1 Problématique

Le code 1 lève une erreur.

```
1 def une_fonction():
2       a = 3
3
4 une_fonction()
5 print(a)
```

Code 1 – Variable locale

À l'inverse le code 2 s'exécute sans erreur.

```
1 a = 3
2 def une_fonction():
3     print(a)
4
5 une_fonction()
```

Code 2 – Variable globale

Quelle est la portée d'une variable ?

2 Portée d'une variable

C'est le domaine d'existence d'une variable, c'est à dire la partie du programme où elle peut être utilisée.

2.1 Variable locale

À retenir

Une variable locale n'est utilisable que dans le bloc ou la fonction où elle a été définie.

Activité 1 :

1. Se rendre sur le site http://pythontutor.com/ .
2. Cliquer sur *Start visualizing your code now.*
3. Écrire le code 1 puis cliquer sur *Visualize Execution.*
4. Exécuter le code en *pas à pas* avec le bouton *Next.*
5. Observer la période d'existence de la variable *a* et justifier alors l'erreur d'exécution de la ligne 5.

2.2 Variable globale

Christophe Viroulaud

À retenir

Une variable globale est accessible depuis n'importe quel point du programme.

Activité 2 : Dérouler le code 2 sur *pythontutor* et justifier la bonne exécution.

3 Pièges et bonnes pratiques

3.1 Effet de bord

Les élèves de NSI ont obtenu les notes suivantes lors du dernier devoir. Avant de les enregistrer définitivement dans Pronote, l'enseignant désire simuler des bonifications pour favoriser les élèves le plus en difficultés.

```
1  def simulation_bonification(limite):
2      for i in range(len(notes)):
3          if notes[i] <= limite:
4              notes[i] += 2
5      return notes
6
7  notes=[7,12,8,5,19,10,7,6,1,15,13,8]
8  print(notes)
9  print(simulation_bonification(6))
10 print(simulation_bonification(8))
```

Code 3 – Bonification

Le code 3 effectue une bonification de deux points pour deux cas de figure :

— les élèves ont 6 ou moins,

— les élèves ont 8 ou moins.

Activité 3 :

1. Recopier le code 3 dans un EDI et repérer l'erreur dans les calculs effectués.
2. Copier le code dans *pythontutor* et expliquer cette erreur.

Effet de bord

En première approche nous pouvons retenir qu'une variable mutable est modifiée quand elle est utilisée dans une fonction. Il est souvent difficile de maîtriser les modifications ce qui peut créer des comportements non désirés du programme.

4 Bonnes pratiques

Il faut visualiser une fonction comme un outil autonome, réutilisable dans plusieurs programmes. Elle ne doit donc pas dépendre de variables globales.
On évitera au maximum d'utiliser une variable globale.

Christophe Viroulaud