1 Affectation, réaffectation (ou assignation, réassignation)

Astuce :
- Pour connaitre l'adresse mémoire d'un objet (ou d'une variable) : `id(ma variable)`
- Outil parfois utile : https ://pythontutor.com/

Exercice n° 1

1.

```
a=3
id(a)

b=a
id(b)

a=5
id(a)
```

Au final, que contiennent les variables a et b ?

2.

```
L=[1,2,3]
id(L)

P=L
id(P)

P[0]=5
id(P)
```

Au final, que contiennent les variables L et P ?

3.

```
import copy
L=[1,2,3]
id(L)

P=copy.deepcopy(L)
id(P)

P[0]=5
```

Au final, que contiennent les variables L et P ?

Exercice n° 2

On souhaite supprimer tous les nombres paires d'une liste de nombres entiers.

- Première tentative

```
L=[11,22,13,44,50,61]
for i in range(len(L)):
    if L[i]%2==0:
        L.pop(i)
print(L)
```

Au final, quel est le résultat affiché ?

- Deuxième tentative

```python
L=[11,22,13,44,50,61]
j=0
for i in range(len(L)):
    if L[j]%2==0:
        L.pop(j)
    j=j+1
print(L)
```

Au final, quel est le résultat affiché ?

— Troisième tentative

Proposer un algorithme qui permet de supprimer tous les nombres paires d'une liste.

2 Valeur égales, valeurs identiques

Deux valeurs sont **égales** lorsqu'elles partagent un même état : par exemple, deux chaînes qui contiennent les mêmes caractères sont égales.
Deux valeurs sont **identiques** si elles font référence à la même mémoire.

En Python, on retrouve ces concepts sous les opérateurs == (égalité) et is (identité).

Exercice n° 3

```
L=[1,2,3]
P=[1,2,3]
print(P==L)
...
print(P is L)
...
```

P et L sont-ils identiques (même adresse mémoire) ?

P et L sont-ils égaux ?

Exercice n° 4

```
L=[1,2,3]
P=L
print(P==L)
...
print(P is L)
...
```

P et L sont-ils identiques (même adresse mémoire) ?

P et L sont-ils égaux ?

Exercice n° 5

```
import copy
L=[1,2,3]
P=copy.deepcopy(L)
print(P==L)
...
print(P is L)
...
```

P et L sont-ils identiques (même adresse mémoire) ?

P et L sont-ils égaux ?
Remarque :
Deux objets identiques (même adresse mémoire) sont égaux mais deux objets égaux ne sont pas forcement identiques (n'ont pas forcement la même adresse mémoire)

3 Dictionnaires

Exercice n° 6

Ecrire une fonction MaxDicoV1 qui renvoi la plus grande valeur dans un dictionnaire.

Exemple :

```
dico1={'a':23, 'z':12, 'e':7, 'r':-1, 't':121, 'y':89}
MaxDicoV1(dico1)
't'
dico1={'a':-23, 'z':-12, 'e':-7, 'r':-1, 't':-121, 'y':-89}
MaxDicoV1(dico1)
'r'
```

Ecrire une fonction MaxDicoV2 qui renvoi l'ensemble des clés qui correspondent à la plus grande valeur dans un dictionnaire.

Exemple :

```
dico1={'a':23, 'z':12, 'e':23, 'r':-1, 't':23, 'y':89}
MaxDicoV2(dico1)
[''a','e','t']
dico1={'a':-23, 'z':-1, 'e':-7, 'r':-1, 't':-121, 'y':-89}
MaxDicoV2(dico1)
['r','z']
```

4 Objets muables (mutables en anglais) et immuables

```
Liste = [1, 2, 3]
# On affiche la place en mémoire de la liste
print(id(liste))
# On modifie la liste avec la méthode append
liste.append(4)
# On affiche de nouveau la place en mémoire de la liste
print(id(liste))
```

```
v=(1,2,3)
print(id(v))
v=v+(4,5)
print(v)
print(id(v))

print(v[0])
v[0]=5
# Comment modifier le (1,2,3) en (5,2,3) ?
```

```
c="Hello"
print(id(v))
c=c+ "!"
print(id(v))

print(v[1])
v[1]="a"
# Comment modifier le "Hello" en "Hallo" ?
```

En Python, les objets de type bool, int, str, tuple, sont immuables.
Les liste et les dictionnaires sont des objets mutables.

Remarque : les clefs d'un dictionnaire doivent être de type non mutable