Choisir entre plusieurs situations

Module 3 • 60 minutes • BTS SIO1 • Nom : _____ Date : _____

Votre mission : donner le bon état du stock de claviers. Vous allez consolider if / else, découvrir elif et vérifier les valeurs qui séparent les cas. Les stocks étudiés sont des entiers positifs ou nuls.

Réactiver les acquis 5 min

Après stock ← 6 puis stock ← stock - 3, stock vaut _____.
La condition stock <= 3 est _____. Comment afficher la valeur de stock en Python ?

.....

Comprendre trois cas 10 min

Règle du service : stock nul → « Rupture » ; stock de 1 à 3 → « À surveiller » ; stock supérieur à 3 → « Stock suffisant ». Lisez sans exécuter.

```
stock ← 2
Si stock = 0 Alors
    Afficher "Rupture"
Sinon Si stock <= 3 Alors
    Afficher "À surveiller"
Sinon
    Afficher "Stock suffisant"
Fin Si
```

Les tests se lisent dans l'ordre. Dès qu'un test est vrai, sa branche est exécutée et les autres sont ignorées. Si aucun test n'est vrai, on exécute Sinon.

Stock	Premier test vrai ou Sinon	Message affiché
0		
1		
3		
4		

Pourquoi stock = 0 ne doit-il pas afficher « À surveiller » ?

.....

Compléter les tests 10 min

Conservez la règle ci-dessus. Complétez ce pseudo-code sans ajouter de test.

```
Si stock ______ 0 Alors
    Afficher "Rupture"
Sinon Si stock ______ 3 Alors
    Afficher "À surveiller"
Sinon
    Afficher "Stock suffisant"
Fin Si
```

Vérifiez les frontières : stock = 0, stock = 3, stock = 4. Aides et mémo : page 3.

Traduire puis vérifier la règle

Traduire en Python 10 min

Traduisez en Python le programme de la page 1. Sinon Si devient elif. Prévoyez puis testez les messages pour les stocks 0, 1, 3 et 4, en changeant la première ligne.

.....

.....

.....

.....

.....

.....

.....

.....

Déboguer un ordre de tests 10 min

Ce programme s'exécute mais n'applique pas la règle du service pour un stock nul.

```
stock = 0
if stock <= 3:
    print("À surveiller")
elif stock == 0:
    print("Rupture")
else:
    print("Stock suffisant")
```

1. Quel message affiche-t-il ? Pourquoi le deuxième test n'est-il pas utilisé ?
2. Corrigez l'ordre des tests et des messages associés. Retestez 0, 3 et 4.

.....

.....

.....

Créer une décision 12 min

Le service classe les demandes de prêt. Pour une demande de 0 matériel, afficher « Aucune demande » ; de 1 à 2, « Traitement direct » ; au-delà de 2, « Validation nécessaire ». Écrivez le pseudo-code puis Python sur le dos de la feuille ou dans votre éditeur. Testez 0, 2 et 3. Les demandes sont des entiers positifs ou nuls.

Notez ici les trois messages obtenus :

.....

.....

Bilan individuel 3 min

Dans une chaîne if / elif / else, combien de branches sont exécutées ? _____

Pour tester une égalité en Python, j'écris _____.

Je sais expliquer l'ordre de mes tests : oui / avec aide / pas encore.

Mémo et aides de la fiche 3

Comparer et choisir

Intention	Pseudo-code	Python
Affecter 0 à stock	stock ← 0	stock = 0
Tester une égalité	stock = 0	stock == 0
Tester un seuil inclus	stock <= 3	stock <= 3
Ajouter un cas	Sinon Si … Alors	elif …:

if, elif et else sont alignés. Chaque branche est décalée de quatre espaces et commence après deux-points. On peut utiliser elif pour choisir entre plusieurs cas sans imbriquer des if.

Rappel sur les frontières

0 est inclus dans « positif ou nul ». « De 1 à 3 » inclut 1 et 3. Après avoir écarté 0, le test stock <= 3 suffit ici : les stocks négatifs ne font pas partie des données étudiées.

Aides à consulter dans cet ordre

Activité	Aide 1	Aide 2
Comprendre	Testez les questions de haut en bas.	Pour 0, le premier test est vrai : on s'arrête à cette branche.
Compléter	Il faut distinguer nul, petit stock et grand stock.	Premier trou : = en pseudo-code ; deuxième : <=.
Traduire	Reprenez le tableau ci-dessus.	Égalité : ==. Sinon Si : elif. Gardez les deux-points.
Déboguer	Quelle condition accepte déjà la valeur 0 ?	Placez stock == 0 et son message avant stock <= 3.
Créer	Traitez d'abord l'absence de demande.	Testez demande == 0, puis demande <= 2.

Aide 3 pour la traduction

À consulter seulement après un premier essai : voici le pseudo-code complet. Traduisez-le, puis expliquez chaque test à votre voisin.

```
stock ← 2
Si stock = 0 Alors
    Afficher "Rupture"
Sinon Si stock <= 3 Alors
    Afficher "À surveiller"
Sinon
    Afficher "Stock suffisant"
Fin Si
```

Défi facultatif : page 4. Son aide complète se trouve en bas de cette page 4 ; masquez-la pendant votre premier essai.

Défi final de la fiche 3

Facultatif • Environ 10 minutes • Prévoir une réservation

Une salle propose 8 places. Pour une demande supérieure aux places disponibles, refuser et garder les places. Si la demande est égale aux places, accepter et annoncer « salle complète ». Sinon, accepter. Après une acceptation, retirer les places réservées. Afficher le nombre de places restant dans tous les cas. La demande est un entier strictement positif.

Compléter puis traduire

```
places ← 8
demande ← 8
Si __________ Alors
    Afficher "Réservation refusée"
Sinon Si __________ Alors
    places ← __________
    Afficher "Réservation acceptée, salle complète"
Sinon
    places ← __________
    Afficher "Réservation acceptée"
Fin Si
Afficher __________
```

Traduisez en Python sur le dos ou dans votre éditeur. Testez des demandes de 3, 8 et 9, en repartant chaque fois de 8 places.

Demande	Message prévu	Places restantes
3		
8		
9		

Aides graduées pour le défi

Aide 1 : commencez par repérer le cas où l'on refuse.

Aide 2 : comparez demande à places ; retirez demande seulement si la réservation est acceptée.

Aide 3 Pseudo-code complet

```
places ← 8
demande ← 8
Si demande > places Alors
    Afficher "Réservation refusée"
Sinon Si demande = places Alors
    places ← places - demande
    Afficher "Réservation acceptée, salle complète"
Sinon
    places ← places - demande
    Afficher "Réservation acceptée"
Fin Si
Afficher places
```