Choisir et écrire une instruction

Mémo • Modules 1 à 10 • BTS SIO1

Commencez par le besoin, puis choisissez la construction. Le pseudo-code décrit l'idée ; Python exige aussi des deux-points et une indentation de quatre espaces. Dans ce mémo, les bornes du pseudo-code Pour sont incluses.

Mémoriser afficher et comparer

Besoin	Pseudo-code	Python
Mémoriser	stock ← 3	stock = 3
Ajouter 1	stock ← stock + 1	stock = stock + 1
Afficher une valeur	Afficher stock	print(stock)
Afficher du texte	Afficher "Prêt"	print("Prêt")
Tester une égalité	stock = 3	stock == 3
Tester une différence	stock != 3	stock != 3
Comparer un seuil	stock <= 3	stock <= 3
Combiner deux tests	stock > 0 Et stock < 3	stock > 0 and stock < 3

Choisir selon une condition

```
Si stock = 0 Alors
    Afficher "Vide"
Sinon Si stock <= 2 Alors
    Afficher "Faible"
Sinon
    Afficher "Suffisant"
Fin Si
```

```
if stock == 0:
    print("Vide")
elif stock <= 2:
    print("Faible")
else:
    print("Suffisant")
```

Les tests sont examinés dans l'ordre. Une seule branche est exécutée. else traite les autres cas. Attention : = affecte une valeur ; == compare deux valeurs en Python.

Répéter un nombre connu de fois

Pseudo-code	Python	Valeurs de i
Pour i de 0 à 3 Faire	for i in range(4):	0, 1, 2, 3
Pour i de 1 à 4 Faire	for i in range(1, 5):	1, 2, 3, 4
Pour i de 2 à 8 par pas de 2 Faire	for i in range(2, 9, 2):	2, 4, 6, 8

En Python, la borne finale de range est exclue. Le pas vaut 1 s'il est omis. Le bloc indenté est répété ; l'instruction qui revient à l'alignement du for est exécutée après la boucle.

Parcourir répéter et renvoyer un résultat

Lire et modifier des listes ou des chaînes

Besoin	Pseudo-code	Python
Créer une liste	codes ← ["PC", "IMP"]	codes = ["PC", "IMP"]
Lire le premier élément	codes[0]	codes[0]
Remplacer un élément	codes[1] ← "BOX"	codes[1] = "BOX"
Ajouter à la fin	Ajouter "NAS" à codes	codes.append("NAS")
Connaître la longueur	Longueur(codes)	len(codes)
Parcourir les valeurs	Pour Chaque code dans codes Faire	for code in codes:
Parcourir les indices	Pour i de 0 à Longueur(codes) - 1 Faire	for i in range(len(codes)):
Parcourir les caractères	Pour Chaque c dans texte Faire	for c in texte:
Construire une chaîne	resultat ← resultat + c	resultat = resultat + c

Les indices commencent à 0. Pour une longueur n, le dernier indice est n - 1. Une liste vide [] et une chaîne vide "" ont une longueur nulle ; les parcours ci-dessus font alors zéro passage. Une chaîne ne se modifie pas par texte[i] = … : on peut en construire une nouvelle.

Répéter tant qu'une condition est vraie

```
stock ← 1
Tant Que stock < 3 Faire
    stock ← stock + 1
Fin Tant Que
Afficher stock
```

```
stock = 1
while stock < 3:
    stock = stock + 1
print(stock)
```

La condition est testée avant chaque passage. Vérifiez qu'une mise à jour permet de la rendre fausse. Le nombre de passages peut être nul.

Définir appeler et utiliser une fonction

```
Fonction suivant(nombre : entier) : entier
    Retourner nombre + 1
Fin Fonction
Afficher suivant(3)
```

```
def suivant(nombre):
    return nombre + 1

print(suivant(3))
```

Un paramètre reçoit la donnée de l'appel. return renvoie un résultat et quitte immédiatement la fonction ; print affiche. Ici, l'appel renvoie 4, puis print affiche 4.