

Les boîtes étiquetées

NSI – Première

Activité d'ouverture – Séquence 1

Matériel, par binôme : 3 post-its (ou petits bouts de papier), un crayon à **papier** muni d'une gomme (c'est important : on doit pouvoir effacer une valeur pour en écrire une nouvelle).

Activité : Exécuter une suite d'instructions, à la main

Posez deux post-its côte à côte sur la table. Étiquetez le premier **a** et le second **b** (écrivez le nom en haut du post-it, il ne changera plus). Le corps du post-it restera vide tant qu'aucune valeur n'a été écrite dessus.

Exécutez, une par une et dans l'ordre, les instructions suivantes. À chaque instruction, mettez à jour le post-it concerné : **effacez** l'ancienne valeur si besoin, puis écrivez la nouvelle.

```
1 a = 5
2 b = 3
3 a = a + b
4 b = a - b
```

1. Après la ligne 1, qu'est-il écrit sur le post-it **a** ? Et sur **b** ?
2. Avant d'exécuter la ligne 3, notez bien les valeurs actuelles de **a** et **b**. Que devez-vous faire concrètement avec le post-it **a** pour exécuter cette ligne ?
3. À la ligne 4, quelle valeur de **a** utilisez-vous : celle d'avant la ligne 3, ou celle d'après ? Justifiez à partir de ce que vous avez physiquement écrit sur le post-it.
4. Donnez les valeurs finales de **a** et de **b** après les quatre lignes.

Activité : Le problème de l'échange

Effacez tout et recommencez avec **a = 5** et **b = 3** écrits sur vos deux post-its.

Léa veut échanger les valeurs de **a** et **b**, pour obtenir à la fin **a = 3** et **b = 5**. Elle propose ce programme :

```
1 a = b
2 b = a
```

1. Exécutez ces deux lignes à la main avec vos post-its, en effaçant et réécrivant à chaque instruction comme précédemment.
2. Quelles sont les valeurs finales de **a** et **b** ? Le programme de Léa fait-il ce qu'elle voulait ?
3. À quel moment précis l'information a-t-elle été perdue ? (Quelle valeur avez-vous effacée, et n'avez-vous jamais réutilisée ailleurs ?)
4. À l'aide d'un **troisième** post-it, que vous étiquetterez comme vous le souhaitez, proposez une suite de trois instructions qui réalise vraiment l'échange. Vérifiez votre solution à la main, post-it par post-it, avant de l'écrire.

Activité : Bilan

1. En mathématiques, si $a = b$ alors on a aussi $b = a$: l'égalité « fonctionne dans les deux sens ». D'après ce que vous venez d'observer, est-ce le cas de l'instruction `a = b` en programmation ? Expliquez la différence en une ou deux phrases.
2. Pourquoi ne peut-on pas simplement écrire une instruction du type « échange les valeurs de `a` et `b` », comme on le dirait à un camarade, et laisser l'ordinateur se débrouiller ?

Remarque : Ce qu'il faut retenir de cette activité

Une **variable** se comporte comme une boîte étiquetée : elle a un nom fixe, mais son contenu peut être effacé et remplacé au fil du programme. Une instruction comme `a = b` **n'est pas une égalité** : elle recopie la valeur actuelle de `b` dans `a`, en écrasant définitivement ce qu'il y avait avant dans `a`. Comme les instructions s'exécutent **une par une, dans l'ordre exact** où elles sont écrites, l'ordre a une importance cruciale : deux instructions écrites dans un ordre différent peuvent produire un résultat totalement différent, voire perdre une information de façon irréversible.