Fonctions

1. Pourquoi ?
   - décomposer un problème
   - éviter le copier-coller
2. Définir et appeler
   - `def f(x):`
   - appel : `f(3)`
   - paramètres et arguments
3. `return`
   - renvoie un résultat
   - termine la fonction
   - sans `return` : `None`
   - afficher n'est pas renvoyer
4. Paramètres
   - par défaut `def f(x, n=2):`
   - arguments nommés
5. Portée
   - variables locales
   - variables globales
   - une locale masque une globale
6. Documenter et tester
   - `assert`
   - docstring
   - annotations de types