

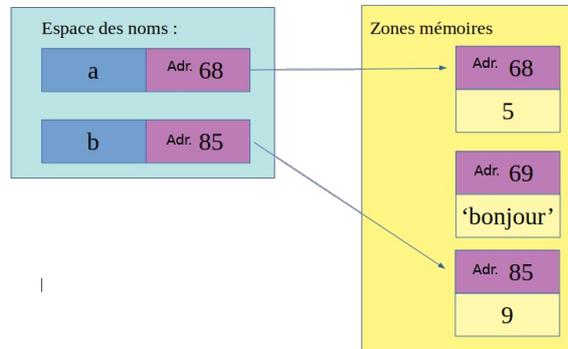
# RÉSUMÉ 6 - PORTÉE DES VARIABLES

Lien vers l'activité : [Portée des variables](#)  
www.infoforall.fr - Dernière modif. : 26 10 2020



## 6.1 ESPACE DES NOMS

En informatique, une variable est en réalité un **alias** vers une adresse-mémoire.



Lorsqu'on utilise une variable dans un programme, l'interpréteur va donc aller lire ce qui est contenu à l'identifiant correspondant. Python ne donne pas directement accès à l'adresse-mémoire réelle mais on peut obtenir un identifiant mémoire à l'aide de la fonction native `id`. Par exemple, on accède à l'identifiant Python d'une variable `a` avec `id(a)`. Si deux variables ont le même identifiant dans Python, c'est qu'elles font bien référence au même contenu-mémoire. Elles sont deux alias vers le même contenu.

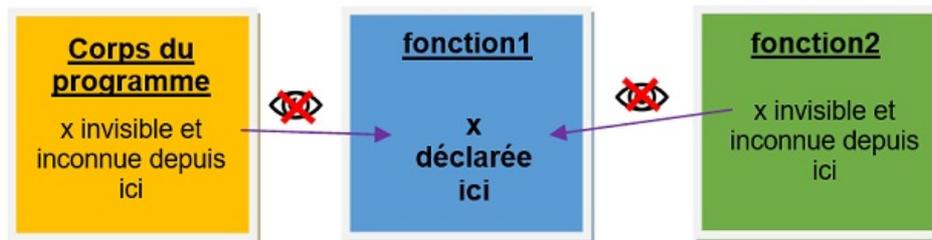
## 6.2 - PORTÉE DES VARIABLES LOCALES

**Variable locale** : variable dont l'affectation est faite dans la fonction. Dans l'exemple ci-dessous, `mot` et `nombre_caracteres` sont des variables locales.

```
1 def compter_lettres() :
2     mot = 'bonjour' # Variable locale
3     nombre_caracteres = len(mot) # Variable locale
```

**Durée d'existence** : une variable locale est détruite lorsqu'on sort de la fonction.

**Portée** : on ne peut ni lire ni modifier une variable locale en dehors de la fonction elle-même.



**Espace des noms** : deux fonctions peuvent contenir les mêmes noms de variables sans aucun problème : l'interpréteur les traite comme des variables différentes grâce à l'**espace des noms**.

```
1 def positif(x) :
2     if x < 0 :
3         x = -x
4     return x
5
6 def pas_negatif(x) :
7     if x < 0 :
8         x = 0
9     return x
10
11 x = -5
12 y = positif(x)
13 z = pas_negatif(x)
```

Ici, il n'y a pas de problème avec le fait que les deux fonctions utilisent des variables ayant le même nom. Chaque fonction n'a accès qu'aux variables qu'elles contiennent. Il ne peut donc pas y avoir de confusion, à part pour un humain qui n'aurait pas compris ce principe. La notion de portée est donc **fondamentale** à comprendre et savoir appliquer.

www.infoforall.fr

