

Python pour les scientifiques

Partie II – Les Références partagées

~ Cyril Desjouy ~

June, 2016

Updated : May 23, 2018



Deux familles d'objets sous Python :

- Les objets **immuables** : objets qui une fois déclarés ne peuvent pas être modifiés (`int`, `float`, `complex`, `bool`, `tuple`, `frozenset`, ...)

Toute méthode qui modifie un objet immuable en modifie une copie à la place

- Les objets **muables** : objets qui une fois déclarés peuvent être modifiés (`list`, `dict`, `set`, `ndarray`, ...)

Lorsqu'un objet muable est modifié, toutes les références à cet objet voient ces modifications

Ce concept de **mutabilité** est illustré par celui de **références partagées**

Les références partagées



Image from Pixabay (Stevepb)

Référence **partagée** à un objet immuable

Interpréteur

```
In : a = 1
```

Espace des variables

Espace des objets

Référence **partagée** à un objet immuable

Interpréteur

```
In : a = 1 ↵
```

Espace des variables

Espace des objets

ref	0	1
type	int	
id	0001	
meth	...	
attr	...	

Référence **partagée** à un objet immuable

Interpréteur

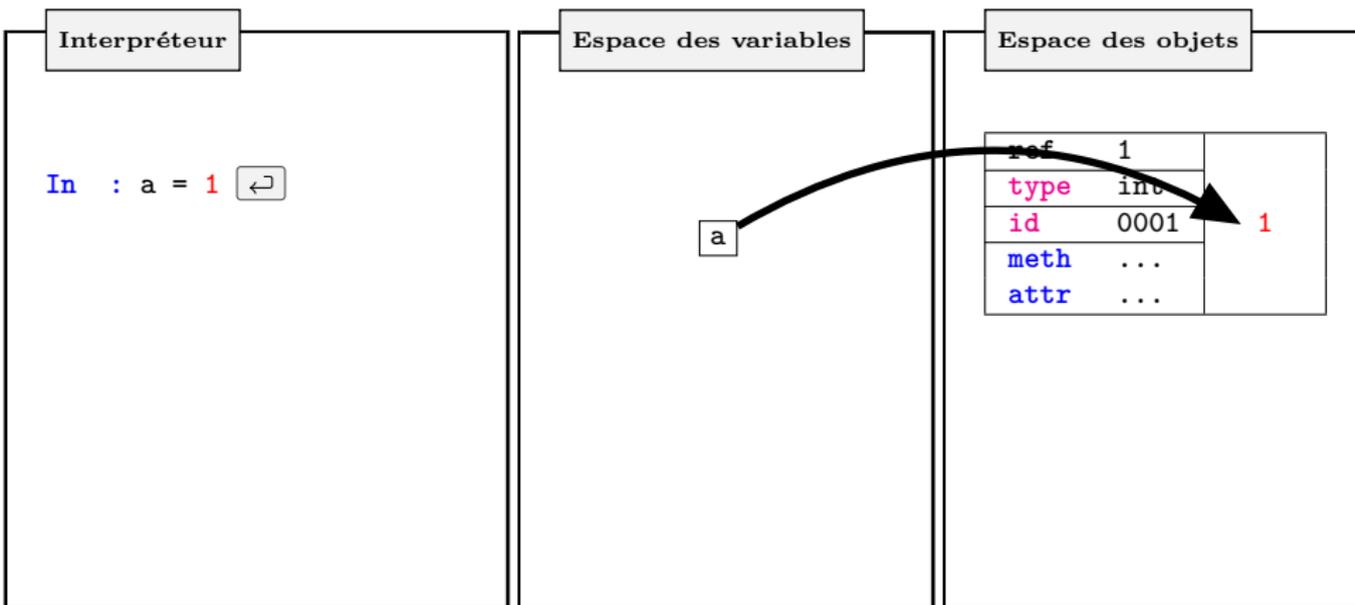
```
In : a = 1 ↵
```

Espace des variables

a

Espace des objets

ref	0	1
type	int	
id	0001	
meth	...	
attr	...	

Référence **partagée** à un objet immuable

Référence **partagée** à un objet immuable

Interpréteur

```
In : a = 1 ↵  
In : b = a
```

Espace des variables

a

Espace des objets

ref	1	1
type	int	
id	0001	
meth	...	
attr	...	

Référence **partagée** à un objet immuable

Interpréteur

```
In : a = 1 ↵  
In : b = a ↵
```

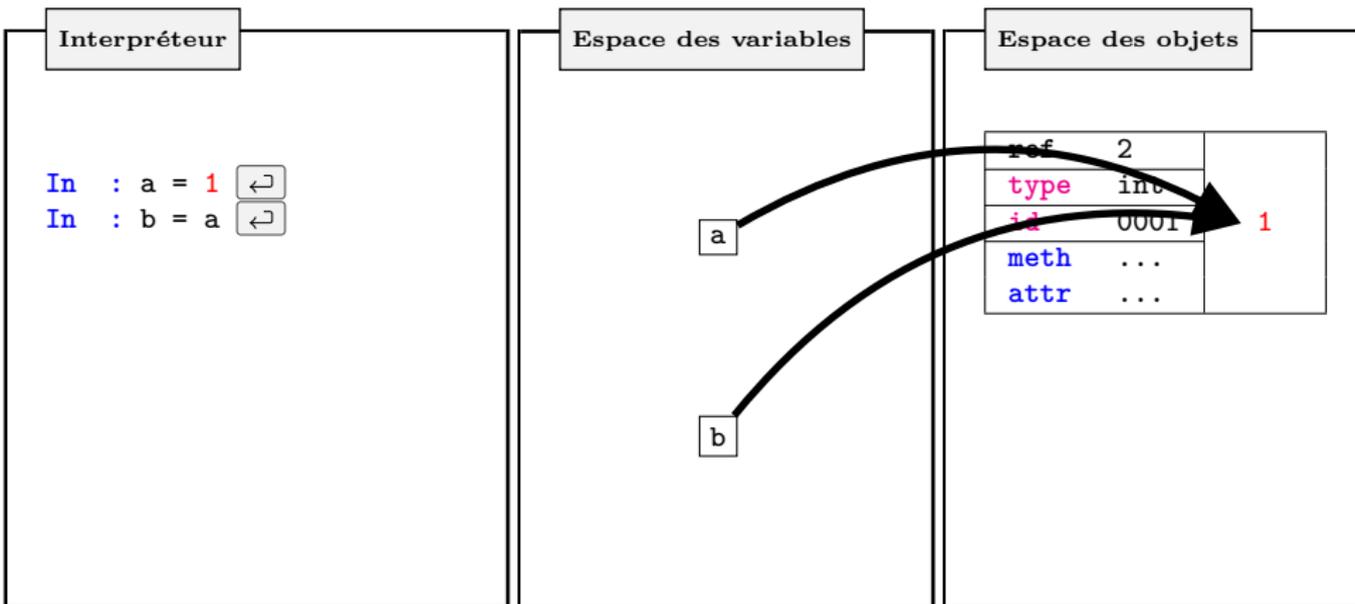
Espace des variables

a

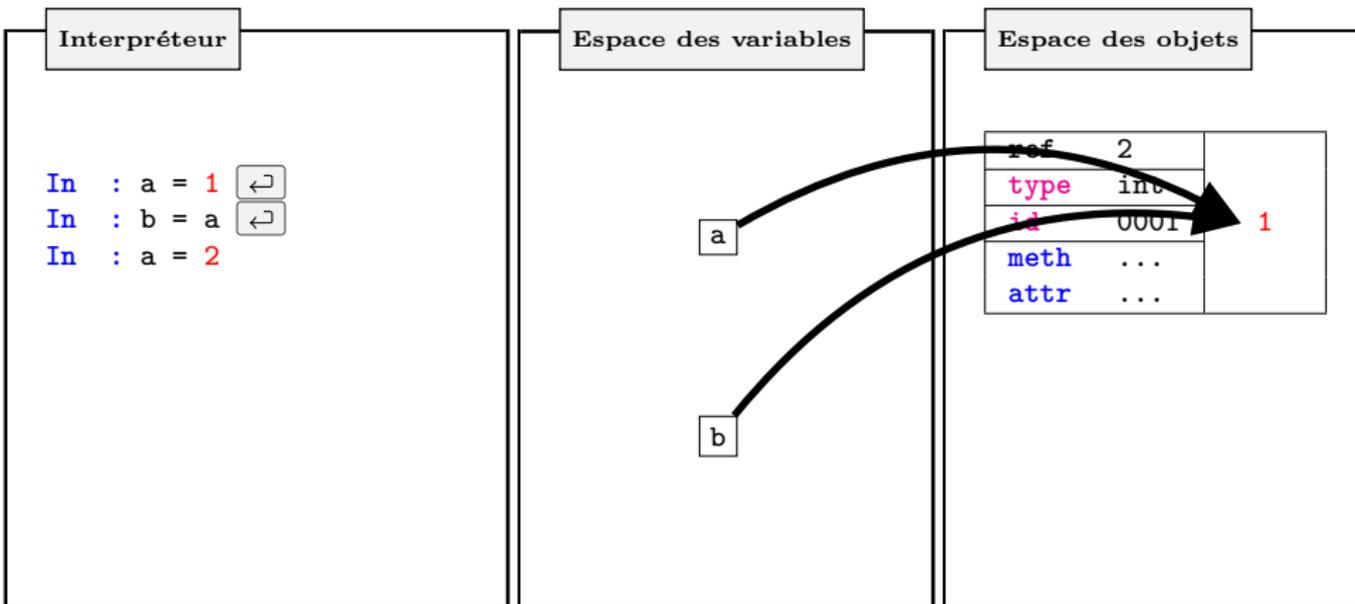
b

Espace des objets

ref	1	1
type	int	
id	0001	
meth	...	
attr	...	

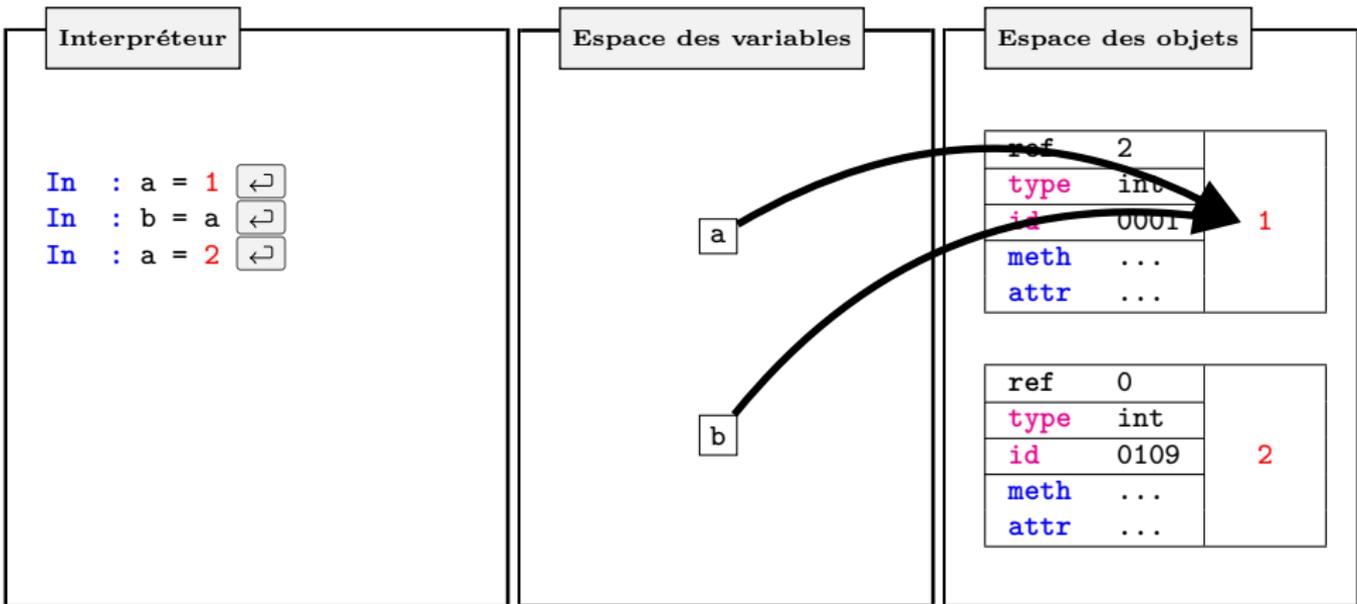
Référence **partagée** à un objet immuable

- Référence partagée : **plusieurs variables réfèrent un même objet**

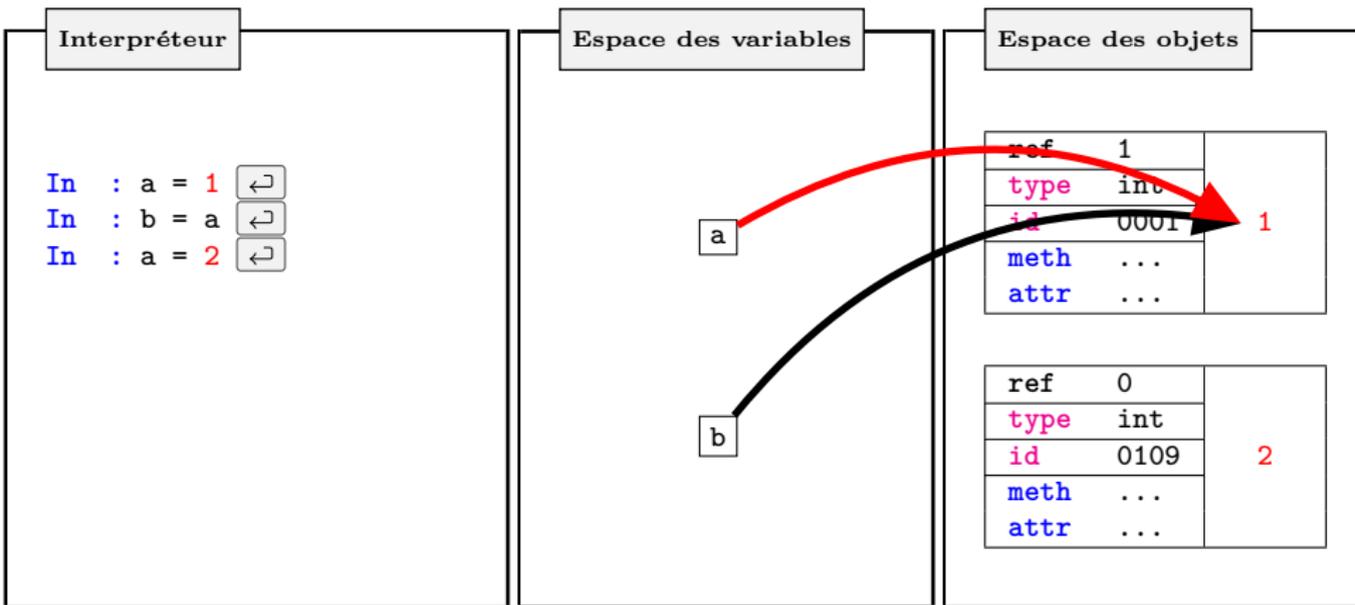
Référence **partagée** à un objet immuable

- Référence partagée : **plusieurs variables réfèrent un même objet**

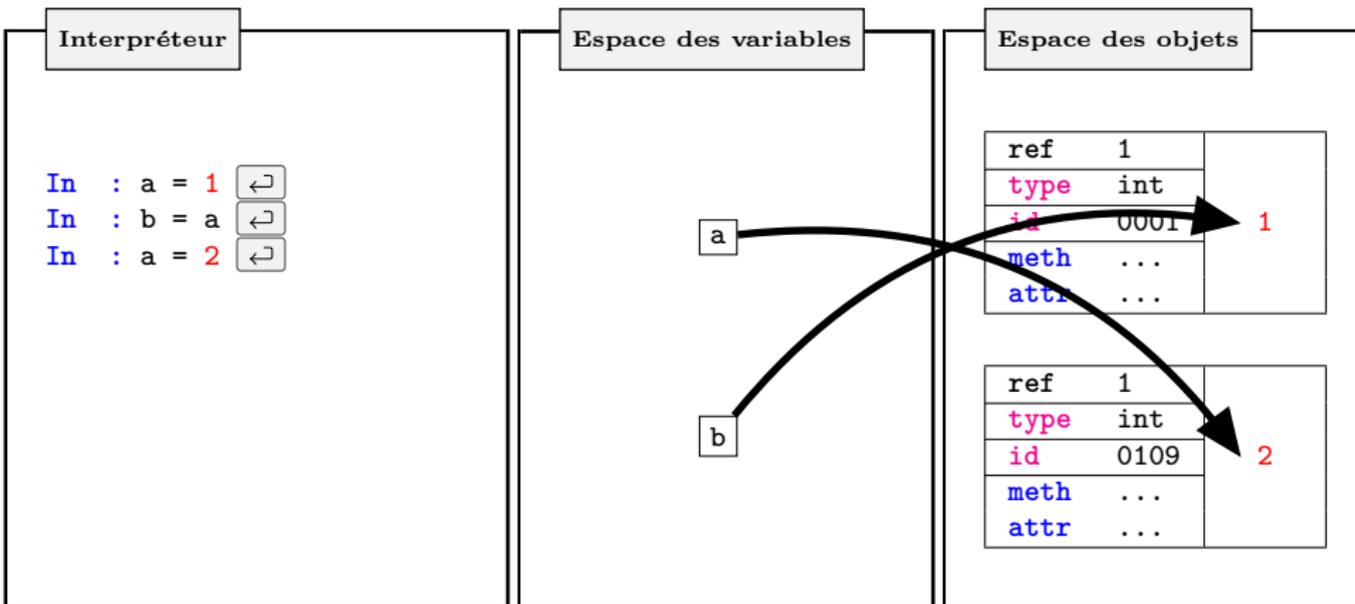
Référence **partagée** à un objet immuable



- Référence partagée : **plusieurs variables référencent un même objet**

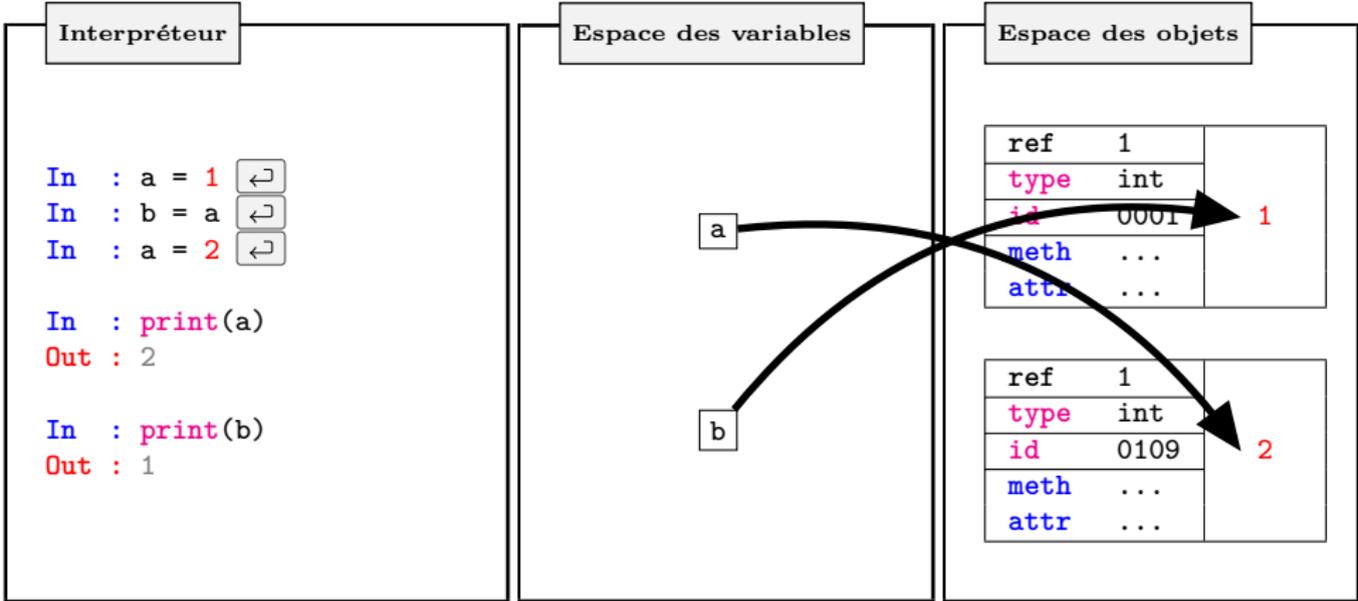
Référence **partagée** à un objet immuable

- Référence partagée : plusieurs variables référencent un même objet

Référence **partagée** à un objet immuable

- Référence partagée : plusieurs variables référencent un même objet

Référence **partagée** à un objet immuable



- **Référence partagée** : plusieurs variables référencent un même objet
- Avec des objets **immuables** : pas d'effets de bord possibles

Référence **partagée** à un objet muable

Interpréteur

```
In : a = [1, 2]
```

Espace des variables

Espace des objets

Référence **partagée** à un objet muable

Interpréteur

```
In : a = [1, 2] ↵
```

Espace des variables

Espace des objets

0	1
int	

0	2
int	

ref	0	[,]
type	list	
id	0001	
meth	...	
attr	...	

Référence **partagée** à un objet muable

Interpréteur

```
In : a = [1, 2] ↵
```

Espace des variables

Espace des objets

1	1	1	2
int		int	

ref	0	[,]
type	list	
id	0001	
meth	...	
attr	...	

Référence **partagée** à un objet muable

Interpréteur

```
In : a = [1, 2] ↵
```

Espace des variables

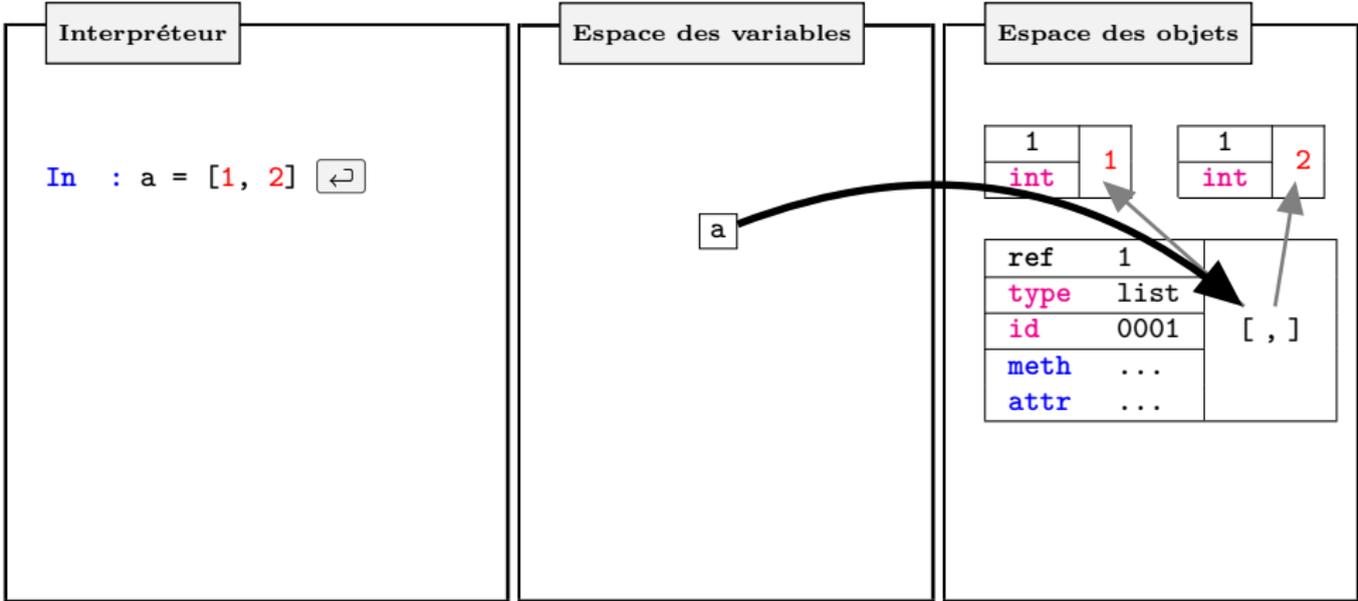
```
a
```

Espace des objets

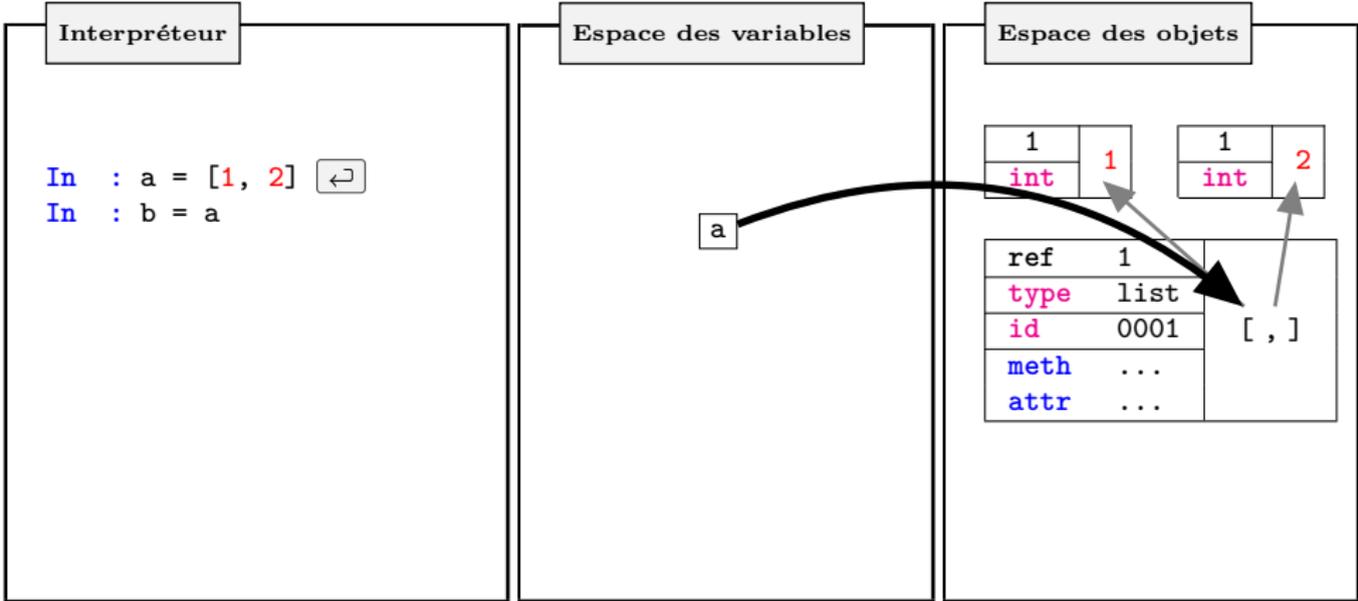
1	1	1	2
int		int	

ref	0	[,]
type	list	
id	0001	
meth	...	
attr	...	

Référence **partagée** à un objet muable



Référence **partagée** à un objet muable



Référence **partagée** à un objet muable

Interpréteur

```
In : a = [1, 2] ↵
In : b = a ↵
```

Espace des variables

a

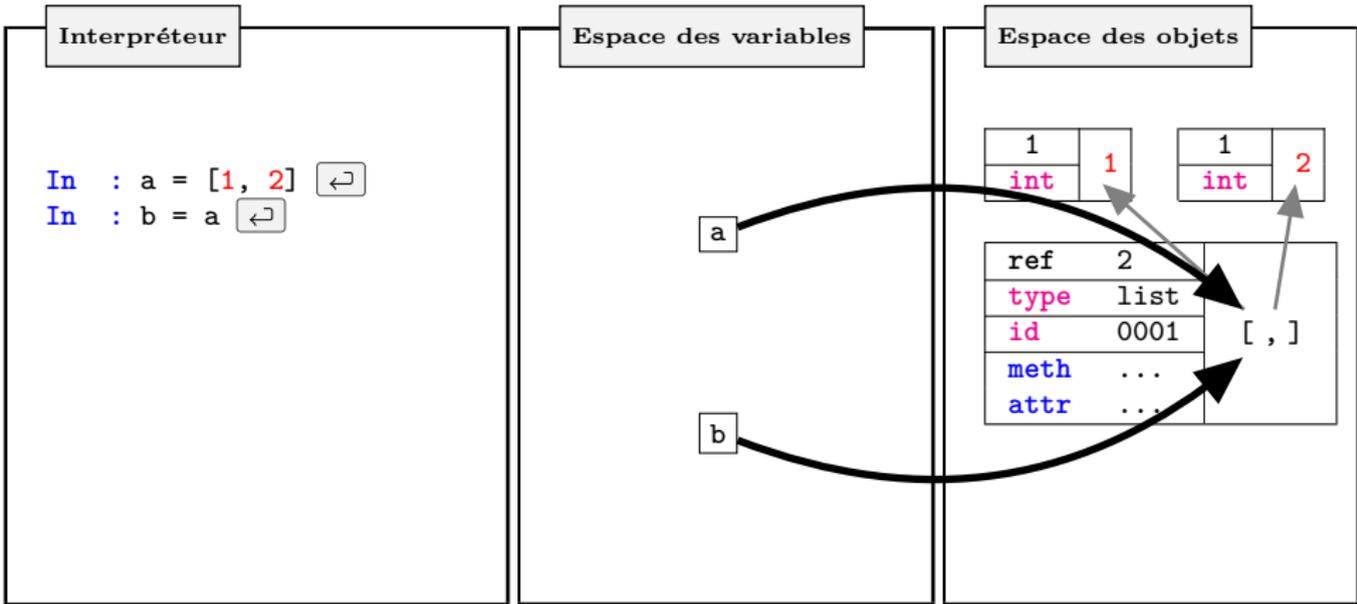
b

Espace des objets

1	1	1	2
int		int	

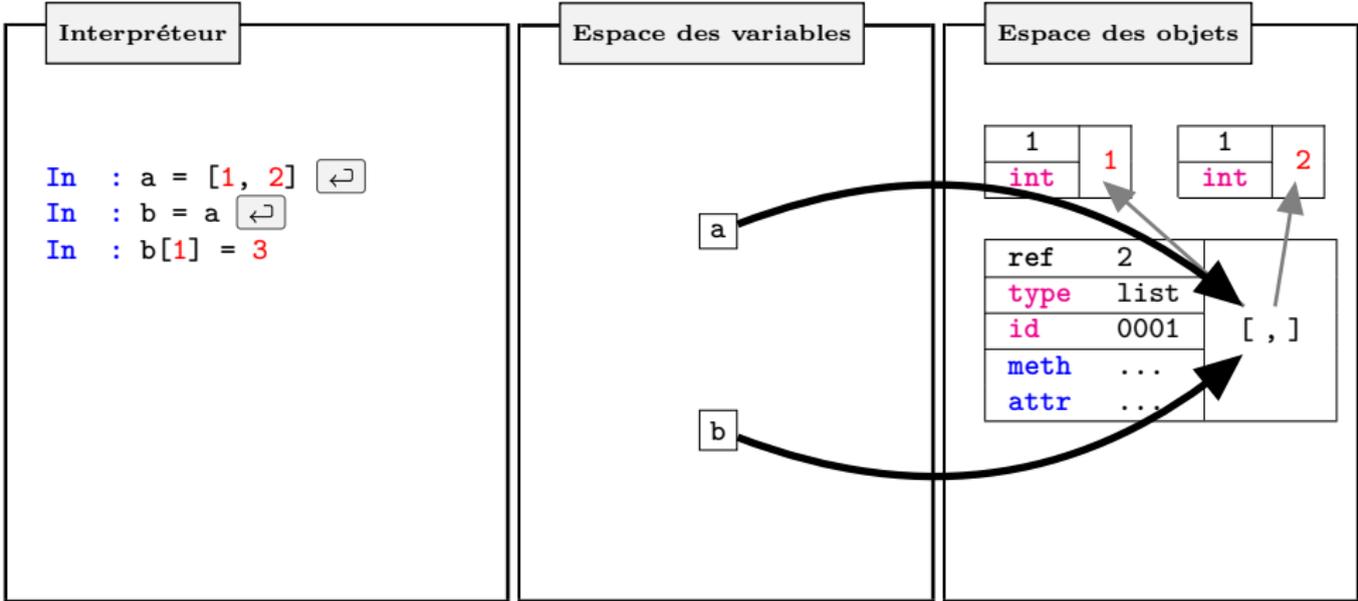
ref	1	[,]
type	list	
id	0001	
meth	...	
attr	...	

Référence **partagée** à un objet muable



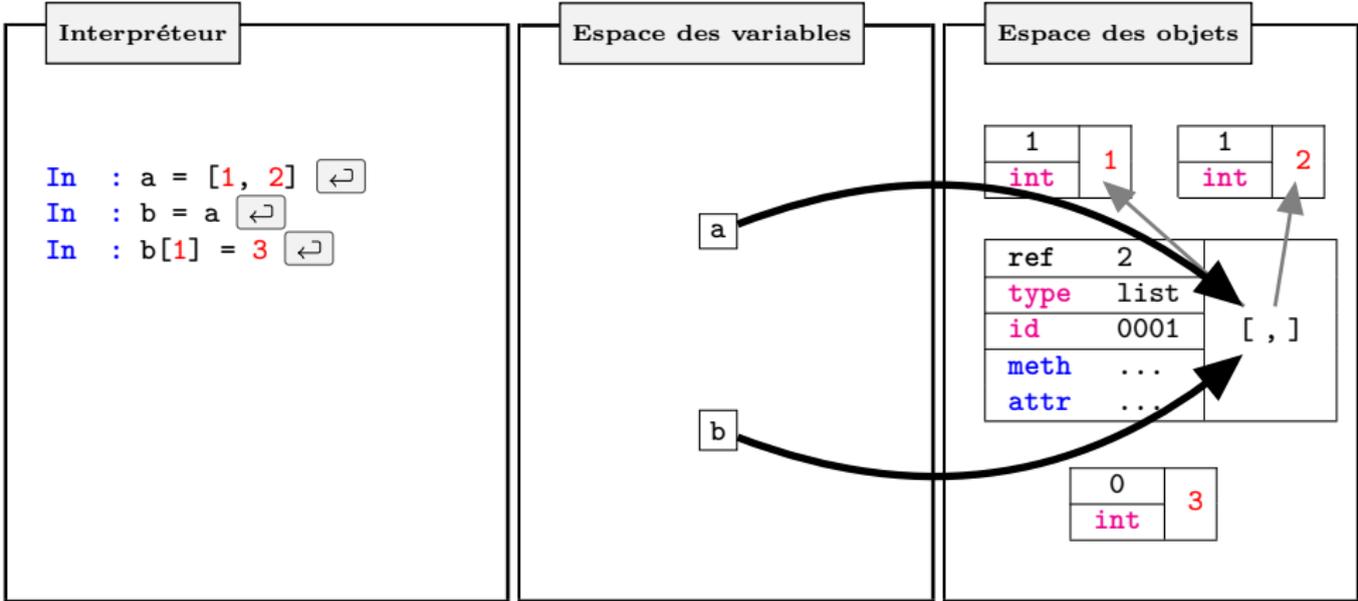
- Référence partagée : **plusieurs variables référencent un même objet**

Référence **partagée** à un objet muable



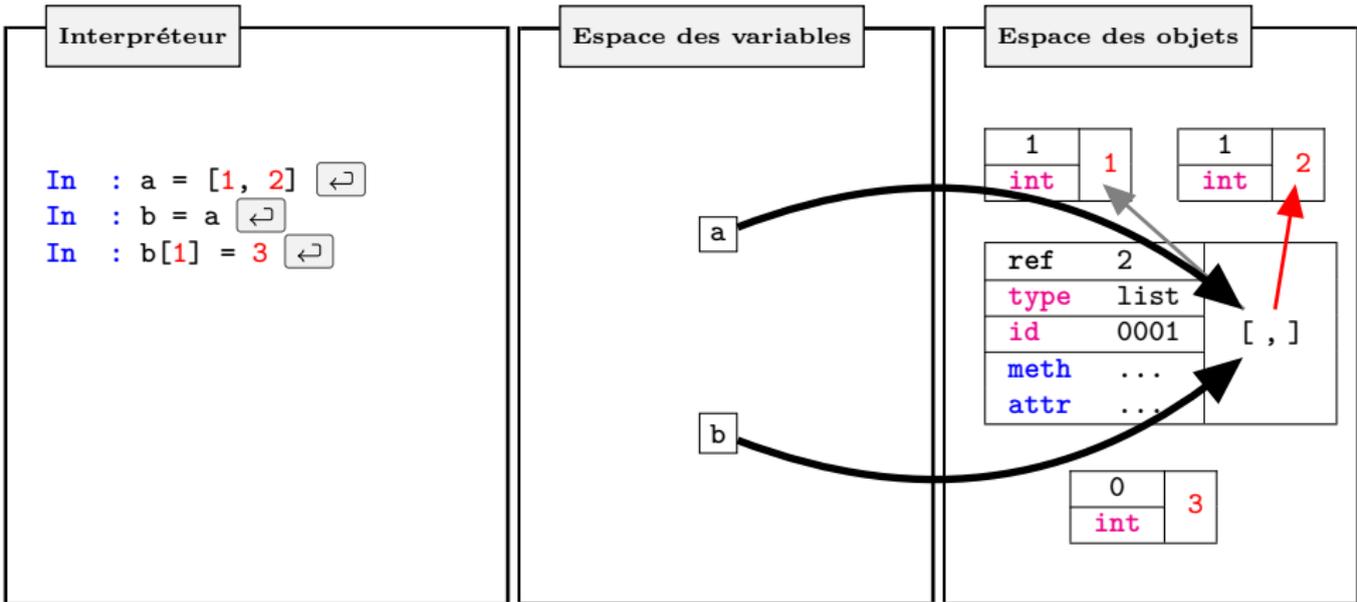
- Référence partagée : **plusieurs variables réfèrent un même objet**

Référence **partagée** à un objet muable



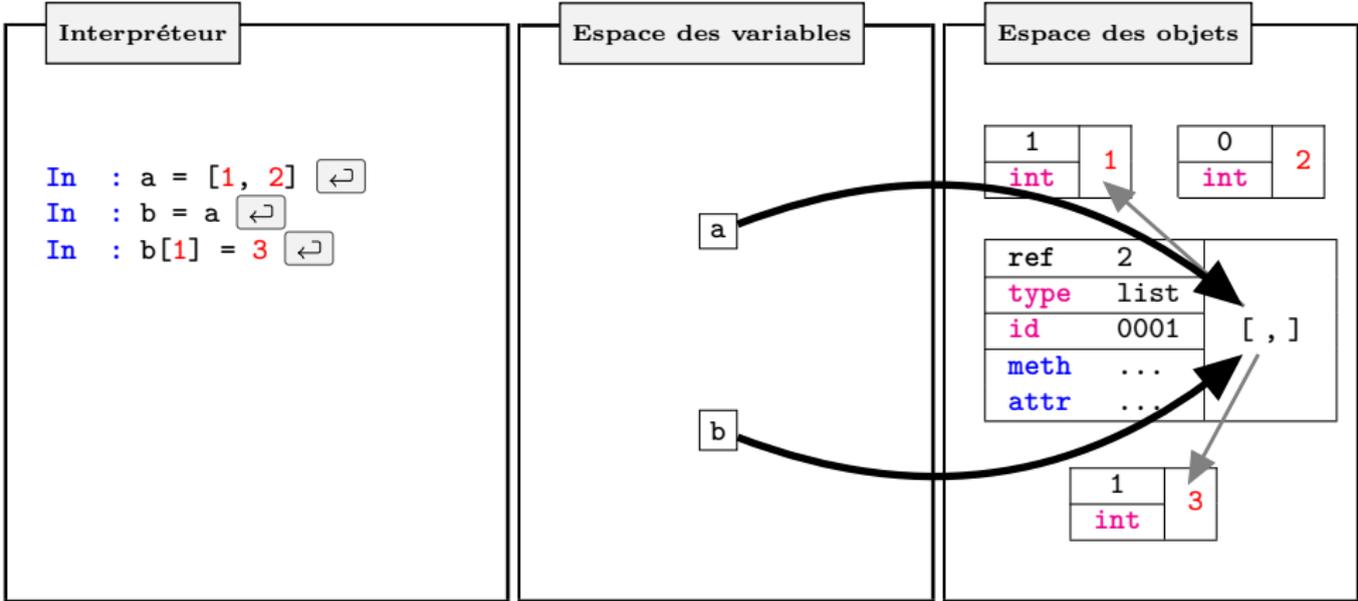
- Référence partagée : **plusieurs variables réfèrent un même objet**

Référence **partagée** à un objet muable



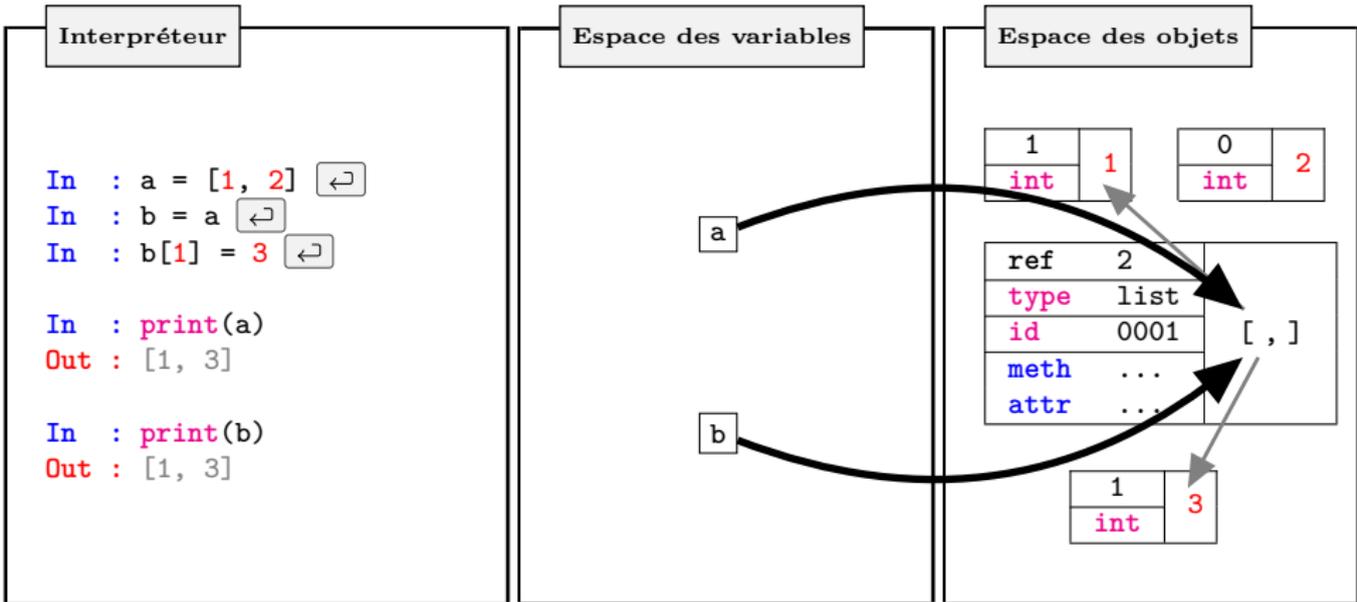
- Référence partagée : **plusieurs variables référencent un même objet**

Référence **partagée** à un objet muable



- Référence partagée : **plusieurs variables référencent un même objet**

Référence **partagée** à un objet muable



- Référence partagée : **plusieurs variables réfèrent un même objet**
- Avec des objets **muables** : effets de bord

La copie superficielle

Interpréteur

In : a = [1, 2]

Espace des variables

Espace des objets

La copie **superficielle**

Interpréteur

In : a = [1, 2] ↵

Espace des variables

Espace des objets

1

2

[,]

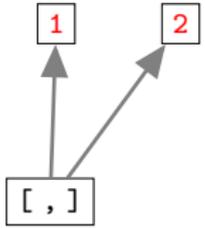
La copie superficielle

Interpréteur

In : a = [1, 2] ↵

Espace des variables

Espace des objets



La copie superficielle

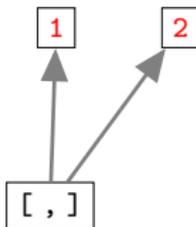
Interpréteur

```
In : a = [1, 2] ↵
```

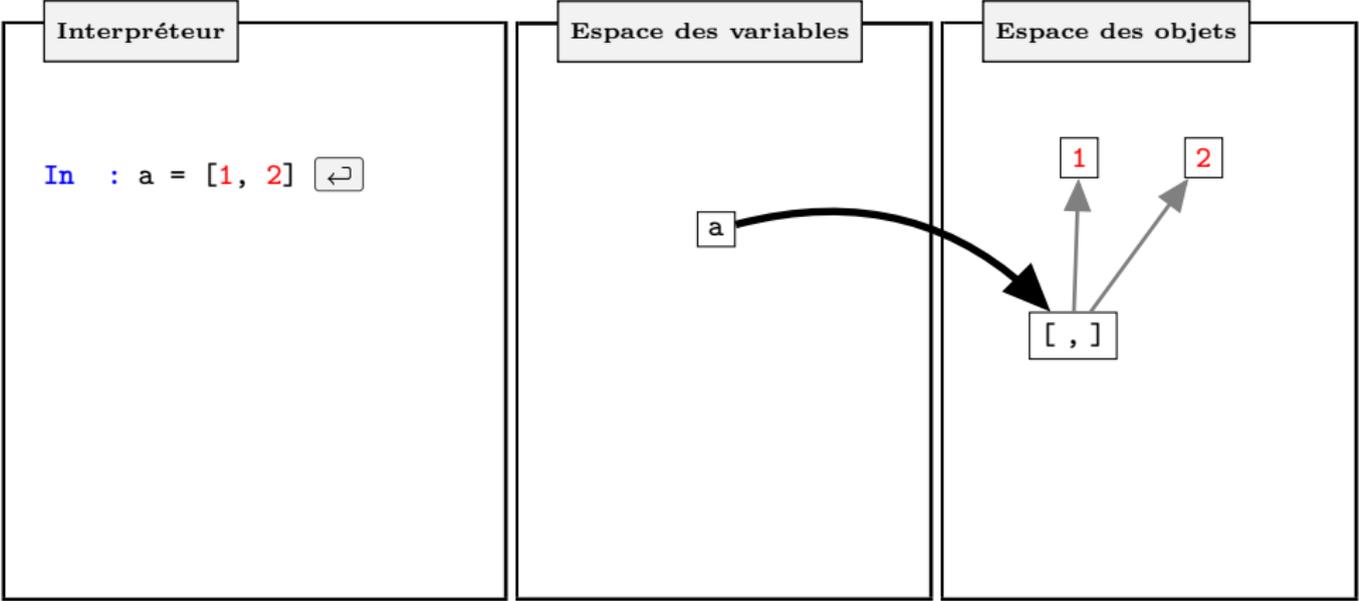
Espace des variables

a

Espace des objets



La copie superficielle



La copie **superficielle**

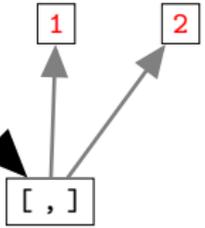
Interpréteur

```
In : a = [1, 2] ↵  
In : b = a[:]
```

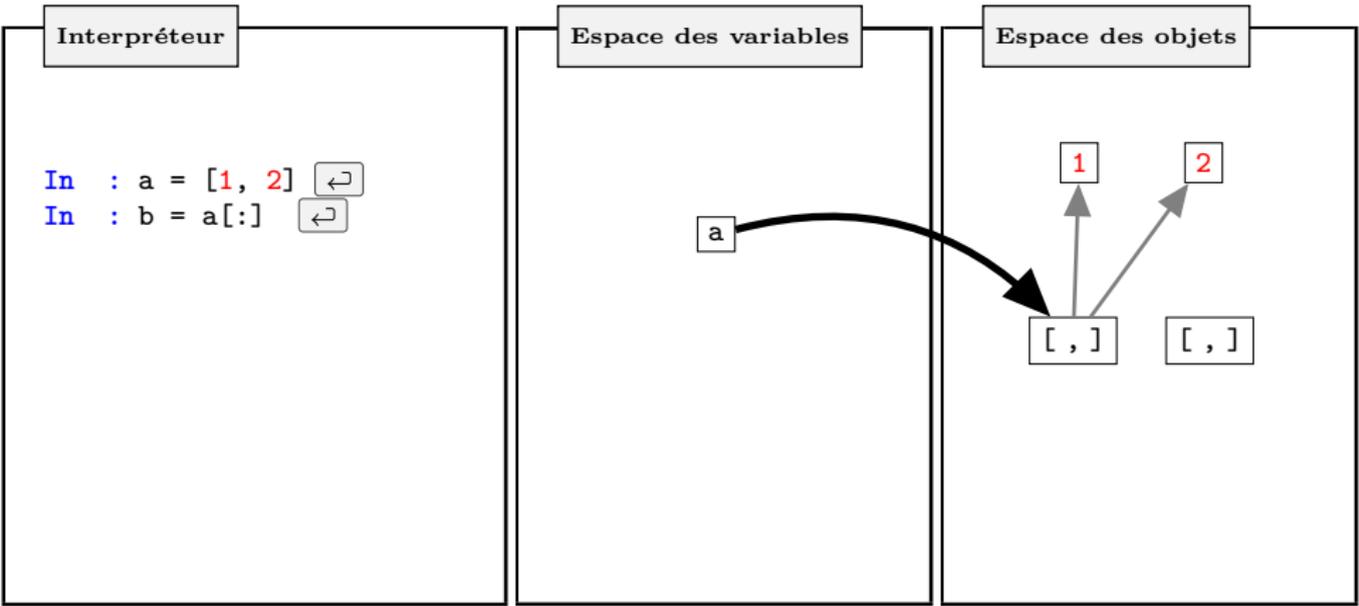
Espace des variables

a

Espace des objets

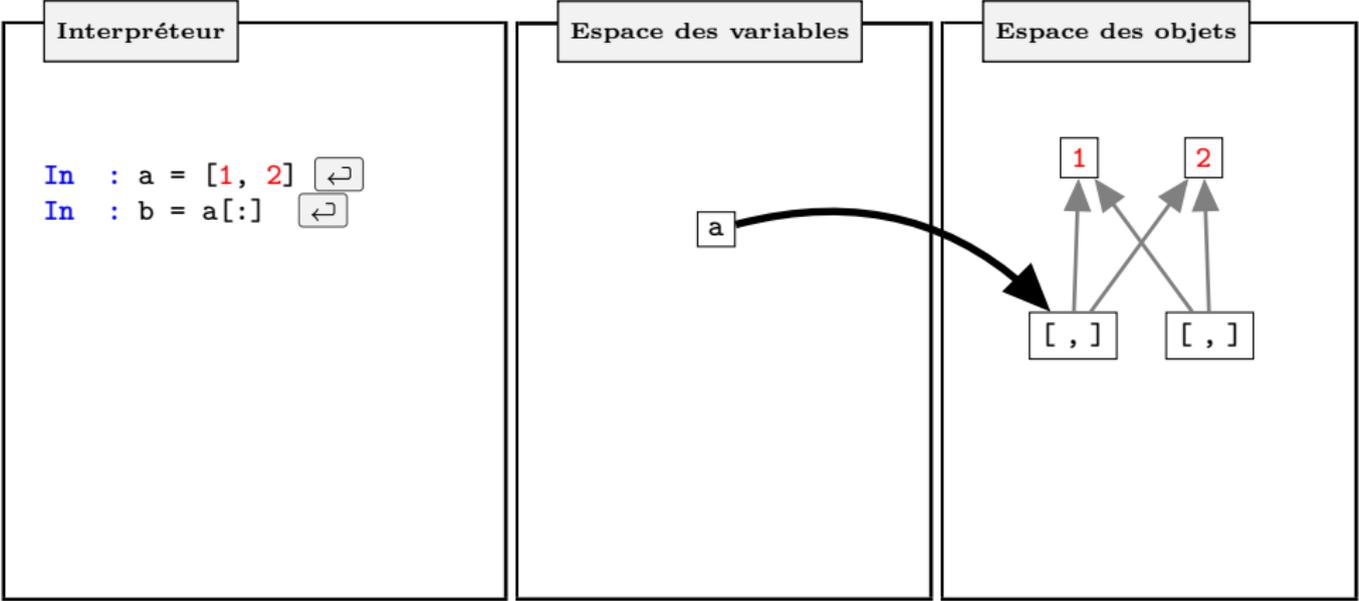


La copie superficielle



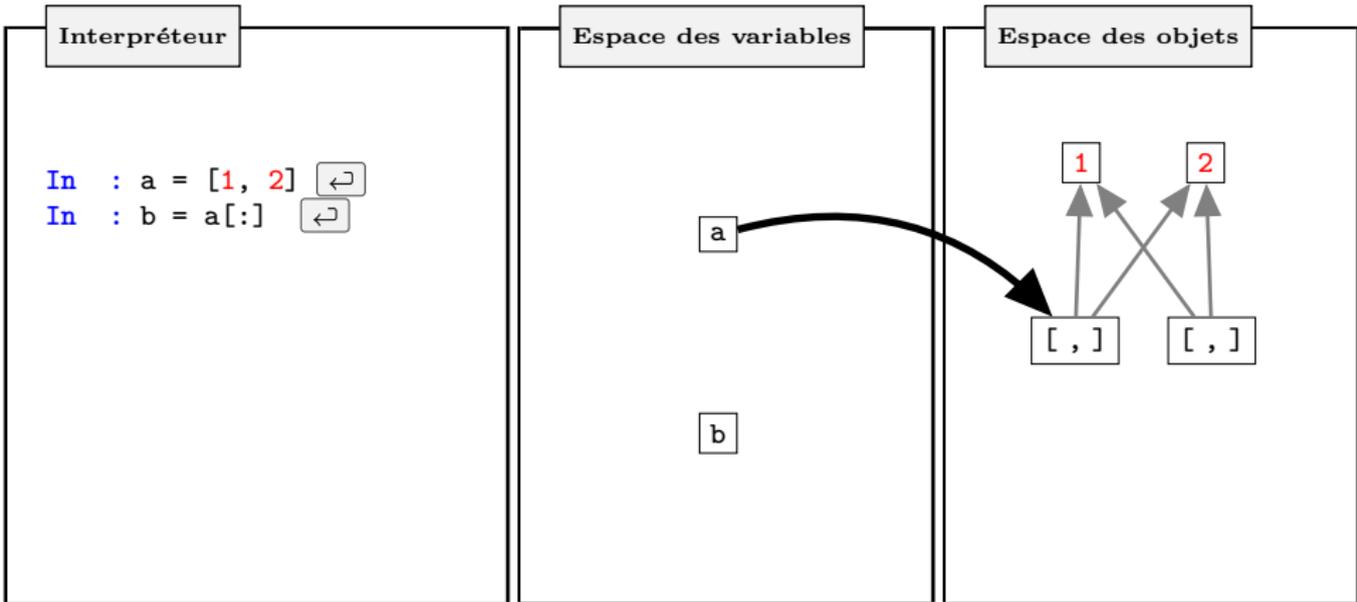
- Copie superficielle (*shallow copy*) : **slice** ou **méthode copy()**

La copie superficielle



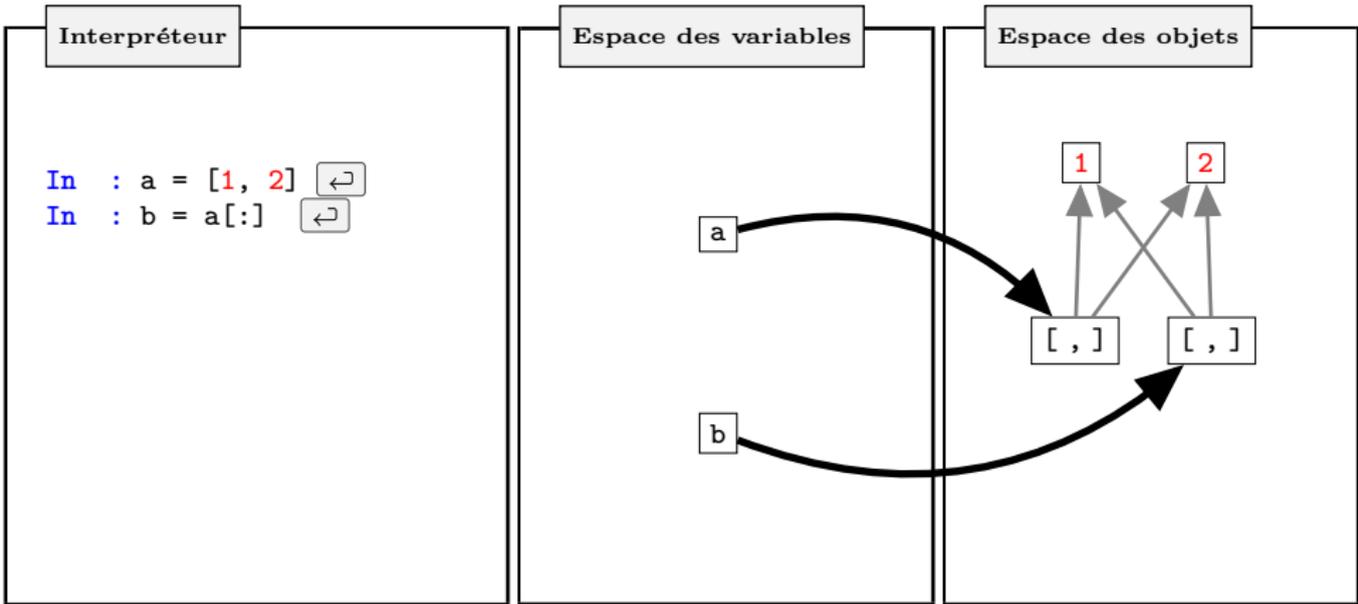
- Copie superficielle (*shallow copy*) : **slice** ou **méthode copy()**

La copie **superficielle**



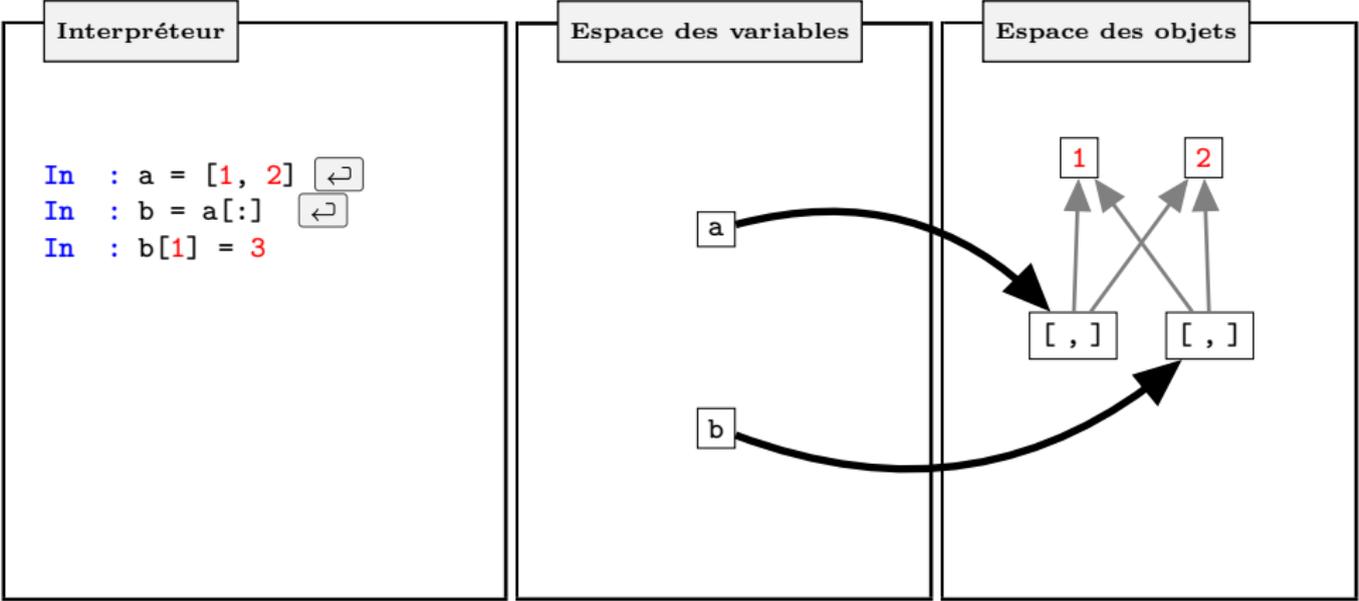
- Copie superficielle (*shallow copy*) : **slice** ou **méthode copy()**

La copie superficielle



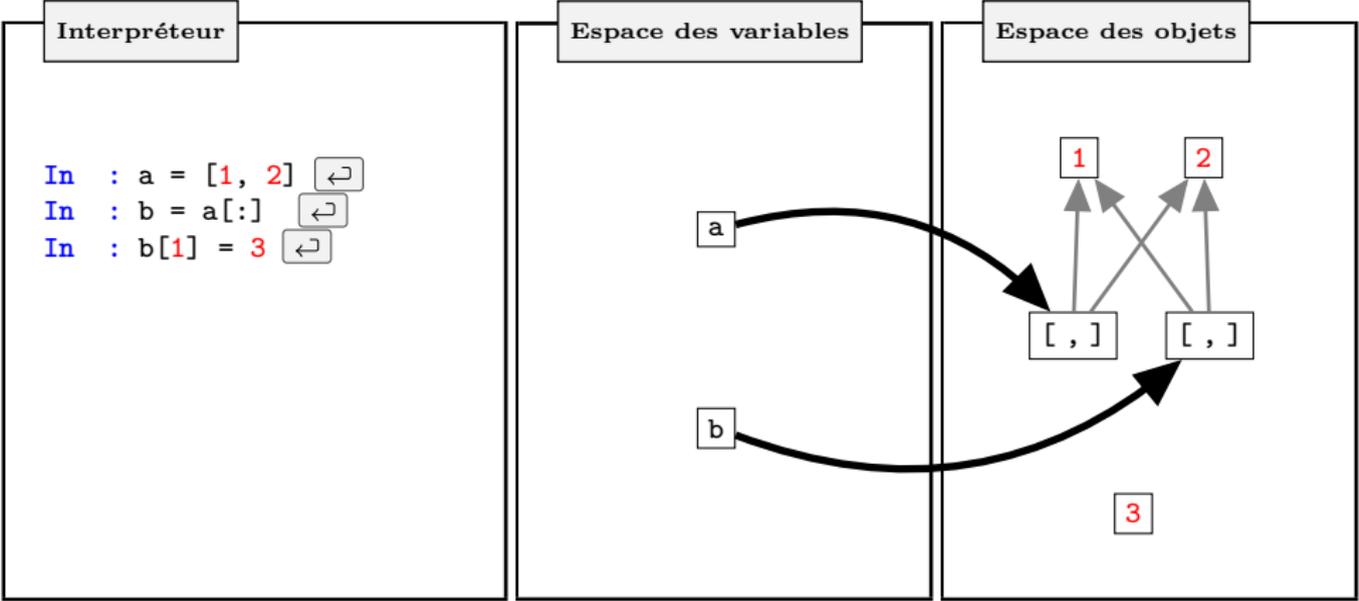
- Copie superficielle (*shallow copy*) : **slice** ou **méthode copy()**

La copie superficielle



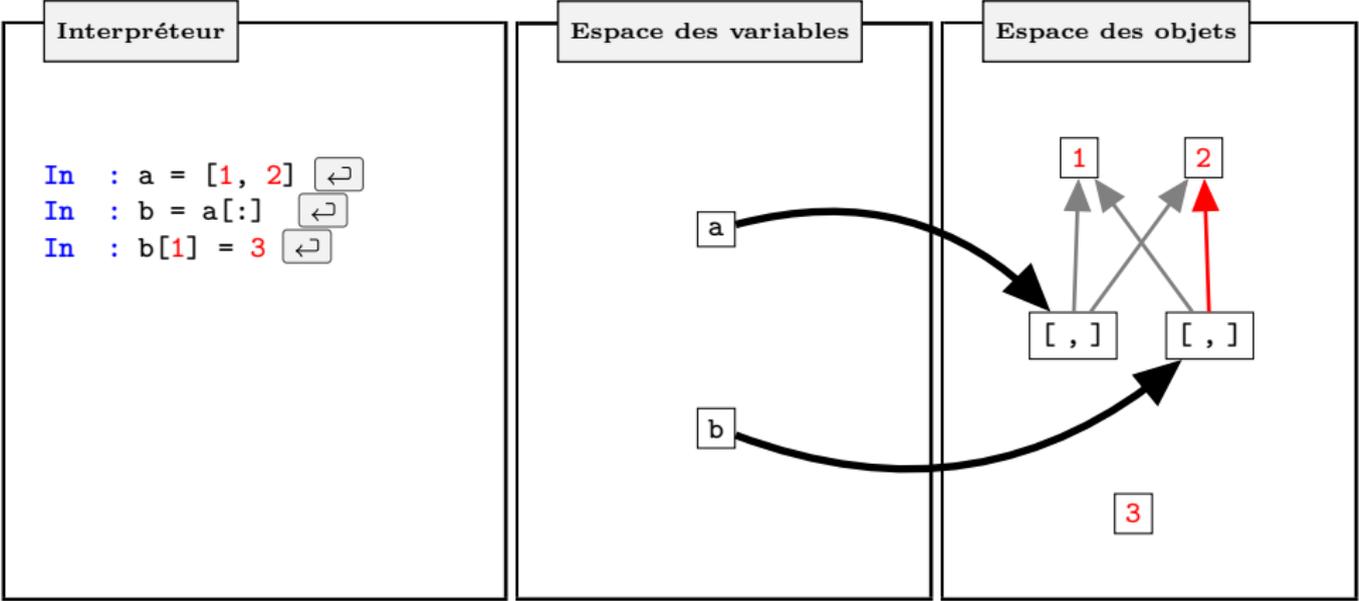
- Copie superficielle (*shallow copy*) : slice ou méthode `copy()`

La copie superficielle



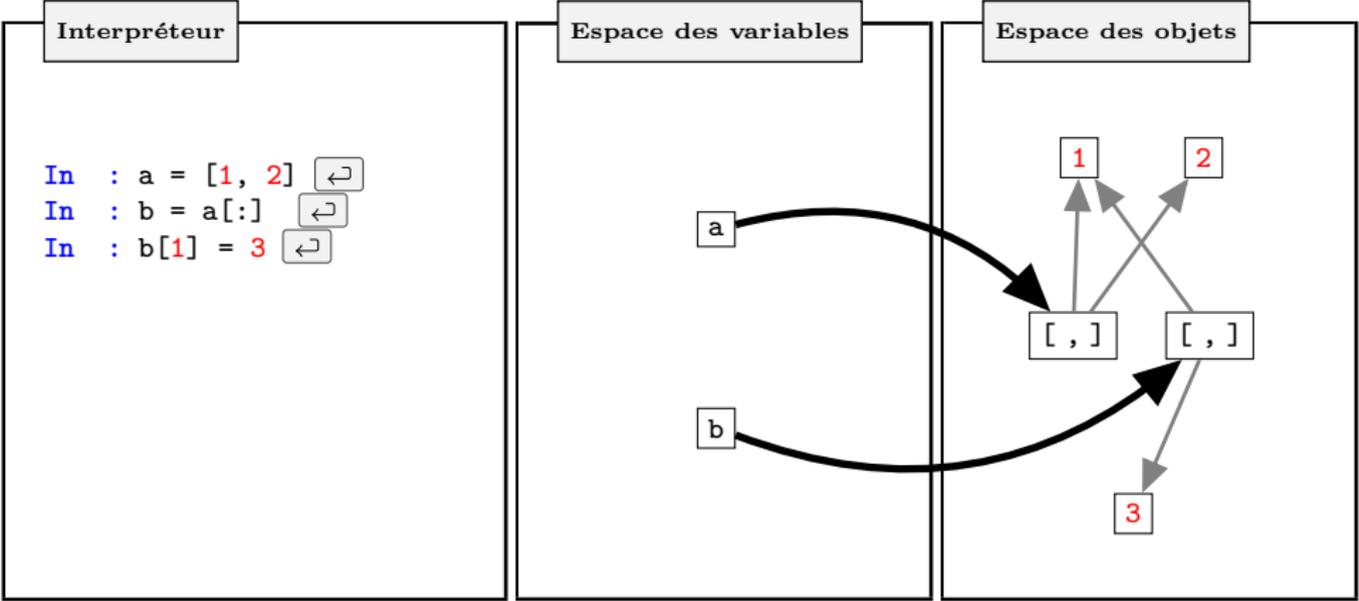
- Copie superficielle (*shallow copy*) : slice ou méthode copy()

La copie superficielle



- Copie superficielle (*shallow copy*) : slice ou méthode copy()

La copie superficielle



- Copie superficielle (*shallow copy*) : slice ou méthode copy()
- Avec des objets **muables** : solutionne à priori les effets de bord

La copie superficielle

Interpréteur

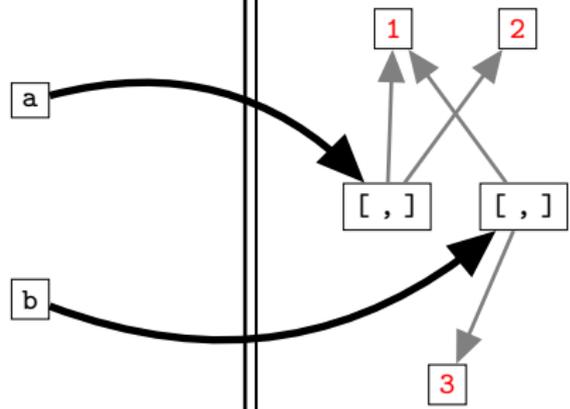
```
In : a = [1, 2] ↵  
In : b = a[:] ↵  
In : b[1] = 3 ↵  
  
In : print(a)  
Out : [1, 2]  
  
In : print(b)  
Out : [1, 3]
```

Espace des variables

a

b

Espace des objets



- Copie superficielle (*shallow copy*) : slice ou méthode `copy()`
- Avec des objets **muables** : solutionne à priori les effets de bord

Limite de la copie superficielle

Interpréteur

```
In : a = [1, [2]]
```

Espace des variables

Espace des objets

Limite de la copie superficielle

Interpréteur

```
In : a = [1, [2]] ↵
```

Espace des variables

Espace des objets

1 [] 2

[,]

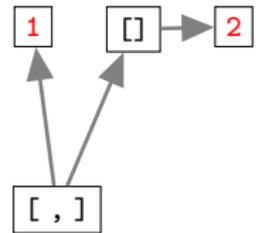
Limite de la copie superficielle

Interpréteur

```
In : a = [1, [2]] ↵
```

Espace des variables

Espace des objets



Limite de la copie superficielle

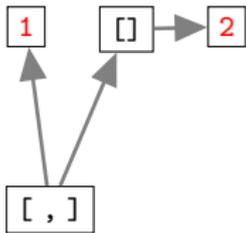
Interpréteur

```
In : a = [1, [2]] ↵
```

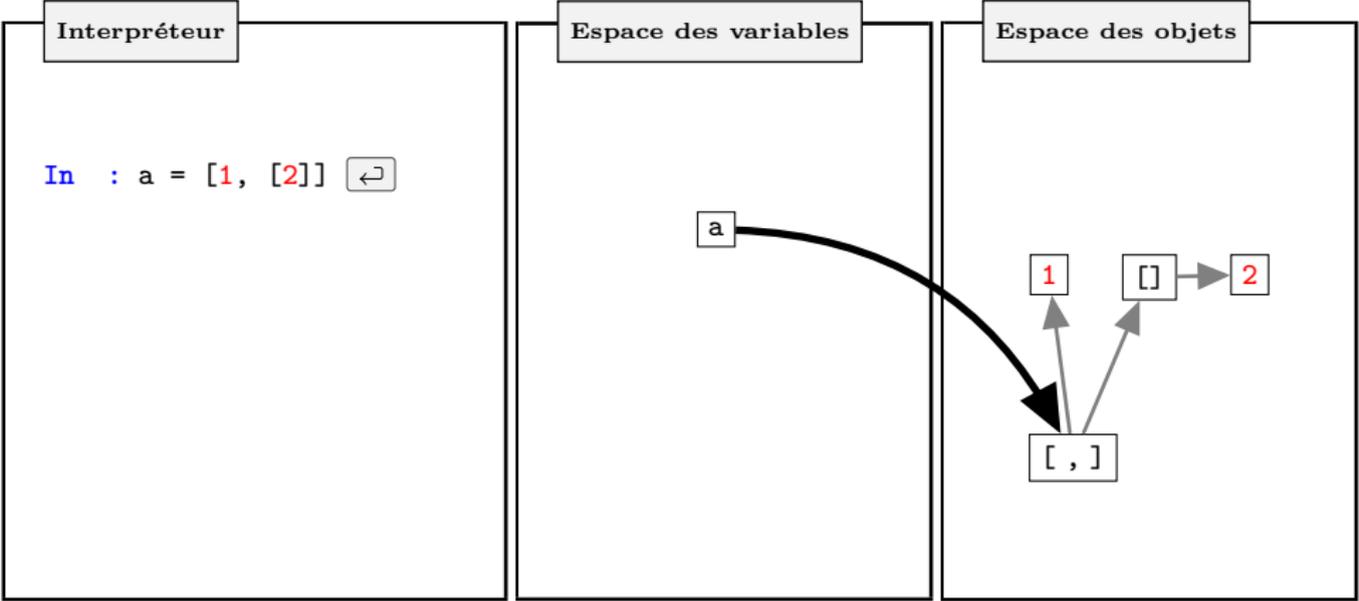
Espace des variables

a

Espace des objets



Limite de la copie superficielle



Limite de la copie superficielle

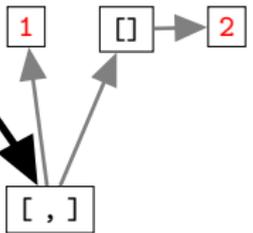
Interpréteur

```
In : a = [1, [2]] ↵  
In : b = a[:]
```

Espace des variables

a

Espace des objets



Limite de la copie superficielle

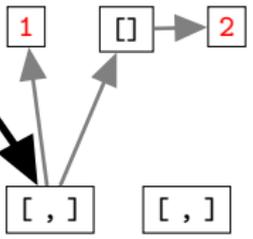
Interpréteur

```
In : a = [1, [2]] ↵  
In : b = a[:] ↵
```

Espace des variables

a

Espace des objets



Limite de la copie superficielle

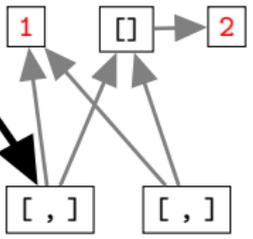
Interpréteur

```
In : a = [1, [2]] ↵  
In : b = a[:] ↵
```

Espace des variables

a

Espace des objets



Limite de la copie superficielle

Interpréteur

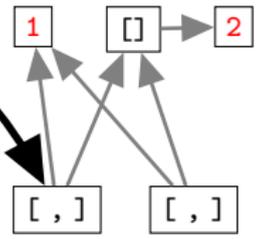
```
In : a = [1, [2]] ↵  
In : b = a[:] ↵
```

Espace des variables

a

b

Espace des objets



Limite de la copie superficielle

Interpréteur

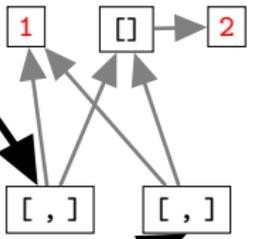
```
In : a = [1, [2]] ↵  
In : b = a[:] ↵
```

Espace des variables

a

b

Espace des objets



Limite de la copie superficielle

Interpréteur

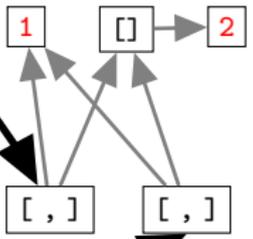
```
In : a = [1, [2]] ↵  
In : b = a[:] ↵  
In : b[1][0] = 3
```

Espace des variables

a

b

Espace des objets



Limite de la copie superficielle

Interpréteur

```
In : a = [1, [2]] ↵  
In : b = a[:] ↵  
In : b[1][0] = 3 ↵
```

Espace des variables

a

b

Espace des objets

3

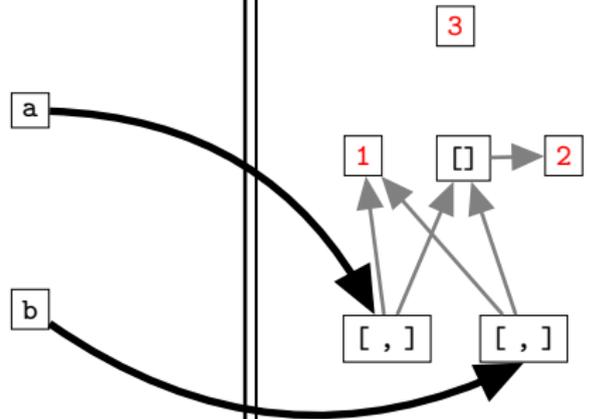
1

[]

2

[,]

[,]



Limite de la copie superficielle

Interpréteur

```
In : a = [1, [2]] ↵
In : b = a[:] ↵
In : b[1][0] = 3 ↵
```

Espace des variables

a

b

Espace des objets

3

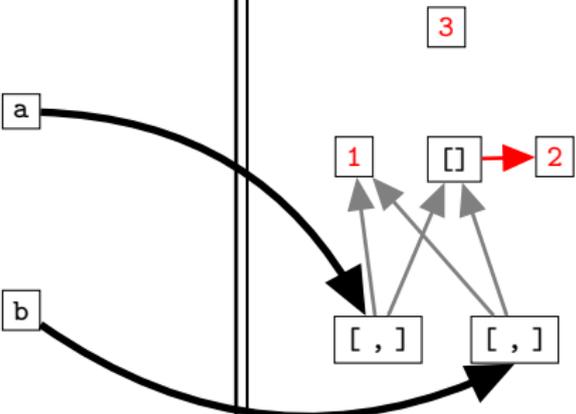
1

[]

2

[,]

[,]



Limite de la copie superficielle

Interpréteur

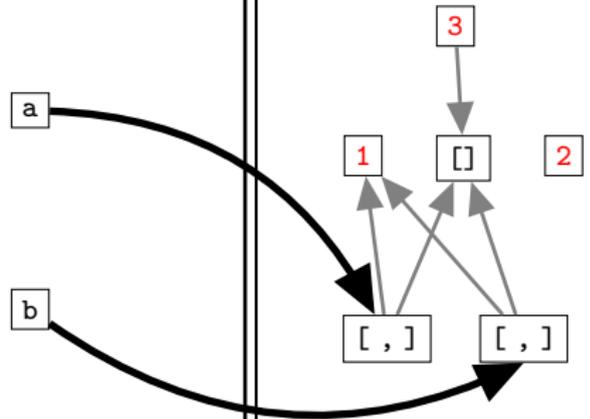
```
In : a = [1, [2]] ↵  
In : b = a[:] ↵  
In : b[1][0] = 3 ↵
```

Espace des variables

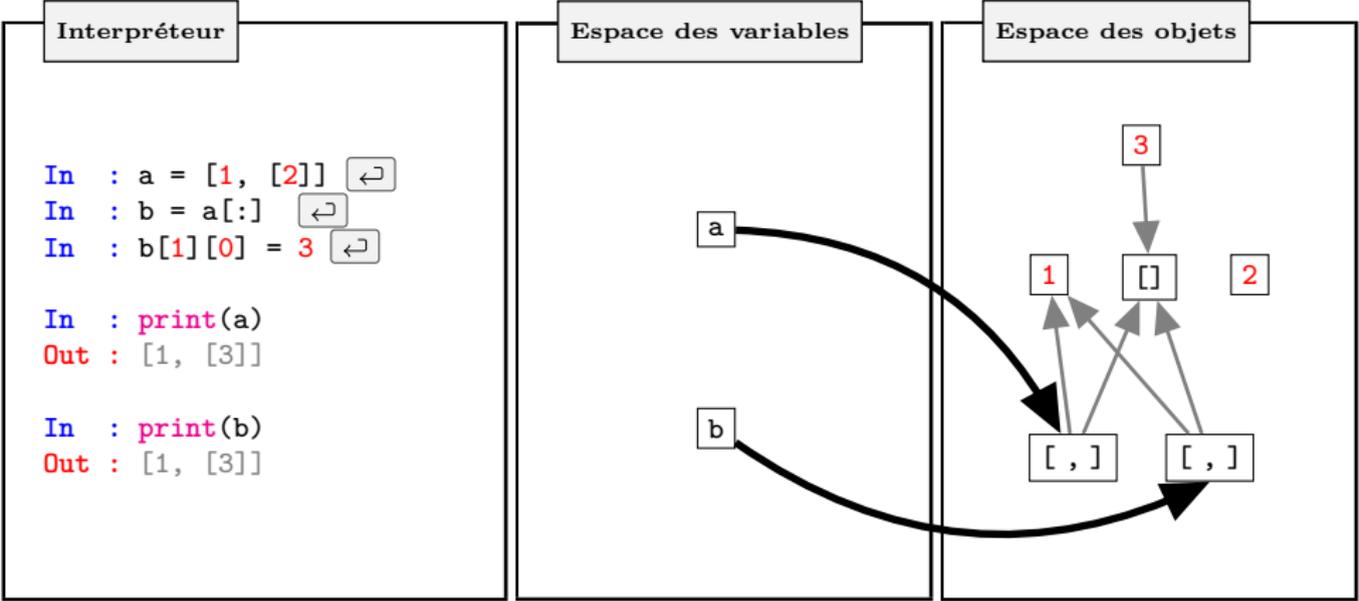
a

b

Espace des objets



Limite de la copie superficielle



- **Limitation : Objet mutable faisant référence à un autre objet mutable**
- **Copie superficielle : effets de bord dans cette situation**

La copie **profonde**

Interpréteur

```
In : import copy  
In : a = [1, [2]]
```

Espace des variables

Espace des objets

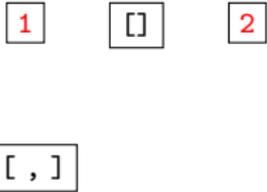
La copie **profonde**

Interpréteur

```
In : import copy  
In : a = [1, [2]]
```

Espace des variables

Espace des objets



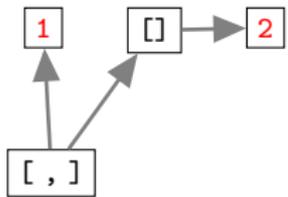
La copie **profonde**

Interpréteur

```
In : import copy  
In : a = [1, [2]]
```

Espace des variables

Espace des objets



La copie **profonde**

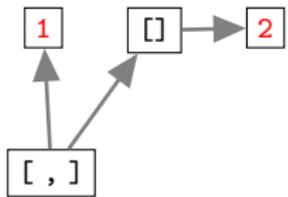
Interpréteur

```
In : import copy  
In : a = [1, [2]]
```

Espace des variables

a

Espace des objets



La copie **profonde**

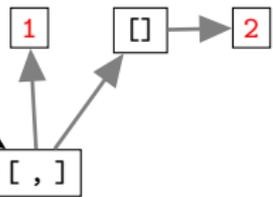
Interpréteur

```
In : import copy  
In : a = [1, [2]]
```

Espace des variables

a

Espace des objets



La copie **profonde**

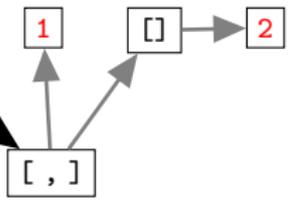
Interpréteur

```
In : import copy  
In : a = [1, [2]]  
In : b = copy.deepcopy(a)
```

Espace des variables

a

Espace des objets



- Copie en profondeur (*deep copy*): **méthode `deepcopy` du module `copy`**

La copie **profonde**

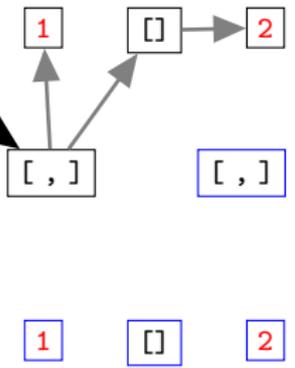
Interpréteur

```
In : import copy  
In : a = [1, [2]]  
In : b = copy.deepcopy(a)
```

Espace des variables



Espace des objets



- Copie en profondeur (*deep copy*): **méthode `deepcopy` du module `copy`**

La copie **profonde**

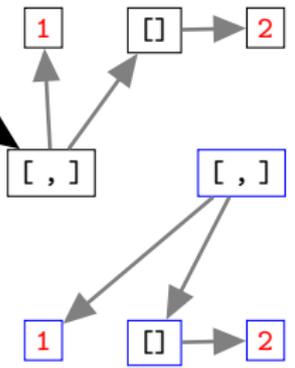
Interpréteur

```
In : import copy
In : a = [1, [2]]
In : b = copy.deepcopy(a)
```

Espace des variables

a

Espace des objets



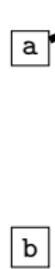
- Copie en profondeur (*deep copy*): **méthode deepcopy du module copy**

La copie **profonde**

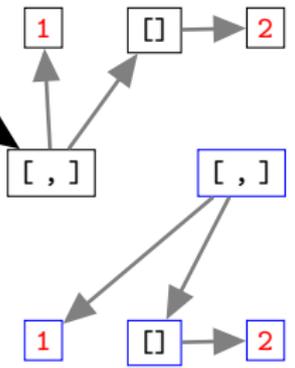
Interpréteur

```
In : import copy
In : a = [1, [2]]
In : b = copy.deepcopy(a)
```

Espace des variables



Espace des objets



- Copie en profondeur (*deep copy*): **méthode deepcopy du module copy**

La copie **profonde**

Interpréteur

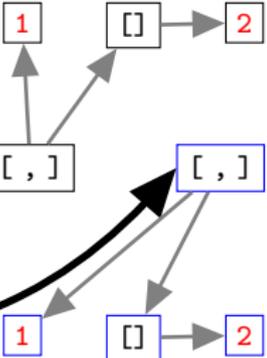
```
In : import copy
In : a = [1, [2]]
In : b = copy.deepcopy(a)
```

Espace des variables

a

b

Espace des objets



- Copie en profondeur (*deep copy*): **méthode deepcopy du module copy**

La copie **profonde**

Interpréteur

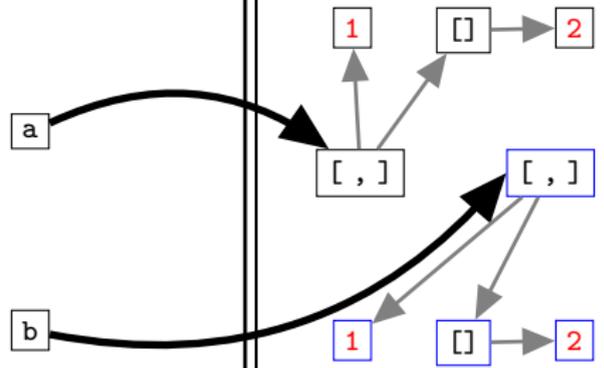
```
In : import copy
In : a = [1, [2]]
In : b = copy.deepcopy(a)
In : b[1][0] = 3
```

Espace des variables

a

b

Espace des objets



- Copie en profondeur (*deep copy*): méthode `deepcopy` du module `copy`

La copie **profonde**

Interpréteur

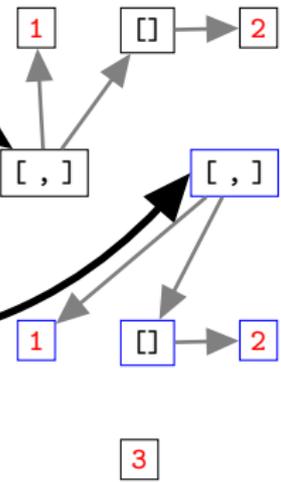
```
In : import copy
In : a = [1, [2]]
In : b = copy.deepcopy(a)
In : b[1][0] = 3
```

Espace des variables

a

b

Espace des objets



- Copie en profondeur (*deep copy*): méthode `deepcopy` du module `copy`

La copie **profonde**

Interpréteur

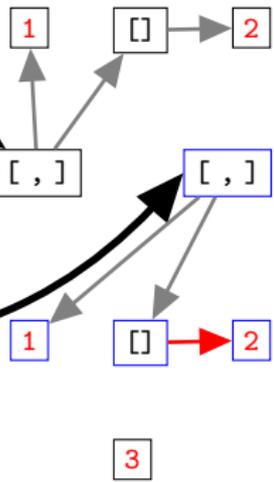
```
In : import copy  
In : a = [1, [2]]  
In : b = copy.deepcopy(a)  
In : b[1][0] = 3
```

Espace des variables

a

b

Espace des objets



- Copie en profondeur (*deep copy*): méthode `deepcopy` du module `copy`

La copie **profonde**

Interpréteur

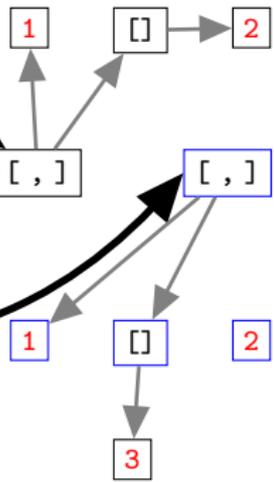
```
In : import copy
In : a = [1, [2]]
In : b = copy.deepcopy(a)
In : b[1][0] = 3
```

Espace des variables

a

b

Espace des objets



- Copie en profondeur (*deep copy*): méthode `deepcopy` du module `copy`

La copie **profonde**

Interpréteur

```
In : import copy
In : a = [1, [2]]
In : b = copy.deepcopy(a)
In : b[1][0] = 3

In : print(a)
Out : [1, [2]]

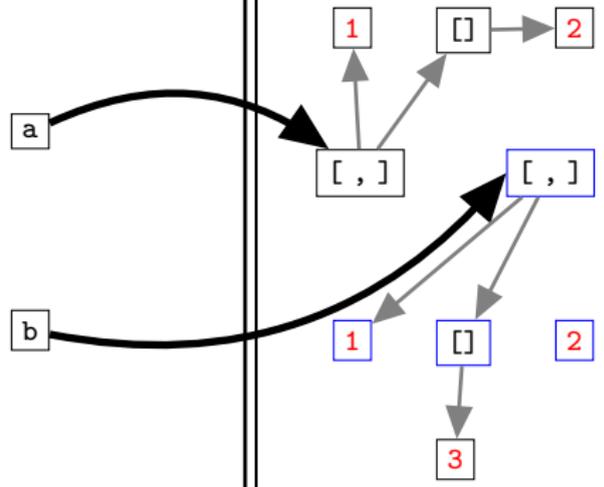
In : print(b)
Out : [1, [3]]
```

Espace des variables

a

b

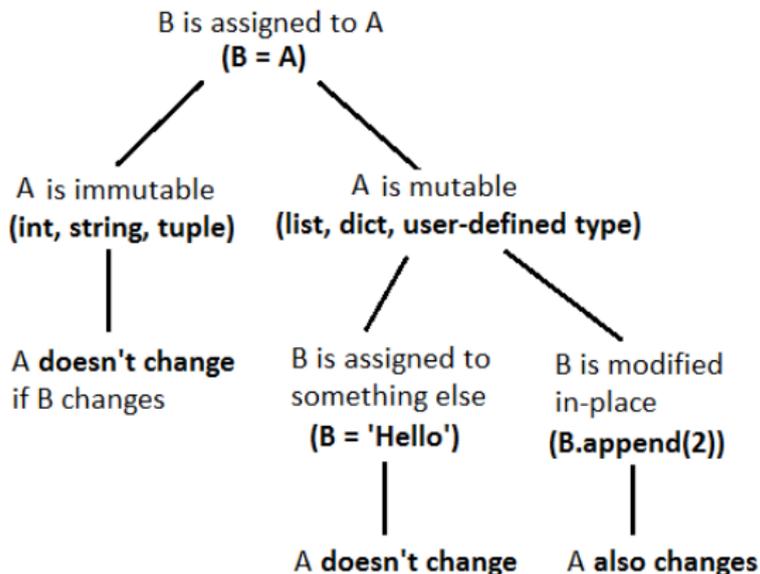
Espace des objets



- Copie en profondeur (*deep copy*): méthode `deepcopy` du module `copy`
- Solution efficace pour éviter tout effets de bord



Une dernière remarque sur la mutabilité d'un objet



From Alex Lord on grownassadultblog.com