

## Corrigé du TP Informatique 4

### Exercice 1

1. On observe :

```
a= 1 b= 2
swap(a,b)= (2, 1)
a= 1 b= 2
```

Les arguments de `swap` sont des variables locales et n'entrent pas en conflit avec les variables globales du même nom. Ainsi, le dernier affichage est identique au premier puisque les variables globales n'ont subi aucune modification.

2. Dans la fonction `swap`, les arguments sont modifiés : cette pratique est à éviter.

### Exercice 2

On saisit :

```
def vit(h):
    return np.sqrt(2*g*h)

h=1
print("h=",h,"m, V=",vit(h),"m/s")
```

### Exercice 3

1. On teste :

```
>>> impair(2)
{'n': 2}
False
```

L'argument `n` est une variable locale de la fonction `impair`.

2. On saisit :

```
def impair(n):
    r=n%2
    print(locals())
    return r==1
```

On observe :

```
>>> impair(2)
{'n': 2, 'r': 0}
False
```

Lors de l'affichage, la variable locale `r` a été créée et elle apparaît dans la liste.

## Exercice 4

On saisit :

```
def taille(n):  
    return len(str(n))
```

## Exercice 5

On saisit :

```
def eq(f,a,b):  
    m=(f(b)-f(a))/(b-a)  
    p=f(a)-a*m  
    return [m,p]
```

On teste :

```
>>> eq(lambda x:x,1,2)  
[1.0, 0.0]  
>>> eq(lambda x:x**2,1,2)  
[3.0, -2.0]
```

## Exercice 6

1. On saisit :

```
def lyc(n):  
    def sym(p):  
        return int(str(p)[::-1])  
    b=str(n+sym(n))  
    return b==b[::-1]
```

2. On enlève au choix le caractère # dans :

```
def lyc(n):  
    def sym(p):  
        # print(locals())  
        return int(str(p)[::-1])  
    b=str(n+sym(n))  
    # print(locals())  
    return b==b[::-1]
```

Pour les variables locales de `sym` avec `n=145`, on obtient

```
{'p': 145}  
n= 145 True
```

et pour celles de `lyc` avec `n=19`

```
{'n': 19, 'b': '110', 'sym': <function lyc.<locals>.sym at 0x000000003310F28>}
```

On constate que la fonction `sym` est bien traitée comme variable locale de `lyc`.

## Exercice 7

On saisit :

```
def numjour(date):
    j=int(date[:2])
    m=int(date[3:])
    tm=[31,28,31,30,31,30,31,31,30,31,30,31]
    return j<=tm[m-1]
```

## Exercice 8

On saisit :

```
def uni(L):
    return L!=[L[0]]*len(L)
```

## Exercice 9

1. On saisit :

```
def renv1(L):
    return L[::-1]
```

2. On saisit :

```
def renv2(L):
    L[:]=L[::-1]
```