ENTIER (ex : 14, -138)
 - REEL (ex : 3.14, 126.45)
 - BOOLEEN (Vrai/Faux *ou* True/False)
 - CARACTERE (ex : 'a', 'H')
 - CHAINE (ex : "Bonjour")
 - <TYPE_PERSONNALISE> (ex : JOUEUR, GRAPH)

VARIABLES (2/2)

- Déclaration d'une variable :
 age_utilisateur EST UN ENTIER
 nom_joueur EST UNE CHAINE
- Affectation / modification d'une variable :
 PI ← 3.14
 pseudo ← "Chuck Norris"

CONDITIONS (1/3)

- Condition simple :

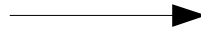
SI <condition> ALORS

<instructions du si...>

SINON

<instructions du sinon...>

FIN SI



SI age_utilisateur > 18 ALORS

AFFICHER("Tu es majeur")

SINON

AFFICHER("Tu es mineur")

FIN SI

CONDITIONS (2/3)

- Condition avancée :

SI <condition> ALORS

<instructions du si...>

SINON SI <condition> ALORS

<instructions du sinon si...>

SINON

<instructions du sinon...>

FIN SI



SI nombre < 0 ALORS

AFFICHER("Plus petit que 0")

SINON SI nombre > 100 ALORS

AFFICHER("Plus grand que 100")

SINON

AFFICHER("Entre 0 et 100")

FIN SI

CONDITIONS (3/3)

- Condition multiple :

SELON <variable>

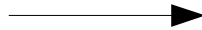
<valeur_1> : <instructions>

<valeur_2> : <instructions>

<etc> : <instructions>

AUTREMENT : <instructions>

FIN SELON



SELON choixCalcul

1 : addition(nb1, nb2)

2 : soustraction(nb1, nb2)

3 : quitter_programme()

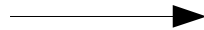
AUTREMENT : <instruction défaut>

FIN SELON

BOUCLES (1/3)

- Boucle simple :

```
TANT QUE <condition>  
    <instructions>  
FIN TANT QUE
```



```
TANT QUE i < 10  
    AFFICHER("i = ", i)  
    i ← i + 1  
FIN TANT QUE
```

BOUCLES (2/3)

- Boucle alternative :

REPETER

<instructions>

TANT QUE <instructions>



REPETER

AFFICHER("i = ", i)

$i \leftarrow i + 1$

TANT QUE $i < 10$

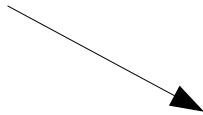
BOUCLES (3/3)

- Boucle de parcours :

POUR <variable> ALLANT DE <min> A <max> [SAUT DE <saut>]

<instructions>

FIN POUR



POUR i ALLANT DE 0 A 10 [SAUT DE 1]

AFFICHER("i = ", i)

FIN POUR

FONCTIONS (1/2)

- Procédure (ne renvoie pas de résultat) :

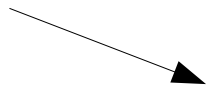
PROCEDURE <nom> (parametres...)

<declarations_variables>

DEBUT

<instructions>

FIN



PROCEDURE addition(nb1, nb2)

res **EST UN ENTIER**

DEBUT

res ← nb1 + nb2

AFFICHER("resultat = ", res)

FIN

FONCTIONS (2/2)

- Fonction (renvoie un résultat) :

FONCTION <nom> (parametres...) : <type_retour>

<declarations_variables>

DEBUT

<instructions>

FIN

FONCTION addition(nb1, nb2) : **ENTIER**

res **EST UN ENTIER**

DEBUT

res ← nb1 + nb2

RETOURNER res

FIN



ANNEXE

- Collections vides (tableau, liste, ...) :
 tab_notes EST UN TABLEAU D'ENTIERES
 liste_joueurs EST UNE LISTE DE JOUEURS
- Collections pré-remplies :
 tab_nombres EST UN TABLEAU D'ENTIERES [14, 28, 366]
 inventaire EST UNE LISTE D'ITEMS ["Épée", "Potion de soin"]