Variables et affectations

Module 1 • 60 minutes • Fiche élève • Nom : _____ Date : _____

Votre mission : suivre le stock de claviers du service informatique. À la fin, vous saurez suivre les valeurs de variables et écrire un programme court en pseudo-code et en Python.

Organisation : cherchez d'abord sur papier, comparez avec un voisin, puis testez en Python.
Vous avez à votre disposition un memo et des aides à la page 3.

Comprendre une affectation 10 min

Une variable est un nom associé à une valeur. stock ← 8 signifie : stock reçoit la valeur 8. Une nouvelle affectation remplace l'ancienne valeur. Les instructions sont exécutées de haut en bas.

```
stock ← 8
prets ← 3
stock ← stock - prets
Afficher stock
```

1. Complétez la trace sans exécuter le programme. « Non définie » signifie qu'aucune valeur n'a encore été affectée.

Après l'instruction	stock	prets
stock ← 8		Non définie
prets ← 3		
stock ← stock - prets		

2. Quelle valeur est affichée ? Pourquoi stock change-t-il ?

.....

.....

Compléter un algorithme 12 min

On dispose de 10 claviers. On en prête 4, puis on en récupère 2. Complétez avec des noms de variables ou des expressions.

```
stock ← 10
prets ← 4
retours ← 2
stock ← stock - _____
stock ← stock + _____
Afficher _____
```

Valeur finale de stock : _____. Expliquez les deux mises à jour.

.....

.....

Écrire puis vérifier

Traduire en Python 10 min

Traduisez en Python, l'algorithme **que vous venez de completer**. Préservez l'ordre des instructions. Avant de lancer le programme, prévoyez le résultat, puis comparez.

.....

.....

.....

.....

.....

.....

Résultat prévu : _____ Résultat obtenu ou vérifié à la main : _____

Déboguer 10 min

L'intention est de retirer 3 claviers d'un stock de 8 et d'afficher le stock restant. Ce programme s'exécute, mais ne produit pas le résultat attendu.

```
stock = 8
prets = 3
stock = prets
print("stock")
```

Repérez et corrigez les deux lignes fautives. Expliquez chaque correction.

.....

.....

.....

.....

Créer un programme court 10 min

Il reste 6 souris au service. Une livraison en apporte 5 et 2 sont distribuées. Choisissez des noms de variables, écrivez le pseudo-code, puis Python. Vérifiez la valeur finale. Utilisez le dos de la feuille si nécessaire.

.....

.....

.....

.....

.....

.....

Bilan individuel 3 min

Sans aide : après a ← 4 puis a ← a + 1, a vaut _____.
Je sais expliquer une affectation : oui / avec aide / pas encore.
Une question qui me reste : _____

Mémo et aides de la séance 1

Passer du pseudo-code à Python

Intention	Pseudo-code	Python
Mémoriser un nombre	stock ← 8	stock = 8
Retirer une quantité	stock ← stock - 3	stock = stock - 3
Afficher une valeur	Afficher stock	print(stock)
Afficher un texte	Afficher "Stock"	print("Stock")

Dans stock = stock - 3, Python calcule d'abord la partie droite avec l'ancienne valeur de stock, puis mémorise le résultat dans stock. Ce signe = sert à affecter une valeur.

Le seul rappel de calcul nécessaire

Ajouter augmente le stock ; soustraire le diminue. Exemple distinct des exercices : 7 objets, 2 prêts → 7 - 2 = 5 objets. Ne faites qu'une opération par ligne. Aucun pourcentage n'est nécessaire.

Aides graduées

Prenez le niveau 1. Si vous bloquez encore après une minute, passez au niveau 2, puis au niveau 3. Refermez l'aide avant de reprendre votre réponse.

Activité	Niveau 1	Niveau 2	Niveau 3
Comprendre	Suivez une ligne à la fois.	Recopiez les valeurs inchangées.	À la 3e ligne, calculez 8 - 3.
Compléter	Repérez ce qui sort et entre.	On retire prets et on ajoute retours.	Première case : prets. Dernière case : stock.
Traduire	Utilisez le tableau du mémo.	← devient = ; Afficher devient print(...).	Début : stock = 10. Fin : print(stock).
Déboguer	Vérifiez calcul et affichage.	La ligne 3 doit soustraire. Les guillemets créent du texte.	Écrivez stock = stock - prets puis print(stock).
Créer	Décrivez départ, entrée, sortie.	Utilisez stock, livraison, distribution.	Commencez par stock = 6, puis ajoutez et retirez.

Pour aller plus loin si vous avez terminé : que valent a et b après a ← 2 ; b ← a ; a ← 9 ? Expliquez pourquoi b ne change pas automatiquement.

Défi final de la séance 1

Facultatif • Pour les plus rapides • Environ 8 minutes

Garder une mémoire du stock de départ

Le service possède 9 casques. Il en prête 3, puis en récupère 1. À la fin, le programme doit afficher le stock de départ et le stock actuel, avec deux variables différentes.

1. Écrivez le pseudo-code, puis Python. Utilisez une variable stock et une variable stock_depart. Mémorisez le départ avant de modifier stock.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Prévoyez les deux valeurs affichées, puis vérifiez : _____

3. Si vous changez seulement le stock initial de 9 à 12, quelles valeurs devez-vous obtenir ? Pourquoi stock_depart garde-t-il sa valeur pendant les prêts et les retours ?

.....

.....

.....

Un coup de pouce si nécessaire

Une affectation copie la valeur disponible à cet instant. Essayez stock_depart ← stock juste après avoir fixé le stock initial. Aucun calcul nouveau n’est nécessaire.